ARTICLE

Check for updates

Neural structure fields with application to crystal structure autoencoders

Naoya Chiba [1,2], Yuta Suzuki [3,4], Tatsunori Taniai [1], Ryo Igarashi [1], Yoshitaka Ushiku [1], Kotaro Saito[5,6] & Kanta Ono [6 ✉]

Representing crystal structures of materials to facilitate determining them via neural networks is crucial for enabling machine-learning applications involving crystal structure estimation. Among these applications, the inverse design of materials can contribute to explore materials with desired properties without relying on luck or serendipity. Here, we propose neural structure fields (NeSF) as an accurate and practical approach for representing crystal structures using neural networks. Inspired by the concepts of vector fields in physics and implicit neural representations in computer vision, the proposed NeSF considers a crystal structure as a continuous field rather than as a discrete set of atoms. Unlike existing grid-based discretized spatial representations, the NeSF overcomes the tradeoff between spatial resolution and computational complexity and can represent any crystal structure. We propose an autoencoder of crystal structures that can recover various crystal structures, such as those of perovskite structure materials and cuprate superconductors. Extensive quantitative results demonstrate the superior performance of the NeSF compared with the existing grid-based approach.

[1] OMRON SINIC X Corporation, Nagase Hongo Building 3F, 5-24-5 Hongo, Bunkyo, Tokyo, Japan. [2] Graduate School of Information Sciences, Tohoku University, 6-6-01 Aramaki-Aza-Aoba, Aoba, Sendai, Miyagi, Japan. [3] TOYOTA Motor Corporation, 1200, Mishuku, Susono, Shizuoka, Japan. [4] The Graduate University for Advanced Studies (SOKENDAI), 1-1 Oho, Tsukuba, Japan. [5] Randeft Inc., TOKIWA Bridge 13F, 2-7-4, Otemachi, Chiyoda, Tokyo, Japan. [6] Department of Applied Physics, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan. ✉email: ono@ap.eng.osaka-u.ac.jp

A fundamental paradigm in materials science considers structure–property relationships assuming that the material properties are tightly coupled with their crystal structures. Thus, for conventional approaches in materials science, theoretical and experimental analyses of the structure–property relationships of materials are conducted in the search for novel materials with superior properties[1,2]. However, these conventional approaches rely on labor-intensive human analysis and even "serendipity".

To automate or assist material analysis and development, data-driven approaches have been actively studied in materials science, establishing the area of materials informatics (MI)[3–6]. Unlike conventional approaches based on the deduction of physical laws, MI aims to unveil materials knowledge (e.g., laws governing structure–property relationships) from datasets of collected materials via statistical and machine-learning (ML) methods. In recent years, MI has been developed rapidly owing to technological advances in ML and the advent of large-scale materials databases[6,7]. Thus, powerful neural-network-based ML methods are becoming key components in MI research[6]. Applications of MI include the prediction of material properties from material characteristic data such as crystal structures[6,8] and compositions[9–12], automated analyses of experimental data[13–15], and natural language processing for knowledge retrieval from scientific literature[16].

Many MI studies have been focused on predicting properties of given materials[17–19], such as the bandgap, Seebeck coefficient, and elastic modulus. When this type of task is viewed as the discovery of *structure-to-property relationships* among materials, there is yet another important type of task, namely, the discovery of *property-to-structure relationships*, which constitutes an inverse design problem[20–24]. Despite the great potential utility of this inverse approach in developing materials, few studies have addressed this[20–24] or its underlying problem[25–27], that is, the estimation of crystal structures under given conditions. Regarding MI and ML, whether to input or output crystal structures (i.e., *encode* or *decode* crystal structures in MI and ML terms) induces a crucial difference. Although encoding crystal structures is suitably established using graph neural networks[11,17,18,28], a technical bottleneck persists in decoding crystal structures. We addressed the bottleneck in this study.

The crystal structure of an inorganic material is a regular and periodic arrangement of atoms in a three-dimensional (3D) space. This arrangement is usually described by the 3D positions and species of atoms in a unit cell and the lattice constants defining the translations of the unit cell in 3D space. The atoms in a unit cell have no explicit order, and their quantity varies from one to hundreds in number. Because ML models, including neural networks, generally accept fixed-dimensional and consistently ordered tensors for processing, treating crystal structures with ML models is not straightforward[17], and determining crystal structures via the models is even more difficult.

We propose a general representation of crystal structures that enables neural networks to decode or determine such structures. The key concept underlying our approach is illustrated in Fig. 1, where a crystal structure is represented as a continuous vector field tied to 3D space rather than as a discrete set of atoms. We refer to our approach as *neural structure fields* (NeSF). The NeSF uses two types of vector fields, namely, the position and species fields, to implicitly represent the positions and species of the atoms in the unit cell of the crystal structure, respectively.

To illustrate the concept of the NeSF, assume that we are given target material information as a fixed-dimensional vector, z, and consider the problem of recovering a crystal structure from z. Input z may specify, for example, information on the crystal structure of a material or some desired criteria for materials to be produced. In the NeSF, we use neural network f as an implicit function to indirectly represent the crystal structure embedded in z instead of letting neural network f directly output the crystal structure as $f(z)$. Specifically, we treat f as a vector field on 3D Cartesian coordinates, p, conditioned on the target material information, z:

$$s = f(p, z) \qquad (1)$$

In the position field, network f is trained to output a 3D vector pointing from query point p to its nearest atom position, a, in the crystal structure of interest. Thus, we expect output s to be $a - p$. If the position field is ideally trained, we can retrieve position a of the nearest atom at any query point p as $p + f(p, z)$. Mathematically, the position field can be interpreted as the gradient vector field, $-\nabla\phi(p)$, of scalar potential $\phi(p) = \frac{1}{2}\min_i|p - a_i|^2$, which represents the squared distance to the nearest atom given atomic positions $\{a_i\}$ in the crystal structure. Analogously, the species field is trained to output a categorical probability distribution that indicates the species of the nearest atom. Thus, the output dimension of the species field is the number of candidate atomic species.

The proposed NeSF is inspired by the concepts of vector fields in classical physics and implicit neural representations[29–33] in computer vision. Implicit neural representations have recently been proposed to handle some representation issues in 3D computer vision applications, such as 3D shape estimation of objects[30–32] and free-viewpoint image synthesis[29,33]. In 3D shape estimation, letting a neural network directly output a 3D mesh or point cloud suffers from representation issues similar to those occurring in crystal structures. To overcome these issues, the signed distance function (SDF) is utilized in DeepSDF[30] to model 3D shapes by letting neural network $f(p)$ indicate whether query point p is outside or inside the object volume with a positive or negative sign in its scalar outputs, respectively. The NeSF follows the basic idea of implicit neural representations and further extends it to the estimation of crystal structures described by atomic positions and species. The precise description of atomic positions in crystal structures is of crucial interest in materials science. Thus, the NeSF outputs vectors pointing to the nearest atoms to represent atomic positions more directly than existing implicit neural representations of 3D geometries[30–32].

