(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 120188167 A
(43)申请公布日 2025.06.20

(21)申请号 202380078286.2

(22)申请日 2023.11.14

(30)优先权数据
2022-186043 2022.11.21 JP
2023-130520 2023.08.09 JP

(85)PCT国际申请进入国家阶段日
2025.05.09

(86)PCT国际申请的申请数据
PCT/JP2023/040970 2023.11.14

(87)PCT国际申请的公布数据
WO2024/111471 JA 2024.05.30

(71)申请人 欧姆龙株式会社
地址 日本国京都府京都市
申请人 国立大学法人大阪大学

(72)发明人 千叶直也 牛久祥孝 谷合龙典
五十岚亮 小野宽太 铃木雄太
齐藤耕太郎

(74)专利代理机构 北京三友知识产权代理有限公司 11127
专利代理师 章琴 邓毅

(51)Int.Cl.
G06N 3/0455(2006.01)
G16C 20/70(2006.01)
G16C 20/80(2006.01)

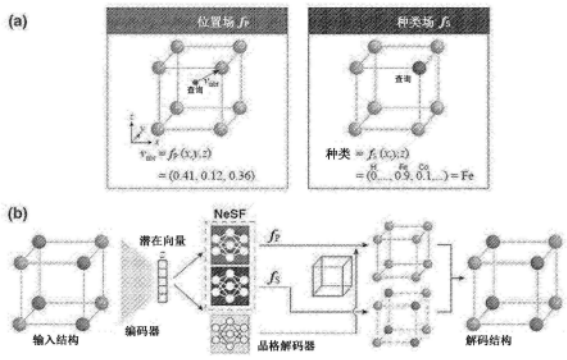
权利要求书3页 说明书19页 附图29页

(54)发明名称

信息处理装置、已学习模型生成装置、信息处理方法、已学习模型生成方法、信息处理程序和已学习模型生成程序

(57)摘要

利用神经网络模型,表征表示由原子点群表示的物质结构的场。



CN 120188167 A

1.一种信息处理装置,其中,该信息处理装置包含处理部,所述处理部利用神经网络模型,表征表示由原子点群构成的物质结构的场。

2.根据权利要求1所述的信息处理装置,其中,

该信息处理装置还包含取得部,所述取得部取得对象向量和物质内的关注点即查询点,

所述神经网络模型是在被输入任意的向量和所述查询点时,输出表示所述查询点处的所述物质结构的场的场数据的已学习神经网络模型,

所述处理部通过将由所述取得部取得的所述对象向量和所述查询点输入到所述已学习神经网络模型,取得与所述查询点对应的所述场数据。

3.根据权利要求2所述的信息处理装置,其中,

所述场数据由表示构成所述物质的晶体的原子的位置的场的位置场、和表示构成所述物质的晶体的原子的种类的场的种类场表示,

所述已学习神经网络模型包含已学习的第1神经网络模型和已学习的第2神经网络模型,

所述已学习的第1神经网络模型在被输入所述任意的向量和所述查询点时,输出与所述查询点对应的所述位置场,

所述已学习的第2神经网络模型在被输入所述任意的向量和所述查询点时,输出与所述查询点对应的所述种类场,

所述处理部通过对所述已学习的第1神经网络模型输入由所述取得部取得的所述对象向量和所述查询点,取得与所述查询点对应的所述位置场,

所述处理部通过对所述已学习的第2神经网络模型输入由所述取得部取得的所述对象向量和所述查询点,取得与所述查询点对应的所述种类场。

4.根据权利要求2或3所述的信息处理装置,其中,

所述已学习神经网络模型是通过机器学习基于学习用晶体数据而预先生成的已学习模型,所述学习用晶体数据是表示所述物质的晶体结构的晶体数据,包含表示构成所述物质的晶体的原子的位置的位置数据、表示构成所述物质的晶体的原子的种类的种类数据、以及所述物质的晶体的晶格常数数据。

5.根据权利要求3所述的信息处理装置,其中,

所述已学习神经网络模型是已学习自动编码器中的已学习的第1解码器和已学习的第2解码器,

所述已学习自动编码器中的已学习的编码器在被输入表示所述物质的晶体结构的晶体数据时,输出表示所述物质的晶体的潜在向量,所述晶体数据包含表示构成所述物质的晶体的原子的位置的位置数据、表示构成所述物质的晶体的原子的种类的种类数据、以及所述物质的晶体的晶格常数数据,

所述已学习的第1解码器在被输入代替所述潜在向量的所述任意的向量和所述查询点时,输出与所述查询点对应的所述位置场,

所述已学习的第2解码器在被输入代替所述潜在向量的所述任意的向量和所述查询点时,输出与所述查询点对应的所述种类场,

所述已学习的第1解码器和所述已学习的第2解码器是通过基于所述晶体数据使具有

编码器、第1解码器以及第2解码器的自动编码器进行无监督机器学习而得到的已学习模型。

6.根据权利要求5所述的信息处理装置,其中,

所述自动编码器还具有第3解码器,

已学习的第3解码器是在被输入代替所述潜在向量的所述任意的向量时输出所述物质的晶体的晶格常数数据的已学习模型。

7.一种已学习模型生成装置,其中,

该已学习模型生成装置包含学习部,所述学习部使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。

8.根据权利要求7所述的已学习模型生成装置,其中,

该已学习模型生成装置还包含学习用取得部,所述学习用取得部取得学习用晶体数据和学习用查询点,所述学习用晶体数据表示学习用的所述物质的晶体结构,包含表示构成学习用的所述物质的晶体的原子的位置的位置数据、表示构成学习用的所述物质的晶体的原子的种类的种类数据、以及学习用的所述物质的晶体的晶格常数数据,所述学习用查询点是学习用的所述物质内的关注点,

所述学习部在使用无监督机器学习使自动编码器进行学习时,

通过对所述自动编码器中的编码器输入所述学习用晶体数据,取得表示所述学习用的物质的晶体结构的潜在向量,

通过对所述自动编码器中的第1解码器输入所述潜在向量与所述学习用查询点的组合,取得表示构成所述学习用的物质的晶体的原子的位置的场的位置场,

基于从所述第1解码器输出的所述位置场,估计构成所述学习用的物质的晶体的原子的位置,

通过对所述自动编码器中的第2解码器输入所述潜在向量与所述学习用查询点的组合,取得表示构成所述学习用的物质的晶体的原子的种类的场的种类场,

基于从所述第2解码器输出的所述种类场,估计构成所述学习用的物质的晶体的原子的种类,

通过对所述自动编码器中的第3解码器输入所述潜在向量与所述学习用查询点的组合,取得所述学习用的物质的晶体的晶格常数数据,

通过使用无监督机器学习使自动编码器进行学习,使得估计出的所述原子的位置、估计出的所述原子的种类以及从所述第3解码器输出的所述晶格常数数据的组合与所述学习用晶体数据中的所述位置数据、所述种类数据以及所述晶格常数数据的组合对应,生成已学习的第1解码器和已学习的第2解码器。

9.一种信息处理方法,其由计算机执行以下处理:利用神经网络模型,表征表示由原子点群构成的物质结构的场。

10.一种已学习模型生成方法,其由计算机执行以下处理:使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。

11.一种信息处理程序,其用于使计算机执行以下处理:利用神经网络模型,表征表示由原子点群构成的物质结构的场。

12.一种已学习模型生成程序,其用于使计算机执行以下处理:使表征表示由原子点群

构成的物质结构的场的神经网络模型进行学习。

信息处理装置、已学习模型生成装置、信息处理方法、已学习模型生成方法、信息处理程序和已学习模型生成程序

技术领域

[0001] 本公开涉及信息处理装置、已学习模型生成装置、信息处理方法、已学习模型生成方法、信息处理程序和已学习模型生成程序。

背景技术

[0002] 以往,已知有与用于3D晶体结构的基于自动编码器的生成式深度表征学习管道相关的技术(例如,参考Callum J.Court,Batuhan Yildirim,Apoorv Jain,and Jacqueline M.Cole,"3-D Inorganic Crystal Structure Generation and Property Prediction via Representation Learning",J.Chem.inf.Model.2020,60,10,4518-4535,[令和4年11月18日检索],互联网<URL:https://pubs.acs.org/doi/10.1021/acs.jcim.0c00464>)。