Our idea of representing crystal structures as continuous vector fields has been partially and implicitly explored using grid-based discretization (i.e., voxelization) in recent MI studies[20,21,23,25–27], but without explicit consideration as discretized vector fields. In those studies, the 3D space within the unit cell is discretized into voxels, and each voxel is then assigned an electron density, which essentially represents the presence or absence of an atom around the voxel. However, compared with one-dimensional (e.g., audio signals) or two-dimensional (e.g., images) data, the discretization of 3D data considerably suffers from the tradeoff between spatial resolution and computational complexity in terms of both computational time and memory space. For example, the ICSG3D method[25] uses $32 \times 32 \times 32$ voxels to represent crystal structures and estimates them using 3D convolutional neural networks (CNNs). Because voxel-based 3D CNNs are computationally and memory intensive, the resolution of $32 \times 32 \times 32$ voxels is an approximate limit for training a voxel-based model on a standard computing system. Meanwhile, existing crystal structures can contain tens or more atoms in their unit cells or have elongated or distorted unit cells. Thus, accurately representing diverse crystal structures with voxels requires a sufficiently high resolution. Moreover, voxel-based models can only provide atomic positions indirectly in representations such as peaks in a scalar field of electron densities discretized in the voxel space.
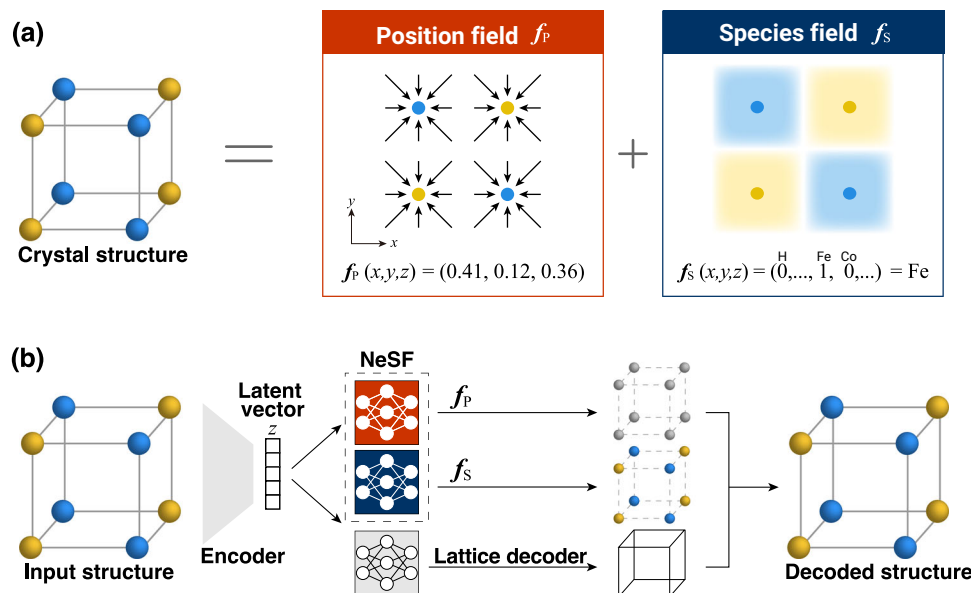
Fig. 1 Overview of the structure field and crystal structure autoencoder using the NeSF. a The structure field consists of two vector fields, namely, position field f_P and species field f_S, which are defined in 3D space. Given a 3D point as a query, the position field is trained to represent the 3D vector pointing to the nearest atom of the query. In addition, the species field is trained to represent the species of the nearest atom as a categorical probability distribution. **b** Network architecture of proposed crystal structure autoencoder using the NeSF. The input crystal structure is transformed into latent vector z via a PointNet-based encoder[35,36]. In the decoder using the NeSF, the position field and species field are queried to recover atomic positions and species, respectively, from material information given as z. The lattice constants are also estimated via the lattice decoder. These encoder and decoder are implemented as simple MLPs. See the Supplementary Note 5 for detailed definitions.

The proposed NeSF overcomes the limitations of voxelization. In the NeSF, there is essentially no tradeoff between the spatial resolution and required memory. Theoretically, the NeSF can achieve infinitely high spatial resolution with compact (memory- and parameter-efficient) neural networks in place of costly 3D CNNs. In addition, the NeSF can effectively represent arbitrary crystal structures, including those with elongated or distorted unit cells. Furthermore, the NeSF can directly provide the Cartesian coordinates of the atomic positions rather than peaks in a scalar field. We believe that the proposed NeSF will break through a technical bottleneck in MI approaches for crystal structure estimation and contribute to the advancement of MI research in this direction.

In the following section, we detail the proposed NeSF and demonstrate its expressive power for various crystal structures through numerical experiments. Notably, the NeSF successfully recovers various crystal structures, from the relatively basic structures of perovskite materials to the complex structures of cuprate superconductors. Results from extensive quantitative evaluations demonstrate that the NeSF outperforms the voxelization approach in ICSG3D[25].

Results and discussion
We first describe the procedures for estimating crystal structures with the NeSF and for training the NeSF. We then present an autoencoder of crystal structures as an application of the NeSF. In this autoencoder, crystal structures are embedded into vectors z (called *latent vectors*) via an encoder, and then the NeSF acts as a decoder to recover the input crystal structures from z. The performance of the NeSF-based autoencoder is quantitatively analyzed by evaluating the reconstruction accuracy, outperforming the voxelization-based ICSG3D baseline. Furthermore, we qualitatively analyze the space of vectors z learned by the proposed autoencoder. This analysis shows that the learned space reflects some similarity between crystal structures instead of merely embedding crystal structures at random.

Crystal structure estimation with NeSF. Given the target material information as vector z, the estimation of the crystal structure from z amounts to estimating the positions and species of the atoms in the unit cell along with the lattice constants. The lattice constants are modeled as lengths a, b, and c and angles α, β, and γ, and estimated via simple multilayer perceptrons (MLPs) with input z. On the other hand, the atomic positions and species are estimated by position field f_P and species field f_S of the NeSF, respectively, as described in the previous section. These fields are also implemented as simple MLPs, each taking query position p and vector z as input and predicting a field value (i.e., 3D pointing vector or categorical probability distribution). An overview of the NeSF network architecture is illustrated in the right part of Fig. 1b.

Given vector z from the encoder, the estimation of the atomic positions and species using the NeSF is illustrated in Fig. 2 and summarized in the five steps:

1. **Initialize particles**. We first estimate the lattice constants via MLPs. Then, we regularly spread initial query points $\{p_i^0\}$, which we call *particles*, at 3D grid points within a bounding box. The bounding box is common to each dataset and is given to loosely encompass the atoms of all training samples.

2. **Move particles**. We update the position of each particle p_i^t using the position field according to $p_i^{t+1} = p_i^t + f_P(p_i^t, z)$. We iterate this process for all particles to obtain $\{p_i\}$ as candidate atomic positions. Because the position field is expected to point to the nearest atom position, the particles travel toward their nearest atoms through this process.

3. **Score particles**. We score each particle p_i and filter outliers. As the norm of the output of the position field, $\|f_P(p_i, z)\|$, indicates an estimated distance from p_i to its nearest atom, we score each particle p_i by $\|f_P(p_i, z)\|$ and discard the particle if the score is above a certain threshold (set to 0.9 Å in this study).

Stage 1: Atom position estimation
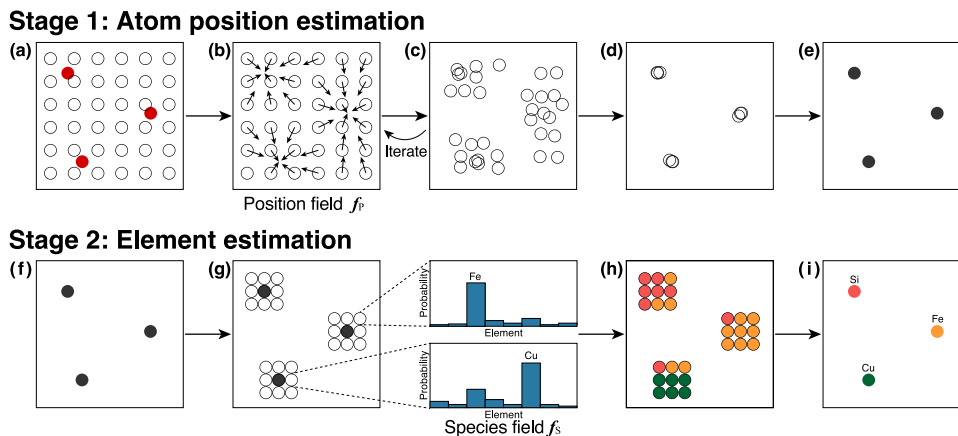

Stage 2: Element estimation

Fig. 2 Algorithm for crystal structure estimation with the NeSF. The algorithm consists of two stages: 1) atomic position estimation (corresponding to steps 1–4 in "Crystal structure estimation with NeSF") and 2) element estimation (corresponding to step 5 in "Crystal structure estimation with NeSF"). **a** To detect true atomic positions (shown as red dots), we regularly place query particles in the position field. **b**, **c** Then, we iteratively infer the vectors pointing to their nearest atoms and update the query particles until convergence. **d**, **e** We detect atoms by clustering the query particles. **f**, **g** For element estimation, we spread new query particles around each detected atom. **h** Then, we query the species field to obtain categorical distributions as votes for the atomic species. **i** We finally apply majority voting to determine the species of each atom.