发明内容

[0003] 发明要解决的课题

[0004] 本公开的目的在于,表征表示由原子点群构成的物质结构的场。

[0005] 用于解决课题的手段

[0006] 为了实现上述目的,本公开的信息处理装置是包含处理部的信息处理装置,该处理部利用神经网络模型,表征表示由原子点群构成的物质结构的场。

[0007] 另外,本公开的信息处理方法是计算机执行以下处理的信息处理方法:利用神经网络模型,表征表示由原子点群构成的物质结构的场。

[0008] 另外,本公开的信息处理程序是用于使计算机执行以下处理的信息处理程序:利用神经网络模型,表征表示由原子点群构成的物质结构的场。

[0009] 发明效果

[0010] 根据本公开的信息处理装置、信息处理方法以及信息处理程序,能够表征表示由原子点群构成的物质结构的场。

附图说明

[0011] 图1是用于说明本实施方式的图。

[0012] 图2是用于说明本实施方式的图。

[0013] 图3是用于说明本实施方式的图。

[0014] 图4是用于说明本实施方式的图。

[0015] 图5是用于说明本实施方式的图。

[0016] 图6是用于说明本实施方式的图。

[0017] 图7是表示本实施方式的信息处理装置的硬件结构的框图。

[0018] 图8是用于说明本实施方式的图。

[0019] 图9是用于说明本实施方式的图。

[0020] 图10是用于说明本实施方式的图。

[0021] 图11是用于说明本实施方式的图。

[0022] 图12是用于说明本实施方式的图。

[0023] 图13是用于说明本实施方式的图。

[0024] 图14是用于说明本实施方式的图。

[0025] 图15是用于说明本实施方式的图。

[0026] 图16是用于说明本实施方式的图。

[0027] 图17是用于说明本实施方式的图。

[0028] 图18是用于说明本实施方式的图。

[0029] 图19是用于说明本实施方式的图。

[0030] 图20是用于说明本实施方式的图。

[0031] 图21是用于说明本实施方式的图。

[0032] 图22是用于说明本实施方式的图。

[0033] 图23是用于说明本实施方式的图。

[0034] 图24是用于说明本实施方式的图。

[0035] 图25是用于说明本实施方式的图。

[0036] 图26是用于说明本实施方式的图。

[0037] 图27是用于说明本实施方式的图。

[0038] 图28是用于说明本实施方式的图。

[0039] 图29是用于说明本实施方式的图。

[0040] 图30是用于说明本实施方式的图。

具体实施方式

[0041] 以下,参照附图说明本公开的实施方式的一例。在本实施方式中,以本公开的信息处理装置为例进行说明。另外,在各附图中,对相同或等效的构成要素和部分标注相同的参照符号。另外,附图的尺寸以及比率为了便于说明而被夸大,有时与实际的比率不同。

[0042] 〈概要〉

[0043] 晶体学、尤其是其中的材料的逆向设计法能够不依赖于“运气”、“偶然发现”,而对探索具有所要求的特性的材料的下一代的方法做出贡献。在本实施方式中,作为用于使用神经网络来表征晶体结构的准确且实用的方法,提案了神经结构场(NeSF)。在物理学中的向量场的概念和计算机视觉中的隐式神经表征中得到构思,所提案的NeSF将晶体结构视为连续的场,而不是原子的离散的集合。与现有的基于网格的离散空间表征不同,NeSF能够克服空间分辨率与计算复杂度之间的折衷,表征任意的晶体结构。在本实施方式中,为了评价NeSF,提案了能够恢复钙钛矿结构材料、铜酸盐超导体等各种晶体结构的晶体结构的自动编码器。广泛的定量结果表明NeSF的性能优于现有的基于网格的方法。另外,以下[ ]内的编号表示图21～图23所示的参考文献编号。

[0044] 〈1.首先〉

[0045] 在材料科学的基本范式中,假设材料特性与晶体结构紧密结合来考虑结构与特性的关系。因此,在材料科学中的以往方法中,进行材料的结构与特性的关系在理论上和实验

上的分析,探索具有优异的特性的新材料[1,2]。但是,这些以往的方法仍依赖于劳动密集型的人工分析或“偶然发现”。为了自动化或辅助材料的分析和开发,数据驱动型的方法正在材料科学中被积极地研究,建立了材料信息学(MI)的领域[3-6]。与基于物理法则的估计的传统方法不同,MI的目的是通过统计和机器学习(ML)方法根据所收集的材料的数据集,阐明材料知识(支配结构与特性之间的关系的法则等)。近年来,由于ML的技术进步和大规模材料数据库的出现,MI正在迅速发展[6,7]。因此,基于强大的神经网络的ML方法正成为MI研究的重要因素[6]。MI的应用包含根据晶体结构[6]、组成[8-11]等材料特性数据的材料特性的预测、实验数据的自动分析[12-14]、以及用于从科学文献的知识检索的自然语言处理[15]。

[0046] 许多MI研究都聚焦于预测带隙、塞贝克系数、弹性模量等特定的材料特性[16-18]。如果将这种类型的任务视为发现材料间的结构与特性的关系,则还有其他重要类型的任务,即发现构成逆问题的特性与结构的关系[19-23]。

[0047] 与材料开发中的该逆向方法的潜在有用性无关,几乎没有应对该[19-23]或其根本问题[24-26]、即特定条件下的晶体结构估计的研究。关于MI和ML,根据是输入还是输出晶体结构(即,用MI和ML用语对晶体结构进行编码或解码)而产生决定性差异。晶体结构的编码使用图神经网络[10、16、17、27]来适当地建立,但晶体结构的解码中残留技术上的瓶颈。在该调查中,应对了瓶颈。

[0048] 无机材料的晶体结构是原子在三维(3D)空间中的规则性和周期性的配置。该配置通常通过晶胞内的原子的3D位置和种类、以及定义3D空间中的晶胞的移动的晶格常数来描述。晶胞内的原子没有明确的顺序,其数量为1个至数百个不等。包含神经网络的ML模型通常为了处理而接受在固定维度上一致地排序的张量,因此在ML模型中处理晶体结构并不简单[16],经由模型来决定晶体结构更困难。

[0049] 在本实施方式中,提案了使得神经网络能够解读或决定这样的结构的晶体结构的通常表征。图1示出作为我们方法根本的重要概念。这里,晶体结构不表示为原子的离散集,而是表示为与3D空间关联的连续向量场。我们将我们的方法称为神经结构场(NeSF)。NeSF使用位置场和种类场这两种向量场,分别隐式地表示晶体结构的晶胞内的原子的位置和种类。

[0050] 为了说明NeSF的概念,假设目标的物质信息被给定为固定维向量z,并考虑从z恢复晶体结构的问题。输入z例如能够指定与材料的晶体结构相关的信息、或者关于所生成的材料的几个期望的基准。在NeSF中,神经网络f不直接将晶体结构作为f(z)输出,而是将神经网络f作为隐函数使用,间接地表征嵌入z的晶体结构。具体而言,将f作为3D笛卡尔坐标p上的向量场处理,将目标材料信息z作为条件。

[0051] [数式1]

[0052] $$s = f(p,z) \quad (1)$$

[0053] 在感兴趣的晶体结构内,在位置场中,网络f被训练为输出指示离查询点p最近的原子位置a的3D向量。因此,预想输出s为a-p。在理想地训练位置场的情况下,能够取得在任意查询点p处最近的原子位置a作为p+f(p,z)。

[0054] 在数学上,位置场能够解释为下式的标量势的梯度向量场$-\varphi(p)$。

[0055] [数式2]

[0056] $$\phi(\boldsymbol{p}) = \frac{1}{2}\min_i |\boldsymbol{p} - \boldsymbol{a}_i|^2$$

[0057] 标量势通过查询点p与最近的原子位置{a_i}之间的距离平方来表征。同样地,种类场被训练为输出表示最近的原子种类的分类概率分布。因此,种类场的输出维度为候选原子种类的数量。

[0058] 所提案的NeSF被触发为经典物理学中的向量场的概念和计算机视觉中的隐式神经表征[30-34]。为了处理对象的3D形状估计[31-33]和自由视点图像合成[30,34]等、3D计算机视觉应用中的一些表征的问题,最近提案了隐式神经表征。在3D形状估计中,如果使神经网络直接输出3D网格或点群,则会产生与晶体结构中产生的同样的表征问题。为了克服这些问题,在DeepSDF[31]中利用带符号的距离函数(SDF),神经网络f(p)通过用正或负的符号表示查询点p是否处于对象体的外侧或内侧来将3D形状模型化,输出各个标量。NeSF按照隐式神经表征的基本思路,进一步将其扩展到通过原子的位置和种类描述的晶体结构的估计。晶体结构中的原子位置的准确描述在材料科学中非常重要。因此,NeSF输出指向最近原子的向量,比3D几何现有的隐式神经表征更直接地表示原子的位置。

[0059] 将3D几何晶体结构表征为连续向量场的我们的思路在最近的MI研究[19,20,22,24-26]中使用基于网格的离散化(即,体素化)部分且隐含地进行了研究,但是没有明确考虑作为离散向量场。在这些研究中,将晶胞内的3D空间离散化为体素,对各体素分配电子密度。电子密度基本上表示体素周围的原子有无。然而,与一维(例如,音频信号)或二维(例如,图像)数据相比,3D数据的离散化在计算时间和存储器空间两者中都受到空间分辨率和计算复杂度之间的折衷的影响。例如,ICSG3D方法[24]使用32×32×32体素表示晶体结构,并使用3D卷积神经网络(CNN)对它们进行估计。由于基于体素的3DCNN消耗大量的计算和存储器,因此32×32×32体素的分辨率是在标准计算系统中训练基于体素的模型的大致极限。另一方面,在现有的晶体结构中,单胞中包含几十个以上的原子,并且单胞伸长或变形。因此,为了用体素准确地表征多种晶体结构,需要足够高的分辨率。进而,基于体素的模型只能通过在体素空间中离散化的电子密度的标量场的峰值等表征来间接地提供原子位置。所提案的NeSF克服了体素化的极限。在NeSF中,空间分辨率和所需存储器之间基本上没有折衷。理论上,NeSF能够使用紧凑的(存储器效率和参数效率高的)神经网络来代替花费成本的3DCNN,实现无限高的空间分辨率。并且,NeSF能够有效地表示包含细长或扭曲的单胞的任意晶体结构。并且,NeSF能够直接提供原子位置的笛卡尔坐标,而不是标量场的峰值。所提案的NeSF被认为突破了用于晶体结构估计的MI方法的技术瓶颈,对该方向上的MI研究的进步做出贡献。