4. **Detect atoms**. Until this point, the particles are expected to form clusters around atoms. Thus, we apply a simple clustering algorithm to detect each cluster position as an atomic position, determining the number of atoms in the crystal structure. Here we employ a well-known clustering algorithm in object detection, non-max suppression. Specifically, we initialize a list of candidate particles as $B_c = \{p_i\}$ and another list of accepted particles as $B_a = \{\}$. 1) We select the particle with the lowest score (i.e., estimated closest to an atom) from B_c and move it to B_a. 2) We remove particles from B_c that are within a spherical area around the selected particle. (Here, we used the spherical radius of 0.5 Å, and if the number of particles in the sphere is <10, we reject the candidate.) By repeating these steps until B_c becomes empty, we obtain atomic positions $\{a_i\}$ as the selected particles stored in B_a.

5. **Estimate species**. Finally, we use species field $f_s(p, z)$ to estimate the atomic species at each atomic position a_i. For robust estimation against errors in a_i, we spread new particles intensively around each a_i as queries to the species field instead of using a_i directly as a query. These query particles are sampled using a local regular grid centered at a_i. Hence, we obtain multiple probability distributions, each predicting the species of atom a_i. We select the most frequent atomic species among them as the final estimate.

Note that the presented algorithm runs deterministically without any random process involved. The hyperparameters used for sampling points in crystal structure estimation are provided in Supplementary Note 6. Sampling parameters in the Supplementary Information (SI). We also analyzed the sensitivity of these hyperparameters in Supplementary Note 3. Hyperparameters for crystal structure estimation in the SI.

Training of NeSF. The training of the NeSF differs from the above estimation algorithm and is much simpler. Specifically, we train the NeSF to directly predict the vectors pointing to the nearest atoms and categorical distributions indicating their species, without iterative position update. To do so, we randomly sample 3D query points $\{p_i^s\}$ in the unit cell and compute loss values for the field outputs at these points, thus supervising $f_p(p_i^s, z)$ and $f_s(p_i^s, z)$ to indicate the position and species of the nearest atom, respectively. However, we cannot densely sample

the query points because of practical limitations in memory usage. Therefore, a sampling strategy for query points is required for training. Existing implicit neural representations for 3D shape estimation, such as DeepSDF[30], sample the training query points near the surface. Curriculum DeepSDF[34] further introduces curriculum learning, in which the sampling density intensifies near the surface as training proceeds.

To consider a desired sampling strategy for training the position and species fields, we consider their dynamics in the proposed algorithm. (1) Particles iteratively move in the position field toward their nearest atoms. Thus, the position field should be sufficiently accurate everywhere to allow the flow of particles to their destinations and highly accurate in the vicinity of atoms. (2) The species field is queried only around atoms. Thus, it does not need to be accurate everywhere but should be robust to errors in the estimated atomic positions.

To meet the above-mentioned requirements, we introduce two sampling methods to train the position and species fields. (1) Global grid sampling: This method considers 3D grid points that uniformly cover the entire unit cell and samples the points with perturbations that follow a Gaussian distribution. (2) Local grid sampling: This method considers local 3D grid points centred at each atomic position and samples the points with perturbations that follow a Gaussian distribution.

To train the position field, we combine both sampling methods. Hence, query points are sampled uniformly over the entire unit cell and densely around the atoms. To train the species field, we use local grid sampling to concentrate training query points in the neighbourhood of atoms. The parameters used for sampling are provided in Supplementary Note 6. Sampling parameters in the SI.

Crystal structure autoencoder. To demonstrate and evaluate the expressive power of the NeSF, we propose an autoencoder of crystal structures. Similar to other common autoencoders, the proposed NeSF-based autoencoder consists of an encoder and decoder. The encoder is a neural network that transforms an input crystal structure (i.e., positions and species of the atoms in the unit cell and lattice constants) into abstract latent vector z. The decoder, for which we use the NeSF, reconstructs the input crystal structure from the latent vector z. Autoencoders are typically used to learn latent vector representations of data via

self-supervised learning, in which the input data can supervise the learning via a reconstruction loss.

While we focus on decoding crystal structures, their encoding has been studied in MI. Because a crystal structure is essentially a set of atoms, its encoding must handle a variable number of atoms with invariance to permutation. In ML, such encoders are generally called *set functions*[35,36]. Among them, the family of graph neural networks[11,17,18,28] serves as popular crystal-structure encoders. However, these networks implicitly represent atomic positions as edges, encoding distances between atoms while discarding the exact coordinates. Although this distance-based graph representation is key to ensuring the invariance of coordinate systems, its information loss in input may unintentionally hinder the reconstruction performance. Thus, we adopt the basic encoder architecture from PointNet[35] and DeepSets[36]. This architecture not only represents the simplest type of set-function-based networks but can also preserve the information of input crystal structures, thus being appropriate for the performance evaluation of the NeSF decoder. The architecture of the proposed autoencoder is detailed in "Neural network architecture" and in Supplementary Note 5. Network architecture in the SI.

Training and evaluation procedures. We trained and evaluated the autoencoder and ICSG3D[25] (baseline) on three materials datasets: ICSG3D, limited cell size 6 Å (LCS6Å), and YBCO-like datasets. These datasets collect the crystal structures of materials from Materials Project[37] and are designed to have different difficulty levels. The ICSG3D dataset is a materials collection used by Court et al.[25] and contains 7897 materials with limited crystal systems (i.e., cubic) and prototypes (i.e., AB, ABX2, and ABX3), and it is intended to be the easiest among the three datasets. The LCS6Å dataset consists of 6005 materials with unit cell sizes of 6 Å or less along the x, y, and z axes and without restrictions on the crystal systems and prototypes. The YBCO-like dataset consists of 100 materials with narrow unit cells along the c axis. These structures typically include those of yttrium barium copper oxide (YBCO) superconductors. Owing to the complexity of the structures and relatively few samples, the YBCO-like dataset is the most challenging among the three evaluated datasets. Further details on these datasets are provided in "Datasets".

For training and evaluation, we randomly split each dataset into training (90.25%), validation (4.75%), and test (5%) sets. The training set was only used to train the ML models. The validation set was used to preliminarily validate the trained ML model, and the test set was used to compute the final evaluation scores after training, validation, and hyperparameter tuning. The hyperparameters were tuned based on the validation scores from the LCS6Å dataset. To reduce the performance variation owing to randomness (e.g., randomness in the initialization of network weights), we repeated training and evaluation 10 times with different random seeds and evaluated the performance using the mean and standard deviation of the scores. Because the YBCO-like dataset has only 100 samples, it was treated slightly differently from the other two evaluated datasets. To reduce the performance variation owing to data splitting, we adopted twentyfold cross-validation for the YBCO-like dataset and performed one trial (instead of repeating 10 times with different random seeds). The iterative training of the neural networks was conducted using stochastic gradient descent with Adam[38] as the optimizer. Detailed training procedures, including the loss function definition, are provided in "Training".

The reconstruction performance was measured in terms of errors in the number of atoms, position, and species. The error in the number of atoms is the rate of materials for which the number of atoms in the unit cell is incorrectly estimated. The position error is the average error of the reconstructed atomic positions. Depending on the denominator of the metric, we evaluated the position error in two ways. An *actual* metric was used to evaluate the mean position errors at the actual atomic sites of the crystal structure by computing their shortest distances to estimated atomic sites. By contrast, a *detected* metric was used to evaluate the errors at the estimated sites by computing their shortest distances to the actual atomic sites. Formally, given sets of actual and detected atomic positions as P_{actual} and $P_{detected}$, the actual and detected metrics are provided as $m(P_{actual}, P_{detected})$ and $m(P_{detected}, P_{actual})$, respectively, with function $m(A, B) = \sum_{\boldsymbol{a} \in A} \min_{\boldsymbol{b} \in B} \| \boldsymbol{a} - \boldsymbol{b} \| / |A|$. The actual metric is more sensitive to errors related to underestimation of the number of atoms, whereas the detected metric is more sensitive to errors related to overestimation. The species error is the average rate of atoms with incorrectly estimated species. Analogous to the position error, the species error was evaluated using actual and detected metrics. Lower values of these metrics indicate better performance.

Quantitative performance comparisons with ICSG3D. Table 1 lists the reconstruction errors of the proposed NeSF-based autoencoder and ICSG3D baseline on the test sets of the three datasets. We also provide the scores for the training and validation sets in Supplementary Note 1. Reconstruction results for the training and validation splits in the SI. Overall, the proposed method consistently outperforms ICSG3D in all the evaluation metrics, with substantial performance improvements for the species error in all datasets and for all the metrics on the YBCO-like dataset. Figure 3 shows crystal structures from the three evaluated datasets, comparing test samples and reconstruction results by the proposed autoencoder and ICSG3D.

For the ICSG3D dataset, which is the easiest of the three datasets, ICSG3D achieves good performance for the error in number of atoms and position error, but provides high species error (~65% in both actual and detected metrics). By contrast, the proposed method achieves slightly better position error and error in a number of atoms and drastically lower species error (approximately 4%). We believe that this is because ICSG3D estimates the atomic species via electron density maps, whereas the proposed method more directly represents atomic species as categorical distributions. Extending ICSG3D to estimate a categorical distribution at each voxel is impractical because it would require approximately 100×32^3 times the memory usage in the output (*i.e.*, requiring 100 species categories for each of the 32^3 voxels in addition to one electron density map).

For the LCS6Å and YBCO-like datasets, which are more challenging than the ICSG3D dataset, the performance advantage of the proposed method is even clearer, especially regarding the position error. The LCS6Å dataset contains a variety of crystal structures (e.g., non-cubic structures, distorted crystal structures), whereas the YBCO-like dataset contains very narrow crystal structures. In addition, the YBCO-like dataset contains few samples, possibly leading to model overfitting (i.e., the performance on the test set is likely to degrade significantly). Despite these difficulties, the proposed method can accurately estimate the atomic positions and species.

For a detailed analysis of the relationship between the method performance and structural complexity, Fig. 4 shows the distributions of reconstruction errors according to the number of atoms given as medians (points) and 68% ranges around them (colored regions) over 10 test trials on materials from the ICSG3D and LCS6Å datasets. The YBCO-like dataset is excluded from this analysis because it contains only materials with 13

Table 1 Reconstruction results (test set).

Method Dataset	Proposed ICSG3D	ICSG3D	Proposed LCS6Å	ICSG3D	Proposed YBCO-like	ICSG3D
Error in number of atoms [%]	0.53 ± 0.25	2.67 ± 0.84	6.35 ± 0.50	17.28 ± 1.13	12.00	91.00
Position error (actual) [Å]	0.0308 ± 0.0112	0.0877 ± 0.0306	0.1161 ± 0.0073	0.2006 ± 0.0132	0.2631	(0.631l)
Position error (detected) [Å]	0.0359 ± 0.0226	0.1057 ± 0.0284	0.1632 ± 0.0123	0.2886 ± 0.0159	0.2448	0.4358
Species error (actual) [%]	4.31 ± 0.39	64.39 ± 1.91	14.78 ± 0.81	55.70 ± 1.40	14.78	78.08
Species error (detected) [%]	4.36 ± 0.39	65.05 ± 1.85	16.11 ± 0.66	58.32 ± 1.45	19.89	54.24
Lattice length error [Å]	0.27 ± 0.74	0.08 ± 0.04	0.05 ± 0.00	0.06 ± 0.01	0.25	0.10
Lattice angle error [deg]	0.37 ± 1.10	0.00 ± 0.00	0.19 ± 0.04	0.34 ± 0.07	2.91	0.06

We compared the performance of the proposed NeSF-based crystal structure autoencoder and ICSG3D method for crystal structure reconstruction on three datasets (ICSG3D, LCS6Å, and YBCO-like). The position and species errors were evaluated in two ways based on the actual or detected atomic sites. For the YBCO-like dataset, ICSG3D failed to output any atom for 44 out of the 100 materials and provided no valid score for the position error (actual). We computed a score by excluding those incorrectly estimated materials (shown in parentheses), indicating that the actual performance of ICSG3D is worse than the displayed value.

atoms in their unit cells. Figure 4a, b shows the signed errors between the numbers of detected and actual atoms. These results indicate that both methods correctly estimate the number of atoms for most (i.e., >68%) of the samples in the ICSG3D dataset (Fig. 4a). However, the ICSG3D method underestimates the number of atoms for the LCS6Å dataset (Fig. 4b). Likewise, Fig. 4c, d show the distributions of position errors and Fig. 4e, f show the distributions of species errors. Because the number of atoms is either correctly estimated or underestimated by both methods in most cases, we report the errors in the actual metrics. Examining the distributions at $x = 2$ in Fig. 4e, we can determine the trends of species errors for the diatomic structures in the ICSG3D dataset. The proposed method provides the correct species of both atoms for >68% of the diatomic materials, whereas ICSG3D often misestimates the species of one of the two atoms.

Overall, although the three types of errors by both methods tend to increase with the number of atoms, our method outperforms ICSG3D consistently for materials with varying numbers of atoms. Compared with our method, the performance of ICSG3D tends to degrade more notably for materials with many atoms. Because ICSG3D often underestimates the number of atoms (Fig. 4b), this overall degradation trend suggests that ICSG3D cannot capture those many-atom structures owing to its spatial resolution limited to $32 \times 32 \times 32$ voxels.

We believe that the notable high performance of the proposed method is attributable to two reasons. First, our method does not use discretization, which is advantageous over the grid-based ICSG3D for estimating complex crystal structures. In a grid-based method, the spatial resolution is limited by the cubically increasing computations and memory usage. The proposed NeSF is free from such tradeoff between resolution and computational complexity, and it can thus effectively represent complex structures. Second, the model size of the proposed NeSF using MLPs is much smaller than that of the 3D CNN architecture of ICSG3D. In general, the number of training samples required for an ML model is correlated with the number of trainable parameters. Grid-based methods use layers of 3D convolution filters that involve many trainable parameters. By contrast, the NeSF employs implicit neural representations to describe the 3D space indirectly as a field instead of voxels. Thus, it is efficiently implemented by MLPs with fewer parameters than a 3D CNN. Specifically, the NeSF-based autoencoder has 0.76 million parameters, which is only 2.24% of the number of parameters in the 3D CNN-based ICSG3D (34 million parameters). This difference can make the NeSF advantageous over grid-based methods, especially on small datasets such as the YBCO-like dataset.

Latent space interpolation. We qualitatively analyzed the characteristics of the proposed NeSF-based autoencoder by inspecting the learned latent space of the crystal structures. In general, a good latent space should map similar items (in terms of properties, characteristics, categories, etc.) closely in the space, thus providing latent data representations that facilitate analysis by humans and machines. To assess the construction of the latent space of crystal structures, we visualized transitions in the latent space as sequences of crystal structures. If the latent space is trained to capture relationships between materials in terms of structural similarity, interpolating between two points in the latent space should produce a sequence of materials with similar crystal structures.