[0060] 在下一节中,将详细说明所提案的NeSF,并且通过数值实验示出了其对于各种晶体结构的表征力。特别是NeSF成功恢复了从钙钛矿材料的较基本的结构到铜酸盐超导体的复杂结构的各种晶体结构。广泛的定量评价结果表明NeSF优于ICSG3D[24]的体素化方法。

[0061] <2结果和考察>

[0062] 首先,将说明使用NeSF估计晶体结构并训练NeSF的过程。接着,作为NeSF的应用,提出晶体结构的自动编码器。在该自动编码器中,晶体结构经由编码器嵌入到向量z(称为潜在向量)中,NeSF作为解码器发挥功能,从z复原所输入的晶体结构。基于NeSF的自动编码器的性能通过评价超过基于体素化的ICSG3D基线的重构精度而被定量分析。通过对学习出

的向量z的空间进行定性分析而提案的自动编码器,该分析表示学习出的空间并非仅随机地嵌入晶体结构,而是反映了晶体结构间的相似性。

[0063] <2.1基于NeSF的晶体结构>

[0064] 当估计目标的材料信息作为向量z而被给出时,根据z来估计晶体结构是指将晶胞内的原子的位置和种类与晶格常数一起估计。晶格常数被模型化为长度a、b以及c和角度α、β以及γ,通过使用了输入z的单纯的多层感知器(MLP)来估计。另一方面,如先前部分所述,通过NeSF的位置场f_p和种类场f_s来分别估计原子位置和原子种类。这些场也被实现为简单的MLP,分别取查询位置p和向量z作为输入,预测场值(即,3D指向向量或类别概率分布)。在图1的(b)的右侧示出NeSF网络架构的概要。

[0065] 当从编码器给出向量z时,使用NeSF的原子位置和种类的估计如图2所示,汇总为5个步骤。

[0066] 1.将粒子初始化。首先,通过MLP估计晶格常数。接着,在边界框内的3D网格点,定期地扩展被称为微粒(particle)的初始查询点$\{p^0_i\}$。边界框对于各数据集是公共的,被指定为大致包围所有训练样本的原子。

[0067] 2.使微粒移动。按照$p^{t+1}_i = p^t_i + f_p(p^t_i, z)$,使用位置场来更新各粒子$p^t_i$的位置。对所有粒子反复该过程,以取得$\{p_i\}$作为原子位置的候选。由于预想位置场指向最近的原子位置,因此粒子通过该过程朝向最近的原子移动。

[0068] 3.对粒子赋予得分。对各粒子p_i进行评分,对离群值进行过滤处理。位置场的输出的范数$||f_p(p_i, z)||$表示从p_i到最近的原子的估计距离,因此用$||f_p(p_i, z)||$对各粒子p_i赋予得分,在得分超过特定的阈值(该情况下设定为0.9Å)的情况下,舍弃该粒子。

[0069] 4.检测原子。直至该时刻为止,预想粒子在原子的周围形成簇。因此,应用简单的聚类算法,检测各簇位置作为原子位置,决定晶体结构内的原子数。这里,使用在目标检测、非极大值抑制(Non-Maximum Suppression)中众所周知的聚类算法。具体而言,将候选粒子的列表初始化$B_c = \{p_i\}$,将接收到的粒子的其他列表初始化为$B_a = \{\}$。1)从B_c中选择得分最低(估计为最接近原子)的粒子,移动到B_a。2)将所选择的粒子周围的球状区域内的粒子从B_c中删除(在该研究中,将球半径设定为了0.5Å)。通过反复这些过程直到B_c变空,取得原子位置$\{a_i\}$作为存储在B_a中的所选择的粒子。

[0070] 5.估计种类。最后,使用种类场$f_s(p, z)$来估计各原子位置a_i的原子种类。为了对a_i的错误进行鲁棒的估计,代替将a_i作为查询直接使用,而作为向种类场的查询,在各a_i的周围集中扩展新的粒子。因此,得到多个概率分布,分别预测原子a_i的种类。其中,选择频度最高的原子种类作为最终的估计值。

[0071] <2.2NeSF的训练>

[0072] NeSF的训练与上述的估计算法不同,非常简单。随机采样晶胞内的3D查询点$\{p^s_i\}$,计算这些点处的场输出的损失值。因此,监视$f_p(p^s_i, z)$和$f_s(p^s_i, z)$,分别表示最近的原子的位置和种类。但是,由于存储器使用量的实际限制,无法对查询点进行密集采样。因此,训练需要查询点的采样策略。DeepSDF[31]等用于3D形状估计的现有隐式神经表征对表面附近的训练查询点进行采样。课程DeepSDF[35]还引入了课程学习,随着训练的进行,采样密度在表面附近增强。

[0073] 为了研究用于训练位置场和种类场的期望的采样策略,通过所提案的算法来研究

它们的动态。1)粒子朝向最近的原子在位置场内反复移动。因此,位置场必须在任何地方都足够准确,以使粒子流到目的地,在原子附近必须非常准确。2)种类场仅在原子的周围被查询。因此,不需要在任何地方都是准确的,但必须对估计出的原子位置的错误具有鲁棒性。

[0074] 为了满足上述要求,引入两种采样方法,对位置场和种类场进行训练。1)全局网格采样:在该方法中,考虑均匀覆盖晶胞整体的3D网格点,通过遵循高斯分布的扰动对点进行采样。2)局部网格采样:在该方法中,考虑以各原子位置为中心的局部3D网格点,通过遵循高斯分布的扰动对点进行采样。

[0075] 为了训练位置场,组合两种采样方法。因此,查询点在晶胞整体中被均匀地采样,且在原子的周围被密集地采样。为了训练种类场,使用局部网格采样使训练查询点集中在原子附近。

[0076] <2.3晶体结构自编码器>

[0077] 为了证实和评价NeSF的表征力,提案了晶体结构的自动编码器。与其他通常的自动编码器同样地,所提案的基于NeSF的自动编码器由编码器和解码器构成。编码器是将输入晶体结构(即,晶胞内的原子的位置和种类、以及晶格常数)转换为抽象的潜在向量z的神经网络。使用NeSF的解码器从潜在向量z重建输入晶体结构。自动编码器通常用于通过自监督学习来学习数据的潜在向量表征。在该学习中,输入数据能够经由重构损失来监视学习。

[0078] 我们聚焦于晶体结构的解读,其编码通过MI进行研究。晶体结构本质上是原子的集合,因此在其编码中需要处理相对于排列不变且可变数的原子。在ML中,这种编码器通常被称为集合函数[28,29]。其中,图神经网络家族[10,16,17,27]作为常见的晶体结构编码器发挥功能。但是,这些网络将原子的位置隐式地表示为边缘,舍弃准确的坐标并对原子间的距离进行编码。该基于距离的图表征是用于确保对于坐标系的不变性的关键,但输入中的信息的损失有可能无意地妨碍重构的性能。因此,采用PointNet[28]和DeepSets[29]的基本的编码器架构。该架构不仅表示基于集合函数的网络的最简单种类,还能够保存输入晶体结构的信息,因此适合于NeSF解码器的性能评价。在图28的4.2节中详细描述了所提案的自动编码器的架构。

[0079] <训练和评价的过程>

[0080] 用ICSG3D、limited cellsize6Å (有限晶胞尺寸LCS6Å)和YBCO-like datasets这三种材料数据集训练和评价了自动编码器和ICSG3D24(基线)。这些数据集被设计为从Materials Project收集材料的晶体结构,具有各种各样的难易度。ICSG3D数据集[24]是材料集合,是包含7897个材料的3个数据集,涵盖了限定的晶系(立方体)和原型(即,AB、ABX2以及ABX3)。LCS6Å数据集由x、y、z轴向的单胞尺寸为6Å以下的6005的材料构成,晶系和原型没有限制。YBCO那样的数据集由沿着c轴具有狭窄的晶胞的100的材料构成。这些结构通常包含钇钡铜氧化物(YBCO)超导体的结构。由于结构复杂且样本较少,因此YBCO那样的数据集在所评价的3个数据集中难度最高。这些数据集的细节可以参照图27中的4.1节。

[0081] 为了训练和评价,将各数据集随机划分为训练(90.25%)、验证(4.75%)和测试(5%)集。训练集仅用于ML模型的训练。验证集用于事先验证已训练的ML模型,测试集用于训练、验证和调整超参数后计算最终的评价得分。超参数根据LCS6Å数据集的验证得分进行了调整。为了减少由验证得分随机性(网络的权重初始化中的随机性等)引起的性能变动,在各种随机种子中反复10次训练和评价,并使用得分的平均值和标准偏差来评价了性

能。在YBCO那样的数据集中仅存在100个样本,因此以与其他2个已评价数据集稍微不同的方法进行了处理。为了减少由数据分割引起的性能变动,对YBCO那样的数据集采用了20倍的交叉验证,但不是评价10次而是1次。神经网络的迭代训练是以Adam[36]为优化器而使用随机梯度下降法实施的。包括损失函数的定义的详细训练过程由图28的4.3节提供。

[0082] 重构性能是关于原子数、位置以及种类的误差而测量的。原子数的误差是未被正确地估计晶胞内的原子数的材料的比例。位置误差是重构的原子位置的平均误差。根据度量的分母,用两种方法评价了位置误差。使用实际的度量,计算到估计原子位点的最短距离,由此评价了晶体结构的实际原子位点处的平均位置误差。与此相对,检测出的度量用于通过计算到实际的原子位点的最短距离来评价估计位点处的错误。实际的度量对与原子数的过小评价关联的错误更敏感,但检测出的度量对与过大评价关联的错误更敏感。种类错误是未被正确地估计种类的原子的平均比例。与位置误差同样地,使用实际的测量基准和检测出的测量基准评价了种类的误差。这些指标的值越低,表明性能越优异。

[0083] <与ICSG3D的定量性能比较>

[0084] 表1示出了三个数据集的测试集中的、所提案的基于NeSF的自动编码器和ICSG3D基线的重构错误。总体上,所提案的方法在所有评价指标上都一致优于ICSG3D,所有数据集的种类错误和YBCO那样的数据集的所有指标的性能都得到了大幅提高。图3表示来自3个评价后的数据集的晶体结构,将测试样本与所提案的自动编码器和基于ICSG3D的重构结果进行了比较。

[0085] [表1]

[0086] 表1.重构结果

[0087]

方法	提案	ICSG3D	提案	ICSG3D	提案	ICSG3D
数据集	ICSG3D		LCS6Å		类 YBCO	
原子数误差[%]	0.53±0.25	2.67±0.84	6.35±0.50	17.28±1.13	12.00	91.00
位置误差（实际）[Å]	0.0308±0.0112	0.0877±0.0306	0.1161±0.0073	0.2006±0.0132	0.2631	(0.6311)
位置误差（检测）[Å]	0.0359±0.0226	0.1057±0.0284	0.1632±0.0123	0.2886±0.0159	0.2448	0.4358
种类误差（实际）[%]	4.31±0.39	64.39±1.91	14.78±0.81	55.70±1.40	14.78	78.08
种类误差（检测）[%]	4.36±0.39	65.05±1.85	16.11±0.66	58.32±1.45	19.89	54.24

[0088] 我们比较了所提案的基于NcSF的晶体结构的自动编码器和ICSG3D方法对三个数据集(ICSG3D、LCS6Å 和类YBCO)的晶体结构重建的性能。基于实际或检测的原子位点以两种方式评价位置和种类误差。对于类YBCO数据集、ICSG3D在100种材料中有44种未能输出任何原子,且未提供位置误差(实际)的有效得分。我们通过排除这些被不正确地估计的材料(括号内所示)计算得出得分,这表明ICSG3D的实际性能比显示值更差。

[0089] 在3个数据集中最简单的ICSG3D数据集的情况下,ICSG3D对原子数的错误和位置错误达到了优良的性能,但种类的错误变大(实际的度量和检测到的度量双方都约为65%)。与此相对,在所提案的方法中,位置误差和原子数的误差稍微改善,种类的误差大幅减少(约为4%)。这被认为是由于ICSG3D经由电子密度图估计原子种类,但所提案的方法将原子种类更直接地表示为类别分布。扩展ICSG3D来估计各体素的类别分布时,在输出中需要约100×323倍的存储器使用量,因此不现实(即,除了1个电子密度图以外,每323体素需要100种类别)。

[0090] 在比ICSG3D数据集难度大的LCS6Å 和YBCO那样的数据集的情况下,特别是关于位置误差,提案的方法在性能上的优点更加明确。LCS6Å 数据集中包含各种晶体结构(非

立方体结构、变形的晶体结构等),但YBCO那样的数据集中包含非常窄的晶体结构。进而,由于在YBCO那样的数据集中几乎不包含样本,所以有可能导致模型的过拟合(即,测试集的性能有可能大幅降低)。尽管有这些困难,但是所提案的方法能够准确地估计原子的位置和种类。

[0091] 为了详细分析方法的性能与结构的复杂度的关系,图4示出了作为中央值(点)而给出的原子的数量和与它们周围的68%的范围(着色的区域)相应的、针对来自ICSG3D和LCS6Å数据集的材料的10次测试试行中的重构错误的分布。YBCO那样的数据组仅包含在晶胞中具有13个原子的材料,因此从该分析中被排除。图4a和4b示出了检测到的原子数与实际原子数之间的带符号误差。这些结果表明,两种方法都正确地估计了ICSG3D数据集内的大部分样本(即68%以上)的原子数(图4a)。其中,ICSG3D方法对LCS6Å数据集的原子数进行了过小评价(图4b)。同样地,图4c和4d表示位置误差的分布,图4c和4d表示位置误差的分布。图4e和4f表示种类的误差分布。在大多数情况下,原子数通过两种方法被正确地估计或被过小评价,因此通基于实际的测量基准来报告错误。如果调查图4e的x=2处的分布,则能够判断ICSG3D数据集的二原子结构的种类错误的趋势。

[0092] 所提案的方法对于二原子物质的68%以上提供两个原子的正确种类,但ICSG3D常常错误地估计2个原子中的1个种类。作为整体,基于两种方法的3种错误存在与原子数一起增加的趋势,但对于原子数变化的材料,始终得到比ICSG3D优异的结果。与我们的方法相比,ICSG3D的性能有在原子数较多的材料中更显著地下降的趋势。ICSG3D具有过小地评价原子数的趋势(图4b),认为该趋势是因为ICSG3D的空间分辨率被限制为32×32×32体素。这暗示了ICSG3D无法捕捉多原子结构。

[0093] 提案方法的应关注的高性能被认为是由两个理由引起的。首先,我们的方法不使用离散化,因此比基于网格的ICSG3D更有利于估计复杂的晶体结构。在基于网格的方法中,空间分辨率受到立方体增加的计算和存储器使用量的限制。所提案的NeSF已经从分辨率和计算复杂度之间的这种折衷中得到了释放,因此能够有效地表示复杂的结构。第二,使用MLP的所提案的NeSF的模型尺寸远小于ICSG3D的3DCNN架构的模型尺寸。通常,ML模型所需的训练样本的数量与可训练的参数的数量相关。在基于网格的方法中,使用包含许多可训练参数的3D卷积滤波器的层。相比之下,NeSF使用隐式的神经表征,将3D空间间接描述为场而不是体素。因此,能够以比3DCNN少的参数通过MLP高效地实现。具体而言,基于NeSF的自动编码器中有76万个参数,这仅为基于3DCNN的ICSG3D(3,400万个参数)的参数的数量的2.24%。由于该差异,特别是在YBCO那样的数据集等小的数据集中,NeSF比基于网格的方法更有利。

[0094] <2.4潜在空间插值>

[0095] 通过检查晶体结构的学习出的潜在空间,定性地分析了所提案的基于NeSF的自动编码器的特性。通常,优良的潜在空间需要在空间内紧密地映射相似的项目(在属性、特性、类别等方面)。由此,提供使人工和机械的分析变得容易的潜在数据表征。为了评价晶体结构的潜在空间的构建,将潜在空间的转变可视化为一系列的晶体结构。在潜在空间被训练为从结构相似性的观点出发捕捉材料间的关系的情况下,如果对潜在空间内的2点间进行插值,则应该生成具有相似的晶体结构的一系列材料。

[0096] 插值分析如下进行。

[0097] 1.设定ICSG3D测试集作为源和目的地材料,经由已训练的编码器取得它们的潜在向量作为z_{src}和z_{dst}。

[0098] 2.为了取得潜在向量序列,在潜在空间内对z_{src}和z_{dst}之间进行线性插值。

[0099] 3.经由已训练解码器(NeSF)对各潜在向量进行解码,取得其晶体结构。

[0100] 由于源和目的地之间的中间晶体结构是经由已训练的解码器而从潜在向量重建的,因此这些结构有时不在数据集中显示。

[0101] 为了便于分析的解释,选择了公知的zinc-blende(闪锌矿)和rock-salt structure families(岩盐结构族)作为基准材料。任一族均具有AX中给出的组成,A为阳离子,X为阴离子,晶体结构基于立方晶系。因此,当源和目的地的材料属于这些族中的任何一个时,需要在整个插值路径中保持特征性结构和结构原型。

[0102] 作为最初的例子,在图5中示出从ZnS(mp-10695)向CdS(mp-2469)的插值结果。所得到的组成式的转变为ZnS→$MgZn_3S_4$($Mg_{0.25}Zn_{0.75}S$)→$MgZnS_2$($Mg_{0.5}Zn_{0.5}S$)→Mg_3ZnS_4($Mg_{0.75}Zn_{0.25}S$)→MgS→$MgCd_3S_4$($Mg_{0.25}Cd_{0.75}S$)→CdS。