Interpolation analysis proceeded as follows:

1. Select two known crystal structures from the ICSG3D test set as the source and destination materials and obtain their latent vectors as z_{src} and z_{dst} via the trained encoder.
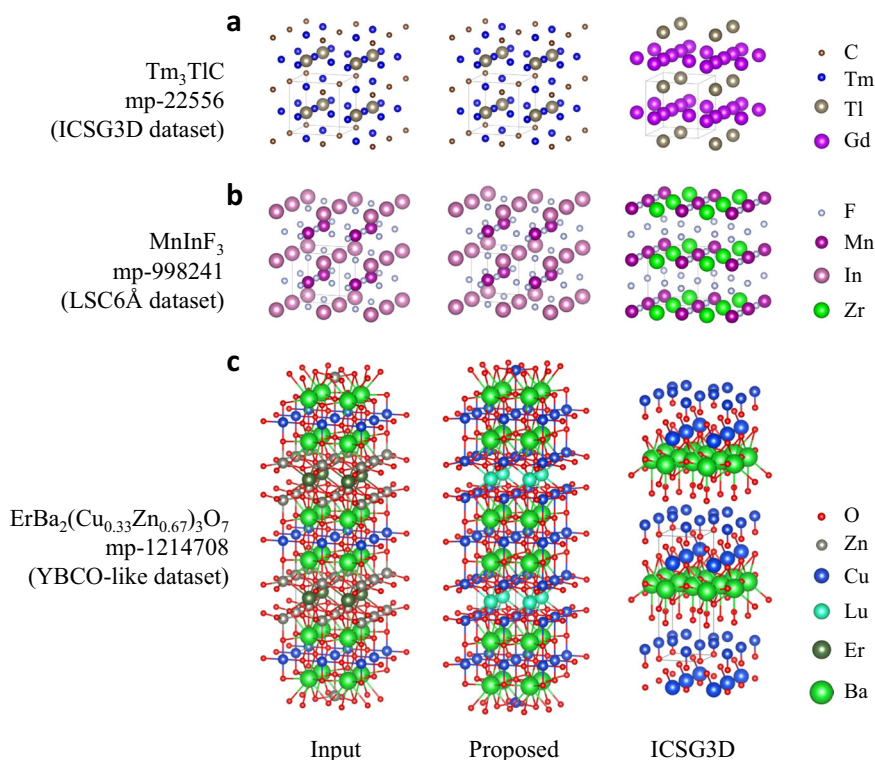
Fig. 3 Reconstruction examples. For the three evaluated datasets (rows), we show test samples (first column) and reconstruction results by the proposed (second column) and ICSG3D (third column) methods. **a** Tm_3TlC (mp-22556, ICSG3D dataset). **b** $MnInF_3$ (mp-998241, LCS6Å dataset). **c** $ErBa_2(Cu_{0.33}Zn_{0.67})_3O_7$ (mp-1214708, YBCO-like dataset).

2. Interpolate between z_{src} and z_{dst} linearly in the latent space to obtain a sequence of latent vectors.
3. Decode each latent vector via the trained decoder (NeSF) to obtain its crystal structure.

Because the intermediate crystal structures between the source and destination are reconstructed from the latent vectors via the trained decoder, these structures may not appear in the dataset.

To facilitate the interpretation of the analysis, we chose the well-known zinc-blende and rock-salt structure families as benchmark materials. Both families have compositions given by AX, where A is a cation, and X is an anion, and the crystal structure is based on a cubic crystal system. Therefore, if the source and destination materials belong to one of these families, the characteristic composition and structural prototype should be preserved throughout the interpolation path.

As a first example, Fig. 5 shows the results of interpolation from ZnS (mp-10695) to CdS (mp-2469). The obtained transition in the compositional formula is $ZnS \rightarrow MgZn_3S_4$ $(Mg_{0.25}Zn_{0.75}S) \rightarrow MgZnS_2$ $(Mg_{0.5}Zn_{0.5}S) \rightarrow Mg_3ZnS_4$ $(Mg_{0.75}Zn_{0.25}S) \rightarrow MgS \rightarrow MgCd_3S_4$ $(Mg_{0.25}Cd_{0.75}S) \rightarrow CdS$.

As a second example, Fig. 6 shows the results of interpolation from MgO (mp-1265) to NaCl (mp-22862). The obtained transition in the compositional formula is $MgO \rightarrow NaMgO_2$ $(Na_{0.5}Mg_{0.5}O) \rightarrow NaO \rightarrow Na_2ClO$ $(NaCl_{0.5}O_{0.5}) \rightarrow Na_4Cl_3O$ $(NaCl_{0.75}O_{0.25}) \rightarrow NaCl$.

Additionally, we provide the results of interpolation from NaCl (mp-22862) to PbS, from MgO (mp-1265) to CaO (mp-2605), from PbS (mp-21276) to CaO (mp-2605), and from BaTiO_3 (cubic; mp-2998) to BaTiO_3 (tetragonal; mp-5986) in Supplementary Note 2. Additional results of latent space interpolation in the SI.

In the interpolation examples, the composition AX and the cubic structure are mostly preserved. Furthermore, the compositions change continuously without collapsing along the

interpolation paths. These results suggest that our encoder learns meaningful continuous representations of crystal structures by capturing their characteristics in an abstract space, and the proposed NeSF model successfully decodes these representations into crystal structures.

Latent space visualization. As another qualitative analysis of the latent space, we visualized the space using t-distributed stochastic neighbor embedding (t-SNE)[39]. While the interpolation analysis in the previous section examines the local smoothness of the space, this t-SNE analysis inspects the space at a more global scale to see if it maps similar structures closely, or as clusters. In Fig. 7, we have embedded the 192-dimensional latent vectors of the materials from the LCS6Å dataset into two-dimensional (2D) plots via t-SNE. Each plot is colored according to the cell volume (Fig. 7a), the number of atoms (Fig. 7b), and space group (Fig. 7c) as structural attributes of the material. These plots form clusters whose colors are well separated or smoothly transitioned, showing that the latent space reflects the structural similarity of the materials.

Limitations and future directions. This study was mainly focused on developing a fundamental approach for crystal structure estimation using implicit neural representations. To suggest room for further improvement and important directions in future work, we identified three main limitations of the NeSF.