[0103] 作为第2个例子,在图6中示出从MgO(mp-1265)向NaCl(mp-22862)的插值结果。所得到的组成式的转变为MgO→$NaMgO_2$($Na_{0.5}Mg_{0.5}O$)→NaO→Na_2ClO($NaCl_{0.5}O_{0.5}$)→$Na_4C_{13}O$($NaCl_{0.75}O_{0.25}$)→NaCl。

[0104] 进而,在SI中,提供从NaCl(mp-22862)向PbS、从MgO(mp-1265)向CaO(mp-2605)、从PbS(mp-21276)向CaO(mp-2605)的内插结果。

[0105] 在插值的例子中,几乎保持了组成AX和立方体结构。此外,构图沿着插值路径连续地变化而不崩溃。这些结果启示,我们的编码器通过在抽象的空间中捕捉晶体结构的特征来学习有意义的连续表征,所提案的NeSF模型将这些表征很好地解码为晶体结构。

[0106] <2.5极限和将来的方向性>

[0107] 该研究主要聚焦于用于使用隐式神经表征来估计晶体结构的基本方法的开发。为了启示进一步改善的余地和将来的作业中的重要的方向性,确定了NeSF的3个主要的限制。

[0108] 首先,NeSF采用比较简单的网络架构,没有充分研究架构设计的选项。例如,NeSF将要素作为相互独立的分类(独热)向量来处理。因此,不会使用要素的物理特性、相似性来显式地训练模型。另一方面,在其他开发中,要将要素的物理特性和功能显式地注入模型。例如,现有的晶体结构编码器[16]使用组和周期、电子密度、原子半径、电负性等指纹来代替使用独热向量来表示输入要素。在ICSG3D24中,原料种类的输出不是在我们的方法中考虑的分类损失,而是使用原子序数的均方误差来训练。通过将这些方案嵌入模型中,有可能能够反映潜在空间中的要素与重构的晶体结构之间的特性和相似性。与NeSF相关的另一个架构上的选择是为了表征晶体结构而使用以往的(当前的选择)或者是否使用原始晶胞(primitive cell)。无论哪种形式,根据应用都有优点和缺点。传统的晶胞提供了更直观的可视化,以便于手动分析,而原始晶胞提供了更紧凑的结构表征,以便于计算机处理。在未来的作业中,需要彻底分析网络架构的设计。

[0109] 另一种限制是所提案的NeSF没有明确考虑空间群的对称性。因此,通过NeSF估计的现有的晶胞内的原子的局部空间配置不一定遵循空间群对称性。ICSG3D[24]也有同样的限制,但对称性是晶体学中的重要概念。因此,将空间群对称性的制约嵌入NeSF是将来的作业的重要方向性。

[0110] 最后,以评价为目的,不采用变分自动编码器[16,25,37]、生成对抗网络[26,37-39]等生成模型,而采用自动编码器架构。这些生成模型有意地打乱潜在的结构表征,生成在数据集中不显示的多种结构。生成模型的这个方面更适合于新的结构的发现,但是缺少真实结构会妨碍定量的和可靠高的性能分析。

[0111] NeSF需要使用适当的性能分析,在将来的作业中应用于生成模型。

[0112] <3.总结>

[0113] 在本实施方式中,提案了使用神经网络来估计晶体结构的NeSF。难以使用神经网络直接决定晶体结构。这是因为这些结构基本上被表示为包含各种数量的原子的未排序的集合。NeSF不是将晶体结构作为离散的原子的集合,而是作为连续的向量场进行处理,由此克服该问题。NeSF的思路借鉴了物理学中的向量场和计算机视觉中的最近的隐式神经表征。隐式神经表征是使用神经网络来表示3D几何的ML方法。NeSF引入位置场和种类场并扩展该方法,分别估计晶体结构的原子位置和种类。与现有的基于网格的用于表征晶体结构的方法不同,NeSF没有空间分辨率和计算复杂度之间的折衷,能够表征任意的晶体结构。

[0114] NeSF被应用为晶体结构的自动编码器,通过具有多种晶体结构的数据集证实了其性能和表征力。定量的性能分析特别是在复杂的晶体结构的估计中,与现有的基于网格的方法相比,显示出了基于NeSF的自动编码器的明确优点。进而,通过学习出的潜在空间的定性分析,明确了自动编码器不是随机地映射晶体结构,而是捕捉晶体结构间的相似性。

[0115] 在材料科学中,晶体结构的设计和构建是检索具有目标特性的材料时的基本过程。ML由于神经网络的发展而迅速进步,使用这些网络来表征任意的晶体结构对于下一代的材料开发是不可或缺的。例如,NeSF能够简单地嵌入到变分自动编码器、生成对抗网络等强大的深层生成模型中,能够发现新的晶体结构。这样的晶体结构的生成模型对于作为MI的主要课题的材料的逆向设计而言是重要的。NeSF能够克服晶体结构估计中ML的技术瓶颈,开展下一代材料开发的途径。

[0116] 图7是表示本实施方式的信息处理装置10的硬件结构的框图。如图7所示,信息处理装置10具有CPU(Central Processing Unit:中央处理器)12、存储器14、存储装置16、输入输出I/F(Interface:接口)18、存储介质读取装置20以及通信I/F22。各结构经由总线24以能够相互通信的方式连接。

[0117] 在存储装置16中存储有用于执行后述的各处理的信息处理程序。CPU 12是中央运算处理单元,执行各种程序或控制各结构。即,CPU 12从存储装置16读出程序,并将存储器14作为作业区域来执行程序。CPU 12按照存储于存储装置16的程序,进行上述各种运算处理。

[0118] 存储器14由RAM(Random Access Memory:随机存取存储器)构成,作为作业区域暂时存储程序和数据。存储装置16由ROM(Read Only Memory:只读存储器)以及HDD(Hard Disk Drive:硬盘驱动器)、SSD(Solid State Drive:固态驱动器)等构成,存储包含操作系统的各种程序以及各种数据。

[0119] 输入输出I/F 18是进行来自外部装置的数据的输入以及向外部装置的数据的输出的接口。另外,例如也可以连接键盘、鼠标等用于进行各种输入的输入装置以及例如显示器、打印机等用于输出各种信息的输出装置。作为输出装置,也可以通过采用触摸面板显示器而作为输入装置发挥功能。

[0120] 存储介质读取装置20进行CD(Compact Disc:压缩盘)-ROM、DVD(Digital Versatile Disc:数字多功能盘)-ROM、蓝光光盘、USB(Universal SerialBus:通用串行总线)存储器等各种存储介质所存储的数据的读入、对存储介质的数据的写入等。

[0121] 通信I/F 22是用于与其他设备进行通信的接口,例如使用Ethernet(注册商标)、FDDI、Wi-Fi(注册商标)等标准。

[0122] 本实施方式的信息处理装置10使用如上所述的方法,使包含编码器和解码器而构成的自动编码器学习。由此,能够利用神经网络模型来表征表示由原子点群构成的物质结构的场。

[0123] 接着,对信息处理装置10的功能结构进行说明。如图8所示,信息处理装置10在功能上包含学习用取得部102、学习部104、取得部108以及处理部110。另外,在信息处理装置10的规定的存储区域设置有数据存储部100和已学习模型存储部106。各功能结构通过CPU 12读出存储于存储装置16的各程序并在存储器14中展开执行来实现。

[0124] 首先,信息处理装置10使神经网络模型进行学习,所述神经网络模型表征表示由原子点群构成的物质结构的场。

[0125] 在数据存储部100中存储有表示物质的晶体结构的学习用晶体数据。学习用晶体数据包含表示构成物质晶体的原子位置的位置数据、表示构成物质晶体的原子种类的种类数据、以及物质晶体的晶格常数数据而构成。

[0126] 图9是表示数据存储部100中存储的多个学习用晶体数据的一例的图。如图9所示,1个学习用晶体数据是表示某1个物质的晶体结构的数据。如图9所示,1个学习用晶体数据将表示构成某物质的晶体的多个原子的位置的位置数据、表示构成该物质的晶体的多个原子的种类的种类数据、以及该物质的晶体的晶格常数数据建立对应地存储。位置数据是多个原子各自的三维位置坐标数据。另外,种类数据是表示多个原子各自的种类的标签数据。另外,晶格常数数据中包含晶轴的长度和轴间角度。

[0127] 图10是用于说明本实施方式的自动编码器的结构和本实施方式的信息处理装置10执行的处理概要的图。图10是将图1的(b)进一步简化的图。

[0128] 如图10所示,本实施方式的自动编码器AE构成为包含编码器E、第1解码器D1、第2解码器D2以及第3解码器D3。

[0129] 如图10所示,编码器E在被输入表示构成物质晶体的原子位置的位置数据P、表示构成物质晶体的原子种类的种类数据S、以及物质晶体的晶格常数数据L的组合时,输出潜在向量z。另外,对于1个物质的晶体结构,设定表示位置数据P、种类数据S和晶格常数数据L的组合的1个晶体数据。