First, the proposed NeSF does not explicitly consider space-group symmetry. Thus, the local spatial arrangements of atoms in conventional unit cells estimated by the NeSF do not necessarily obey space-group symmetry. Although ICSG3D[25] shares the same limitation, symmetry is an important concept in crystallography. Thus, incorporating the constraint of space-group symmetry into the NeSF is an important direction of future work.
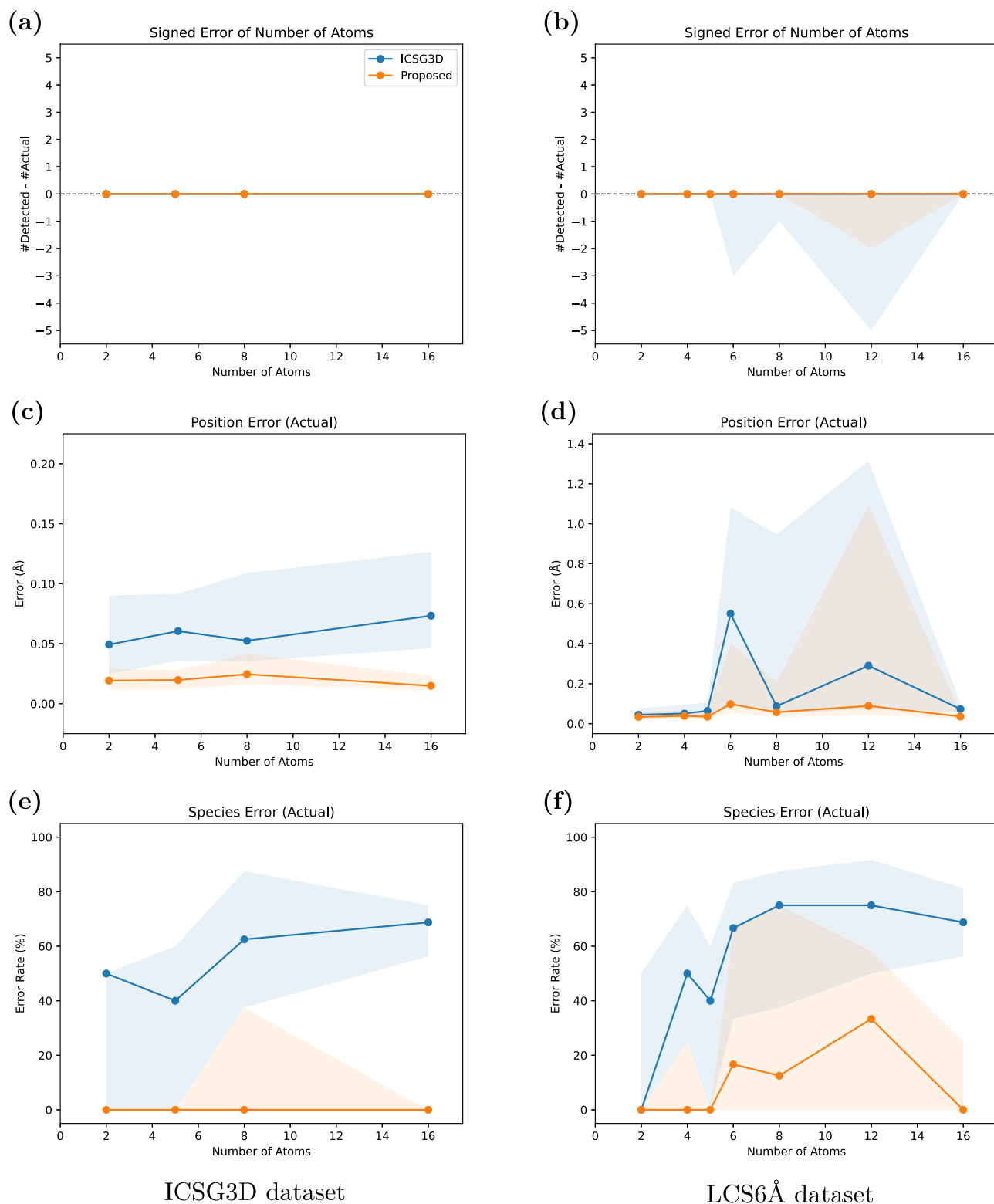
Fig. 4 Relationships between reconstruction performance and number of atoms in conventional unit cell. Three kinds of error distributions (rows) are shown for the ICSG3D (first column) and LCS6Å (second column) datasets described by medians (points) and 68% ranges (colored regions) similar to means ± σ ranges. **a**, **b** Distributions of signed errors between detected and actual numbers of atoms. **c**, **d** Distributions of position errors according to a number of atoms averaged over individual structures. **e**, **f** Distributions of species errors according to number of atoms averaged over individual structures. For statistical reliability, we omit the x axis points counting <10 materials samples from the visualization (see "Datasets" for detailed dataset statistics).

Second, for evaluation purposes, we adopted the autoencoder architecture rather than generative models, such as variational autoencoders[17,26,40] and generative adversarial networks[27,40–42]. These generative models intentionally perturb latent structural

representations to produce diverse structures that do not appear in the dataset. While this aspect of the generative models is more suitable for novel structure discovery, the lack of ground-truth structures prevents quantitative and reliable performance
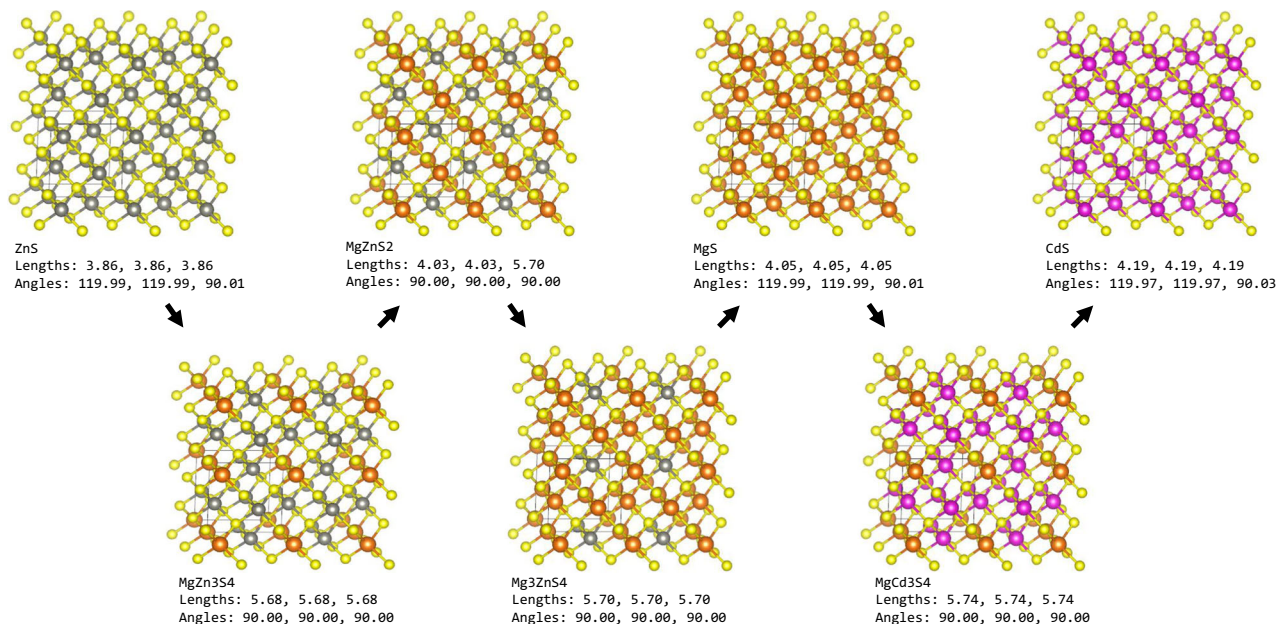
Fig. 5 Results of interpolation from ZnS to CdS. To verify the smoothness of the latent vector space, we examine an interpolation of the latent vectors. It shows that the latent space is smoothly interpolated into two structures.
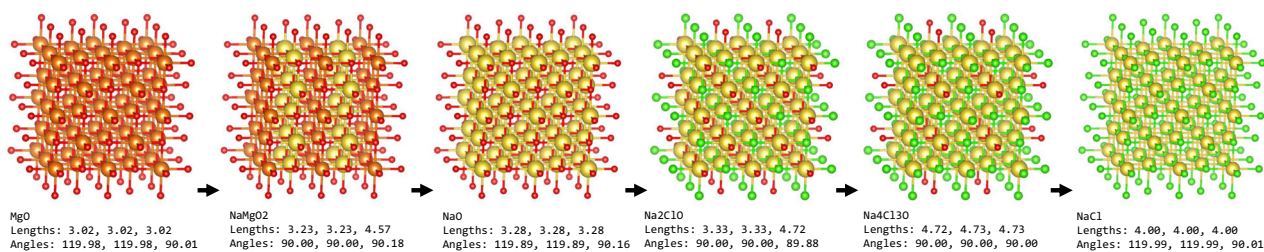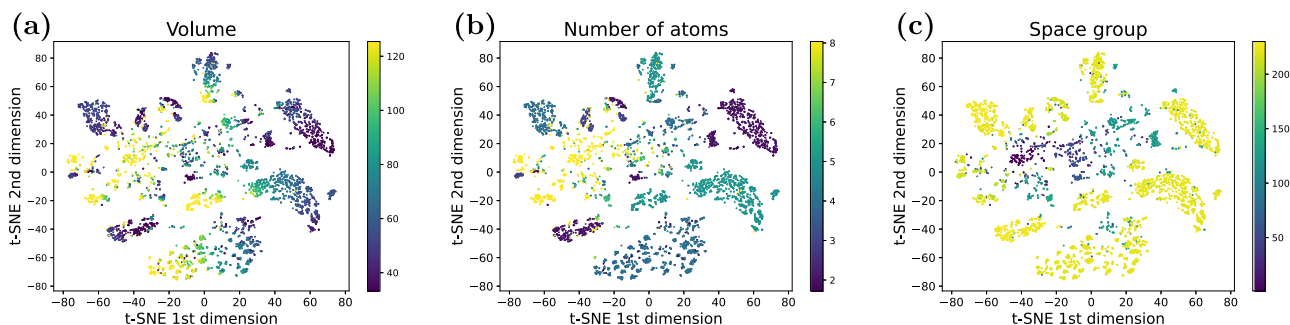

Fig. 6 Results of interpolation from MgO to NaCl. To verify the smoothness of the latent vector space, we examine an interpolation of the latent vectors. It shows that the latent space is smoothly interpolated into two structures.

Fig. 7 t-SNE visualizations of the latent space. Each 2D plot represents the latent vector of material from the LCS6Å dataset. The plot colors visualize **a** cell volume, **b** number of atoms in the cell, and **c** space group as structural attributes of the materials.

analysis. The NeSF should be applied to generative models in future work with appropriate performance analysis.

Third, the trained latent vectors have limited capability as material descriptors for property prediction, as shown in an additional analysis provided in Supplementary Note 4. Prediction of material property in the SI. Although this result is understandable since the latent space is not directly trained for property prediction, learning a latent space that effectively disentangles material properties from structural features is another future direction. Such a latent space will be useful not only for property prediction but also for the inverse design of materials with desired

properties when combined with the aforementioned generative approach.

Conclusion

We propose the NeSF to estimate crystal structures using neural networks. Determining crystal structures directly using neural networks is challenging because these structures are essentially represented as an unordered set with varying numbers of atoms. NeSF overcomes this difficulty by treating the crystal structure as a continuous vector field rather than as a discrete set of atoms.

We borrow the idea of the NeSF from vector fields in physics and the recent implicit neural representations in computer vision. An implicit neural representation is an ML technique to represent 3D geometries using neural networks. The NeSF extends this technique by introducing the position and species fields to estimate the atomic positions and species of crystal structures, respectively. Unlike existing grid-based approaches for representing crystal structures, the NeSF is free from the tradeoff between spatial resolution and computational complexity and can represent any crystal structure.

The NeSF was applied as an autoencoder for crystal structures and demonstrated its performance and expressive power on datasets with diverse crystal structures. Quantitative performance analysis showed a clear advantage of the NeSF-based autoencoder over an existing grid-based method, especially for estimating complex crystal structures. Furthermore, a qualitative analysis of the learned latent space revealed that the autoencoder captures similarities between crystal structures rather than mapping crystal structures randomly.

In materials science, the design and construction of crystal structures are fundamental processes when searching for materials with the desired properties. ML is advancing rapidly with the development of neural networks, and representing arbitrary crystal structures using those networks is essential for the next-generation development of materials. For instance, the NeSF can be applied to powerful deep generative models, such as variational autoencoders and generative adversarial networks, to discover novel crystal structures. Such generative models for crystal structures will be important for the inverse design of materials, which is a major challenge in MI. We believe that the NeSF can overcome the technical bottleneck of ML for crystal structure estimation and pave the way for the next-generation development of materials.

Methods
Datasets. We evaluated the proposed NeSF-based autoencoder using datasets from the Materials Project[37].

- ICSG3D dataset: To compare with the ICSG3D baseline, we reproduced the dataset used in the original paper[25]. Unfortunately, the proponents of ICSG3D did not specify the identifiers of the materials used for their experiments. Therefore, by following the procedures described in their paper, we crawled the three material classes with cubic structures, namely, binary alloys (AB), ternary perovskites (ABX_3), and Heusler compounds (ABX_2), from the Materials Project[37]. Then, we split the dataset into training (7194 samples), validation (342 samples), and test

(360 samples) sets. The numbers of atoms per unit cell are listed in Table 2.
- LCS6Å dataset: From Materials Project[37], we collected all the materials samples with unit cell sizes of 6 Å or less along the x, y, and z axes. Unlike the ICSG3D dataset, the LCS6Å dataset has no restriction on material classes and prototypes and covers 4.31% of the Materials Project database. This dataset was also split into training (5418 samples), validation (286 samples), and test (301 samples) sets, and the numbers of atoms in the samples are listed in Table 3.
- YBCO-like dataset: From the Materials Project[37], we extracted 100 samples with structures similar to YBCO ($YBa_2Cu_3O_7$), that is, narrow crystal structures along the c axis. The structures of these materials have narrow and anisotropic unit cells and contain various elements. Given the complexity of the structures and the limited number of samples, this dataset is considered the hardest and most practical among the three evaluated datasets. This dataset was also split into training (90 samples), validation (5 samples), and test (5 samples) sets. The numbers of atoms in the samples are listed in Table 4.

All the crystal structures were described by a conventional cell for the input and output of the autoencoders.

Neural-network architecture. The proposed NeSF-based autoencoder consists of simple MLPs. Specifically, the encoder first transforms the atomic position and species of each atom into a 512-dimensional feature vector. These feature vectors per atom are then aggregated into a single feature vector via a max pooling and converted into a 192-dimensional feature vector, which represents latent vector $\mathbf{z}$ of the input crystal structure. Given $\mathbf{z}$ as the input, the decoder estimates four types of structural information features using four separate MLPs. The atomic positions are estimated via the position field implemented as an MLP with nine fully connected layers. The atomic species are estimated via the species field implemented as an MLP with three fully connected layers. Lattice lengths (a, b, c) and angles (α, β, γ) are estimated using the corresponding MLPs with two fully connected layers. More details on the architecture are provided in Supplementary Note 5. Network architecture in the SI.

Training. During training, the four types of outputs from the decoder (i.e., atomic position, species, lattice lengths, and angles) were evaluated using the following loss function:

$$L = \lambda_{pos} L_{pos} + \lambda_{spe} L_{spe} + \lambda_{len} L_{len} + \lambda_{ang} L_{ang} \qquad (2)$$

where L_{pos} is the mean squared error between the estimated and true atomic positions, L_{spe} is the cross-entropy loss function for the atomic species distributions, and L_{len} and L_{ang} are the mean squared error of the lattice lengths and angles, respectively. The total loss function, L, is given by the weighted sum of these loss functions with weights (λ_{pos}, λ_{spe}, λ_{len}, λ_{ang}) = (10, 0.1, 1, 1). We optimized the loss function using stochastic gradient descent with a batch size of 128. We used Adam as the optimizer with an initial learning rate of 10^{-3}, decaying every 640 epochs by a factor of 0.5. For each dataset, we conducted iterative training for 3200

Table 2 Numbers of atoms per unit cell in ICSG3D dataset.

No. atoms per cell	2	5	6	8	12	15	16	20	32	40
Training	484	1413	7	572	1	12	4616	17	18	54
Validation	18	66		32			223	1		2
Test	24	61		34			237			4

Table 3 Numbers of atoms per unit cell in LCS6Å dataset.

No. atoms per cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	20	21	22	24	28
Training	21	599	44	1411	1325	244	69	686	58	137	9	352	9	24	4	402	5	6	1	2	9	1
Validation		35	2	80	81	14	3	41	4	5		13		2		20					1	
Test		30	3	83	68	17	2	35		5		19		1		23						

Table 4 Numbers of atoms per unit cell in YBCO-like dataset.