[0130] 另外,如图10所示,第1解码器D1在被输入作为物质内的关注点的查询点p与潜在向量z的组合时,输出表示构成物质晶体的原子位置的场的位置场f_p。基于该位置场f_p,计算构成该物质晶体的原子的估计位置数据P_e。计算方法将后述。

[0131] 另外,如图10所示,第2解码器D2在被输入作为物质内的关注点的查询点p与潜在向量z的组合时,输出表示构成物质晶体的原子种类的场的种类场f_s。基于该种类场f_s,计算构成该物质晶体的原子的估计种类数据S_e。计算方法将后述。

[0132] 另外,如图10所示,第3解码器D3在被输入作为物质内的关注点的查询点p和潜在向量z时,输出物质晶体的估计晶格常数数据L_e。此外,在本实施方式中,以第3解码器D3为

一个的情况为例进行说明,但第3解码器D3也可以是两个。在该情况下,例如,第3解码器D3由输出晶轴的长度的解码器和输出轴间角度的解码器构成。

[0133] 本实施方式的信息处理装置10通过无监督机器学习(或有自监督学习)来学习自动编码器AE的各参数,使得输入到自动编码器AE的位置数据P、种类数据S以及晶格常数数据L的组合与从自动编码器AE输出的估计位置数据P_e、估计种类数据S_e以及估计晶格常数数据L_e的组合一致。由此,得到已学习自动编码器AE。另外,得到作为已学习自动编码器AE的构成要素的已学习的第1解码器D1、已学习的第2解码器D2以及已学习的第3解码器D3。

[0134] 学习用取得部102在受理使自动编码器AE学习的意思的指示信号时,读出存储于数据存储部100的学习用晶体数据。此外,学习用取得部102设定作为学习用的物质内的关注点的学习用查询点。另外,学习用取得部102也可以通过随机地采样学习用的物质内的空间中的位置,来设定学习用查询点。

[0135] 学习部104在使用无监督机器学习使自动编码器AE学习时,通过对自动编码器AE中的编码器E输入学习用晶体数据,取得表示学习用的物质的晶体结构的潜在向量z。此外,编码器E例如能够通过使用上述的PointNet[28]或DeepSets[29]的思路来实现。

[0136] 接着,学习部104通过对自动编码器AE中的第1解码器D1输入潜在向量z与学习用查询点p的组合,来取得表示构成学习用的物质晶体的原子位置的场的位置场f_p。然后,学习部104基于从第1解码器D1输出的位置场f_p,估计构成学习用的物质晶体的原子的位置。

[0137] 图11是用于说明学习用查询点的设定和位置场的图。另外,图11是与图2同样的图,但为了说明而再次示出。如图11所示,在物质内的空间M中,设定多个学习用查询点。图11所示的白圆圈表示学习用查询点。另外,图11的(a)所示的A1、A2、A3表示物质内的空间M中的实际的原子位置。

[0138] 学习部104通过对第1解码器D1输入学习用查询点p和潜在向量z的组合,取得如图11的(b)所示的位置场f_p。图11所示的箭头分别与位置场f_p对应。

[0139] 然后,如图11的(b)、(c)所示,学习部104根据从第1解码器D1输出的位置场f_p,反复更新多个学习用查询点中的各个学习用查询点的位置。具体而言,学习部104通过对多个学习用查询点中的各个学习用查询点的位置加上位置场f_p所表示的向量,生成新的学习用查询点的位置。由此,如图11的(d)所示,多个学习用查询点中的各个学习用查询点的位置向实际的原子位置收敛。然后,学习部104基于多个学习用查询点中的各个学习用查询点的位置,例如通过使用Non-max Suppression(非极大值抑制),得到如图11的(e)所示的原子的估计位置P_e1、P_e2、P_e3。

[0140] 接着,学习部104通过对自动编码器AE中的第2解码器D2输入潜在向量z与学习用查询点p的组合,来取得表示构成学习用的物质的晶体的原子的种类的场的种类场f_s。然后,学习部104基于从第2解码器D2输出的种类场f_s,估计构成学习用的物质的晶体的原子的种类。

[0141] 图12是用于说明学习用查询点的设定和种类场的图。另外,图12是与图2同样的图,但为了说明而再次示出。如图12所示,在物质内的空间M中,设定多个学习用查询点。与图12同样地,图12所示的白圆圈表示学习用查询点。

[0142] 具体而言,如图12的(f)(g)所示,学习部104对原子的估计位置P_e1、P_e2、P_e3的位置和估计位置P_e1、P_e2、P_e3的周围位置设定多个学习用查询点。

[0143] 然后,学习部104通过对第2解码器D2输入学习用查询点p和潜在向量z的组合,取得种类场f_s。如图12的(g)所示,种类场f_s相当于表示按每个学习用查询点得到的原子的种类的概率。在图12的(g)(h)所示的例子中,示出了位于某个学习用查询点的原子的种类是铁Fe的概率最高,位于其他学习用查询点的原子的种类是铜Cu的概率最高。

[0144] 然后,如图12的(g)、(h)所示,学习部104根据从第2解码器D2输出的种类场f_s,估计与多个学习用查询点各自的位置相应的原子的种类。例如,学习部104针对估计位置P_e1、P_e2、P_e3中的每一个,确定在与该估计位置相应的学习用查询点中概率最高的元素和在该估计位置的周围的学习用查询点中概率最高的元素。然后,学习部104将针对估计位置P_e1、P_e2、P_e3中的每一个设定的多个学习查询点中最频繁出现的元素估计为估计位置P_e1、P_e2、P_e3处的原子种类。因此,得到存在于估计位置P_e1处的原子的估计种类S_e1、存在于估计位置P_e2处的原子的估计种类S_e2以及存在于估计位置P_e3处的原子的估计种类S_e3。

[0145] 例如,如图12的(i)所示,估计位于估计位置P_e1的原子的种类为硅Si,估计位于估计位置P_e2的原子的种类为铁Fi,估计位于估计位置P_e3的原子的种类为铜Cu。

[0146] 接着,学习部104通过对自动编码器AE中的第3解码器D3输入潜在向量z,来取得学习用的物质的晶体的晶格常数数据L_e。从第3解码器D3输出的晶格常数数据L_e包含晶轴的长度和轴间角度。

[0147] 然后,学习部104使用无监督机器学习使自动编码器AE进行学习,以使估计出的原子的位置、估计出的原子的种类以及从第3解码器D3输出的晶格常数数据的组合与学习用晶体数据中的原子的位置、原子的种类以及晶格常数数据的组合对应,由此生成已学习的第1解码器D1和已学习的第2解码器D2。

[0148] 例如,在上述例子中,学习部104使用无监督机器学习使自动编码器AE进行学习,以使估计出的原子的估计位置P_e1、P_e2、P_e3与学习用晶体数据中的原子的位置数据P1、P2、P3一致,且估计出的原子的估计种类S_e1、S_e2、S_e3与学习用晶体数据中的原子的种类数据S1、S2、S3一致,且从第3解码器D3输出的晶格常数数据L_e与学习用晶体数据中的晶格常数数据L一致,由此生成已学习的第1解码器D1和已学习的第2解码器D2。

[0149] 学习部104将已学习自动编码器AE存储到已学习模型存储部106。在已学习自动编码器AE中还存储有已学习的第1解码器D1、已学习的第2解码器D2以及已学习的第3解码器D3,因此这些已学习模型也被存储到已学习模型存储部106。

[0150] 此外,已学习的编码器E在被输入表示物质的晶体结构的晶体数据,即包含表示构成物质晶体的原子位置的位置数据、表示构成物质晶体的原子种类的种类数据、以及物质晶体的晶格常数数据的晶体数据时,输出表示物质晶体的潜在向量z。

[0151] 如上所述,表示本实施方式的物质结构的场的场数据由表示构成物质晶体的原子位置的场的位置场和表示构成物质晶体的原子种类的场的种类场表示。因此,作为已学习的第1神经网络模型的一例的已学习的第1解码器D1在被输入任意的向量和查询点时,输出与查询点对应的位置场f_p。另外,作为已学习的第2神经网络模型的一例的第2解码器D2在被输入任意的向量和查询点时,输出与查询点对应的种类场f_s。此外,已学习的第1解码器D1和已学习的第2解码器D2是在被输入任意的向量和查询点时,输出表示查询点处的物质结构的场的场数据的已学习神经网络模型的一例。

[0152] 已学习的第3解码器在被输入代替潜在向量z的任意的向量与查询点的组合时,输

出物质晶体的晶格常数数据L_e。

[0153] 取得部108在受理了取得对象物质的晶体结构的场数据的意思的指示信号时,读出存储在已学习模型存储部106中的已学习第1解码器D1和已学习第2解码器D2。另外,取得部108例如取得从用户输入的对象向量和物质内的关注点即查询点。

[0154] 此外,对象向量是代替潜在向量z的任意的向量,只要与潜在向量同样地描述材料的性质,则可以是任意的向量。例如,对象向量也可以是表示用户期望的物性的向量。例如,也可以是在对象向量的第1成分中存储[bandgap=xxx]、在第2成分中存储[formation energy=xxx]那样的表示物质的物性的向量。在该情况下,已学习的第1解码器D1和已学习的第2解码器D2被作为事先学习模型来处理,通过表示物质物性的向量进行再学习,由此从已学习的第1解码器D1和已学习的第2解码器D2输出有用的场数据。

[0155] 处理部110通过对已学习第1解码器D1和已学习第2解码器D2输入由取得部108取得到的对象向量和查询点,来取得与查询点对应的场数据。

[0156] 具体而言,处理部110通过对已学习第1解码器D1输入由取得部108取得到的对象向量和查询点,来取得与查询点对应的位置场f_p。另外,处理部110通过对已学习第2解码器D2输入由取得部108取得到的对象向量和查询点,来取得与查询点对应的种类场f_s。

[0157] 从已学习第1解码器D1和已学习第2解码器D2输出的场数据是与对象向量和查询点对应的、表示物质结构的场的数据。通过向已学习的第1解码器D1和已学习的第2解码器D2输入作为任意向量的对象向量来代替潜在向量,能够预测对象向量所表示的晶体结构等。

[0158] 例如,在对象向量是上述那样的表示期望的物性的向量的情况下,输出与该向量对应的场数据。通过将多个查询点输入到已学习的第2解码器D2和已学习的第2解码器D2,来取得这多个查询点处的场数据,因此还能够基于这些场数据来生成具有期望物性的物质的晶体结构。

[0159] 作为对象向量,例如还能够利用XRD(X射线衍射)的测量数据。在该情况下,通过将对象物质的XRD的测量数据输入到已学习第1解码器D1和已学习第2解码器D2,能够预测对象物质的晶体结构。另外,还能够将本实施方式的已学习第1解码器D1和已学习第2解码器D2与将文本数据等其他模态作为处理对象的编码器组合起来利用。

[0160] <信息处理装置10的作用>

[0161] 接下来,参照附图,对实施方式的信息处理装置10的作用进行说明。信息处理装置10在受理学习用晶体数据时,将其存储于数据存储部100。然后,信息处理装置10在受理学习处理开始的指示信号时,执行图13所示的已学习模型生成处理例程。

[0162] <已学习模型生成处理例程>

[0163] 在步骤S100中,学习用取得部102取得存储在数据存储部100中的多个学习用晶体数据。

[0164] 在步骤S102中,学习用取得部102从在上述步骤S100中取得的多个学习用晶体数据中设定一个学习用晶体数据。然后,学习部104在所设定的学习用晶体数据表示的物质内的空间中设定多个学习用查询点p_i。另外,i是用于识别学习用查询点的索引。

[0165] 在步骤S104中,学习部104通过对自动编码器AE中的编码器E输入在上述步骤S102中设定的学习用晶体数据中的位置数据P、种类数据S以及晶格常数数据L,来取得潜在向量

z。

[0166] 在步骤S106中,学习部104通过将在步骤S102中设定的学习用查询点p_i和在步骤S104中设定的潜在向量z的组合输入到第1解码器D1,取得位置场f_p。另外,学习部104针对多个学习用查询点的每一个,取得位置场f_p。

[0167] 在步骤S108中,学习部104通过将在步骤S102中设定的学习用查询点p_i和在步骤S104中设定的潜在向量z的组合输入到第2解码器D2,取得种类场f_s。另外,学习部104针对多个学习用查询点的每一个,取得种类场f_s。

[0168] 在步骤S110中,学习部104通过将在步骤S102中设定的学习用查询点p_i和在步骤S104中设定的潜在向量z的组合输入到第3解码器D3,来取得晶格常数数据L_e。另外,学习部104针对多个学习用查询点的每一个,取得晶格常数数据L_e。

[0169] 在步骤S112中,学习部104基于在步骤S106中得到的位置场f_p来估计原子的位置。另外,学习部104通过上述的方法,更新多个学习用查询点的位置,得到最终的估计位置P_e。

[0170] 在步骤S114中,学习部104基于在步骤S112中得到的原子的估计位置P_e和在步骤S108中得到的种类场f_s来估计原子的种类。因此,学习部104得到存在于估计位置P_e处的原子的估计种类S_e。

[0171] 在步骤S116中,学习部104使用无监督机器学习使自动编码器AE进行学习,使得在步骤S112中得到的原子的估计位置P_e与学习用晶体数据中的原子的位置数据P一致,并且在步骤S114中得到的原子的估计种类S_e与学习用晶体数据中的原子的种类数据S一致,并且在步骤S110中得到的晶格常数数据L_e与学习用晶体数据中的晶格常数数据L一致,由此生成已学习的第1解码器D1和已学习的第2解码器D2。

[0172] 反复步骤S102～步骤S116的处理,直到满足机器学习的结束条件为止。例如,作为机器学习的结束条件,可采用执行了规定的反复次数的机器学习的处理、或者从自动编码器AE输出的数据与学习用晶体数据之间的误差为规定的阈值以下这样的条件。此外,在上述中,以设定一个学习用晶体数据来执行学习的情况为例进行了说明,但并不限定于此,也可以使用多个学习用晶体数据来一次执行机器学习。

[0173] 在步骤S118中,学习部104确定是否满足上述结束条件。在满足结束条件的情况下,转移到步骤S120。另一方面,在不满足结束条件的情况下,返回步骤S102。

[0174] 在步骤S120中,学习部104将通过步骤S102～步骤S116的机器学习处理得到的已学习自动编码器AE存储到已学习模型存储部106。

[0175] 接下来,信息处理装置10在受理了对象向量和查询点时,执行图14所示的估计处理例程。

[0176] <估计处理例程>

[0177] 在步骤S200中,取得部108取得对象向量和查询点。

[0178] 在步骤S202中,处理部110读出存储在已学习模型存储部106中的已学习的第1解码器D1和已学习的第2解码器D2。

[0179] 在步骤S204中,处理部110对在步骤S202中读出的已学习的第1解码器D1输入在步骤S200中取得到的对象向量和查询点,由此取得与对象向量和查询点相应的位置场f_p。

[0180] 在步骤S206中,处理部110对在步骤S202中读出的已学习的第2解码器D2输入在步骤S200中取得到的对象向量和查询点,由此取得与对象向量和查询点相应的种类场f_s。

[0181] 在步骤S208中,处理部110将在步骤S204中所取得的位置场f_p和在步骤S206中所取得的种类场f_s作为结果而输出。

[0182] 如以上说明的那样,根据信息处理装置10,能够使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。另外,根据信息处理装置10,能够利用神经网络模型来表征表示由原子点群构成的物质结构的场。

[0183] 具体而言,信息处理装置10取得学习用晶体数据和学习用查询点,所述学习用晶体数据表示学习用的物质的晶体结构,包含表示构成学习用的物质的晶体的原子位置的位置数据、表示构成学习用的物质的晶体的原子种类的种类数据和学习用的物质的晶体的晶格常数数据,所述学习用查询点是学习用的物质内的关注点。信息处理装置10在使用无监督机器学习使自动编码器学习时,通过对自动编码器中的编码器输入学习用晶体数据,取得表示学习用的物质的晶体结构的潜在向量。信息处理装置10通过对自动编码器中的第1解码器输入潜在向量与学习用查询点的组合,取得表示构成学习用的物质的晶体的原子位置的场的位置场。信息处理装置10基于从第1解码器输出的位置场,估计构成学习用的物质的晶体的原子的位置。信息处理装置10通过对自动编码器中的第2解码器输入潜在向量与学习用查询点的组合,取得表示构成学习用的物质的晶体的原子的种类的场的种类场数据。信息处理装置10基于从第2解码器输出的种类场,估计构成学习用的物质的晶体的原子的种类。信息处理装置10通过对自动编码器中的第3解码器输入与潜在向量学习用查询点的组合,取得学习用的物质的晶体的晶格常数数据。信息处理装置10使用无监督机器学习使自动编码器学习,以使估计出的原子的位置、估计出的原子的种类以及从第3解码器输出的晶格常数数据的组合与学习用晶体数据中的位置数据、种类数据以及晶格常数数据的组合对应,由此生成已学习的第1解码器和已学习的第2解码器。

[0184] 在本实施方式中,通过采用上述那样的方法,能够通过神经网络来表征表示物质结构的场。另外,能够根据固定长度的向量生成晶体数据。

[0185] 此外,信息处理装置10取得对象向量和作为物质中的关注点的查询点,并且将对象向量和查询点输入到作为已学习神经网络模型的一例的第1解码器和第2解码器,由此取得与查询点对应的场数据。该场数据由表示构成物质晶体的原子位置的场的位置场和表示构成物质晶体的原子种类的场的种类场表示。第1解码器在被输入代替潜在向量的任意向量和查询点时,输出与查询点对应的位置场。第2解码器在被输入代替潜在向量的任意向量和查询点时,输出与查询点对应的种类场。信息处理装置10通过将对象向量和查询点输入到第1解码器来取得与查询点对应的位置场。此外,信息处理装置10通过将对象向量和查询点输入到第2解码器来取得与查询点对应的种类场。例如,在对象向量是表示期望的物性的向量的情况下,输出与该向量对应的场数据。通过将多个查询点输入到已学习的第2解码器D2和已学习的第2解码器D2,来取得这多个查询点处的场数据,因此还能够基于这些场数据来生成具有期望物性的物质的晶体结构。