No. atoms per cell	13
Training	90
Validation	5
Test	5

epochs for all the material samples in the dataset. We applied early stopping based on the validation score for both the proposed method and ICSG3D, except for the YBCO-like dataset, for which the few validation samples may lead to unreliable validation scores. The training of the NeSF model took approximately 11 hours on the ICSG3D dataset, 9 hours on the LCS6Å dataset, and 1 hour on the YBCO-like dataset using a computer equipped with a single Quadro RTX8000 graphics processor. For details regarding our strategies for validating the trained models and tuning the hyperparameters (e.g., latent dimensionality, number of network layers, training batch size), see Supplementary Note 7. Hyperparameter search in the SI.

Data availability

The dataset generation procedure is available in Supplementary Note 1. Reconstruction results for the training and validation splits in the SI. Datasets will be available online here: https://github.com/omron-sinicx/neural-structure-field and also available from the corresponding author upon reasonable request.

Code availability

The implementation details are available in Supplementary Note 5. Network architecture in the SI. The code will be available online here: https://github.com/omron-sinicx/neural-structure-field and also available from the corresponding author upon reasonable request.

References

1. De Graef, M. & McHenry, M. E. Structure of materials: an introduction to crystallography, diffraction and symmetry. an introduction to crystallography, diffraction and symmetry (Cambridge University Press, 2012).
2. Callister, W. D. & Rethwisch, D. G. Materials Science and Engineering (John Wiley and Sons, 2010).
3. Lookman, T., Alexander, F. J. & Rajan, K. Information Science for Materials Discovery and Design (Springer, 2015).
4. Schmidt, J., Marques, M. R. G., Botti, S. & Marques, M. A. L. Recent advances and applications of machine learning in solid-state materials science. npj Comput. Mater. 5, 1–36 (2019).
5. Szymanski, N. J. et al. Toward autonomous design and synthesis of novel inorganic materials. Mater. Horiz. 8, 2169–2198 (2021).
6. Choudhary, K. et al. Recent advances and applications of deep learning methods in materials science. npj Comput. Mater. 8, 1–26 (2022).
7. Agrawal, A. & Choudhary, A. Perspective: materials informatics and big data: realization of the "fourth paradigm" of science in materials science. APL Mater. 4, 053208 (2016).
8. Suzuki, Y., Taniai, T., Saito, K., Ushiku, Y. & Ono, K. Self-supervised learning of materials concepts from crystal structures via deep neural networks. Mach. Learn: Sci. Technol. 3, 045034 (2022).
9. Jha, D. et al. ElemNet: deep learning the chemistry of materials from only elemental composition. Sci. Rep. 8, 17593 (2018).
10. Goodall, R. E. A. & Lee, A. A. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. Nat. Commun. 11, 6280 (2020).
11. Park, C. W. & Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. Phys. Rev. Mater. 4, 063801 (2020).
12. Wang, A. Y.-T., Kauwe, S. K., Murdock, R. J. & Sparks, T. D. Compositionally restricted attention-based network for materials property predictions. npj Comput. Mater. 7, 1–10 (2021).
13. Park, W. B. et al. Classification of crystal structure using a convolutional neural network. IUCrJ 4, 486–494 (2017).
14. Oviedo, F. et al. Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. npj Computat. Mater. 5, 60 (2019).
15. Szymanski, N. J., Bartel, C. J., Zeng, Y., Tu, Q. & Ceder, G. Probabilistic deep learning approach to automate the interpretation of multi-phase diffraction spectra. Chem. Mater. 33, 4204–4215 (2021).
16. Tshitoyan, V. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. Nature 571, 95–98 (2019).
17. Xie, T. & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Phys. Rev. Let. 120, 145301 (2018).
18. Cheng, J., Zhang, C. & Dong, L. A geometric-information-enhanced crystal graph network for predicting properties of materials. Commun. Mater. 2, 1–11 (2021).
19. Choudhary, K. & DeCost, B. Atomistic Line Graph Neural Network for improved materials property predictions. npj Comput. Mater. 7, 1–8 (2021).
20. Noh, J. et al. Inverse design of solid-state materials via a continuous representation. Matter 1, 1370–1384 (2019).
21. Noh, J., Gu, G. H., Kim, S. & Jung, Y. Machine-enabled inverse design of inorganic solid materials: promises and challenges. Chem. Sci. 11, 4871–4881 (2020).
22. Yao, Z. et al. Inverse design of nanoporous crystalline reticular materials with deep generative models. Nat. Mach. Intell. 3, 76–86 (2021).
23. Long, T. et al. Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. npj Comput. Mater. 7, 1–7 (2021).
24. Fung, V., Zhang, J., Hu, G., Ganesh, P. & Sumpter, B. G. Inverse design of two-dimensional materials with invertible neural networks. npj Comput. Mater. 7, 1–9 (2021).
25. Court, C. J., Yildirim, B., Jain, A. & Cole, J. M. 3-D inorganic crystal structure generation and property prediction via representation learning. J. Chem. Inf. Model. 60, 4518–4535 (2020).
26. Hoffmann, J. et al. Data-driven approach to encoding and decoding 3-d crystal structures. arXiv https://arxiv.org/abs/1909.00949# (2019).
27. Kim, S., Noh, J., Gu, G. H., Aspuru-Guzik, A. & Jung, Y. Generative adversarial networks for crystal structure prediction. ACS Cent. Sci. 6, 1412–1420 (2020).
28. Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 31, 3564–3572 (2019).
29. Mildenhall, B. et al. NeRF: representing scenes as neural radiance fields for view synthesis. Commun. ACM 65, 99–106 (2021).
30. Park, J. J., Florence, P., Straub, J., Newcombe, R. & Lovegrove, S. Deepsdf: learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019).
31. Chen, Z. & Zhang, H. Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019).
32. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. & Geiger, A. Occupancy networks: learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019).
33. Xie, Y. et al. Neural fields in visual computing and beyond. Computer Graphics Forum https://arxiv.org/abs/2111.11426 (2022).
34. Duan, Y. et al. Curriculum deepsdf. In: Proceedings of European Conference on Computer Vision, 51–67 (2020).
35. Charles, R. Q., Su, H., Kaichun, M. & Guibas, L. J. Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017).
36. Zaheer, M. et al. Deep sets. In: Proceedings of the Neural Information Processing Systems (2017).
37. Jain, A. et al. Commentary: The Materials Project: a materials genome approach to accelerating materials innovation. APL Materials 1, 011002 (2013).
38. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In: Proceedings of The International Conference on Learning Representations (2015).
39. van der Maaten, L. & Hinton, G. Visualizing data using t-sne. J. Mach. Learn. Res. 9, 2579–2605 (2008).
40. Goodfellow, I. J., Bengio, Y. & Courville, A. Deep Learning (MIT Press, 2016).
41. Goodfellow, I. et al. Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. (eds.) Proceedings of the Neural Information Processing Systems, vol. 27 (Curran Associates, Inc., 2014).

42. Nouira, A., Sokolovska, N. & Crivello, J.-C. CrystalGAN: learning to discover crystallographic structures with generative adversarial networks. *arXiv* https://arxiv.org/abs/1810.11203 (2019).

Author contributions

N.C. conceived the idea for the present work, did most of the implementation, and performed the numerical experiments. Y.S. conceived the idea for the present work and performed the numerical experiments. T.T. assisted in the implementation of the data analysis and discussed the results from a machine-learning perspective. R.I. reviewed the implementation and discussed the results from a materials science perspective. Y.U. initiated the idea of decoding crystal structures as point clouds directed the project and discussed the results from a machine-learning perspective. K.S. helped with the material data handling and discussed the results from a materials science perspective. K.O. directed the project and discussed the results from a materials science perspective. All authors discussed the results and wrote the manuscript together.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43246-023-00432-w.

Correspondence and requests for materials should be addressed to Kanta Ono.

Peer review information *Communications Materials* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Aldo Isidori. A peer review file is available.

Reprints and permission information is available at http://www.nature.com/reprints

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