[0186] 此外,在变更对象向量的情况下,优选使已学习的第1解码器D1和已学习的第2解码器D2再学习。在该情况下,已学习的第1解码器D1和已学习的第2解码器D2被用作事先学习模型。

[0187] 此外,在上述实施方式中,以通过神经网络将表示晶体结构的场模型化的情况为例进行了说明,但并不限定于此。不限于包含重复结构的晶体结构,也可以通过神经网络将

不包含重复结构的由原子点群构成的物质结构模型化。在该情况下,可能不需要晶格常数的估计,因此可能不需要图1所示的晶格解码器(Lattice Decorder)。

[0188] 另外,在上述实施方式中,以输入到解码器的向量z是从编码器输出的潜在向量的情况为例进行了说明,但不限于此。例如,也可以构成为将表示用户期望的物性的向量设为z,能够将该向量z提供给解码器(例如,用向量表征bandgap＝xxx、formation energy＝xxx等条件)。另外,也可以构成为利用随机噪声作为向量z。

[0189] 在上述实施方式中,作为神经网络模型,以利用包含编码器和解码器而构成的自动编码器的情况为例进行了说明,但并不限定于此。作为神经网络模型,也可以利用其他模型,例如,也可以利用生成对抗网络(GAN)。

[0190] 另外,图15～图30表示用于说明本实施方式的详情的图。

[0191] 另外,也可以由CPU以外的各种处理器执行在上述实施方式中CPU读入软件(程序)并执行的各处理。作为该情况下的处理器,例示FPGA(Field-Programmable Gate Array:现场可编程门阵列)等在制造后能够变更电路结构的PLD(Programmable Logic Device:可编程逻辑器件)、以及ASIC(Application Specific Integrated Circuit:专用集成电路)等具有为了执行特定的处理而专门设计的电路结构的处理器即专用电路等。另外,可以由这各种处理器中的一个执行各处理,也可以由相同种类或不同种类的两个以上的处理器的组合(例如,多个FPGA以及CPU与FPGA的组合等)执行各处理。另外,更具体而言,这各种处理器的硬件结构是组合了半导体元件等电路元件的电路。

[0192] 另外,在上述实施方式中,对各程序预先存储(例如,安装)于存储装置的方式进行了说明,但并不限定于此。程序也可以以存储于CD-ROM、DVD-ROM、蓝光光盘、USB存储器等存储介质的方式提供。另外,程序也可以设为经由网络从外部装置下载的方式。

[0193] (附记)

[0194] 以下,对本公开的方式进行附记。

[0195] (附记1)

[0196] 一种信息处理装置,其中,该信息处理装置包含处理部,所述处理部利用神经网络模型,表征表示由原子点群构成的物质结构的场。

[0197] (附记2)

[0198] 在附记1所记载的信息处理装置中,

[0199] 该信息处理装置还包含取得部,所述取得部取得对象向量和物质内的关注点即查询点,

[0200] 所述神经网络模型是在被输入任意的向量和所述查询点时,输出表示所述查询点处的所述物质结构的场的场数据的已学习神经网络模型,

[0201] 所述处理部通过将由所述取得部取得的所述对象向量和所述查询点输入到所述已学习神经网络模型,取得与所述查询点对应的所述场数据。

[0202] (附记3)

[0203] 在附记2所记载的信息处理装置中,

[0204] 所述场数据由表示构成所述物质的晶体的原子的位置的场的位置场、和表示构成所述物质的晶体的原子的种类的场的种类场表示,

[0205] 所述已学习神经网络模型包含已学习的第1神经网络模型和已学习的第2神经网

络模型，

[0206] 所述已学习的第1神经网络模型在被输入所述任意的向量和所述查询点时，输出与所述查询点对应的所述位置场，

[0207] 所述已学习的第2神经网络模型在被输入所述任意的向量和所述查询点时，输出与所述查询点对应的所述种类场，

[0208] 所述处理部通过对所述已学习的第1神经网络模型输入由所述取得部取得的所述对象向量和所述查询点，取得与所述查询点对应的所述位置场，

[0209] 所述处理部通过对所述已学习的第2神经网络模型输入由所述取得部取得的所述对象向量和所述查询点，取得与所述查询点对应的所述种类场。

[0210] (附记4)

[0211] 在附记2或附记3所记载的信息处理装置中，

[0212] 所述已学习神经网络模型是通过机器学习基于学习用晶体数据而预先生成的已学习模型，所述学习用晶体数据是表示所述物质的晶体结构的晶体数据，包含表示构成所述物质的晶体的原子的位置的位置数据、表示构成所述物质的晶体的原子的种类的种类数据、以及所述物质的晶体的晶格常数数据。

[0213] (附记5)

[0214] 在附记3所记载的信息处理装置中，

[0215] 所述已学习神经网络模型是已学习自动编码器中的已学习的第1解码器和已学习的第2解码器，

[0216] 所述已学习自动编码器中的已学习的编码器在被输入表示所述物质的晶体结构的晶体数据时，输出表示所述物质的晶体的潜在向量，所述晶体数据包含表示构成所述物质的晶体的原子的位置的位置数据、表示构成所述物质的晶体的原子的种类的种类数据、以及所述物质的晶体的晶格常数数据，

[0217] 所述已学习的第1解码器在被输入代替所述潜在向量的所述任意的向量和所述查询点时，输出与所述查询点对应的所述位置场，

[0218] 所述已学习的第2解码器在被输入代替所述潜在向量的所述任意的向量和所述查询点时，输出与所述查询点对应的所述种类场，

[0219] 所述已学习的第1解码器和所述已学习的第2解码器是通过基于所述晶体数据使具有编码器、第1解码器以及第2解码器的自动编码器进行无监督机器学习而得到的已学习模型。

[0220] (附记5)

[0221] 在附记4所记载的信息处理装置中，

[0222] 所述自动编码器还具有第3解码器，

[0223] 已学习的第3解码器是在被输入代替所述潜在向量的所述任意的向量时输出所述物质的晶体的晶格常数数据的已学习模型。

[0224] (附记7)

[0225] 一种已学习模型生成装置，其中，该已学习模型生成装置包含学习部，所述学习部使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。

[0226] (附记8)

[0227] 在附记7所记载的已学习模型生成装置中，

[0228] 该已学习模型生成装置还包含学习用取得部，所述学习用取得部取得学习用晶体数据和学习用查询点，所述学习用晶体数据表示学习用的所述物质的晶体结构，包含表示构成学习用的所述物质的晶体的原子的位置的位置数据、表示构成学习用的所述物质的晶体的原子的种类的种类数据、以及学习用的所述物质的晶体的晶格常数数据，所述学习用查询点是学习用的所述物质内的关注点，

[0229] 所述学习部在使用无监督机器学习使自动编码器进行学习时，

[0230] 通过对所述自动编码器中的编码器输入所述学习用晶体数据，取得表示所述学习用的物质的晶体结构的潜在向量，

[0231] 通过对所述自动编码器中的第1解码器输入所述潜在向量与所述学习用查询点的组合，取得表示构成所述学习用的物质的晶体的原子的位置的场的位置场，

[0232] 基于从所述第1解码器输出的所述位置场，估计构成所述学习用的物质的晶体的原子的位置，

[0233] 通过对所述自动编码器中的第2解码器输入所述潜在向量与所述学习用查询点的组合，取得表示构成所述学习用的物质的晶体的原子的种类的场的种类场数据，

[0234] 基于从所述第2解码器输出的所述种类场，估计构成所述学习用的物质的晶体的原子的种类，

[0235] 通过对所述自动编码器中的第3解码器输入所述潜在向量与所述学习用查询点的组合，取得所述学习用的物质的晶体的晶格常数数据，

[0236] 通过使用无监督机器学习使自动编码器进行学习，使得估计出的所述原子的位置、估计出的所述原子的种类以及从所述第3解码器输出的所述晶格常数数据的组合与所述学习用晶体数据中的所述位置数据、所述种类数据以及所述晶格常数数据的组合对应，生成已学习的第1解码器和已学习的第2解码器。

[0237] (附记9)

[0238] 一种信息处理方法，其由计算机执行以下处理：利用神经网络模型，表征表示由原子点群构成的物质结构的场。

[0239] (附记10)

[0240] 一种已学习模型生成方法，其由计算机执行以下处理：使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。

[0241] (附记11)

[0242] 一种信息处理程序，其用于使计算机执行以下处理：利用神经网络模型，表征表示由原子点群构成的物质结构的场。

[0243] (附记12)

[0244] 一种已学习模型生成程序，其用于使计算机执行以下处理：使表征表示由原子点群构成的物质结构的场的神经网络模型进行学习。

[0245] 在2022年11月21日申请的日本专利申请2022-186043号及在2023年8月9日申请的日本专利申请2023-130520号的公开整体通过参照在本说明书中被引用。在本说明书中通过参照引用记载于本说明书的全部文献、专利申请以及技术标准，引用的程度与具体且单独地描述通过参照引用各个文献、专利申请以及技术标准的情况相同。

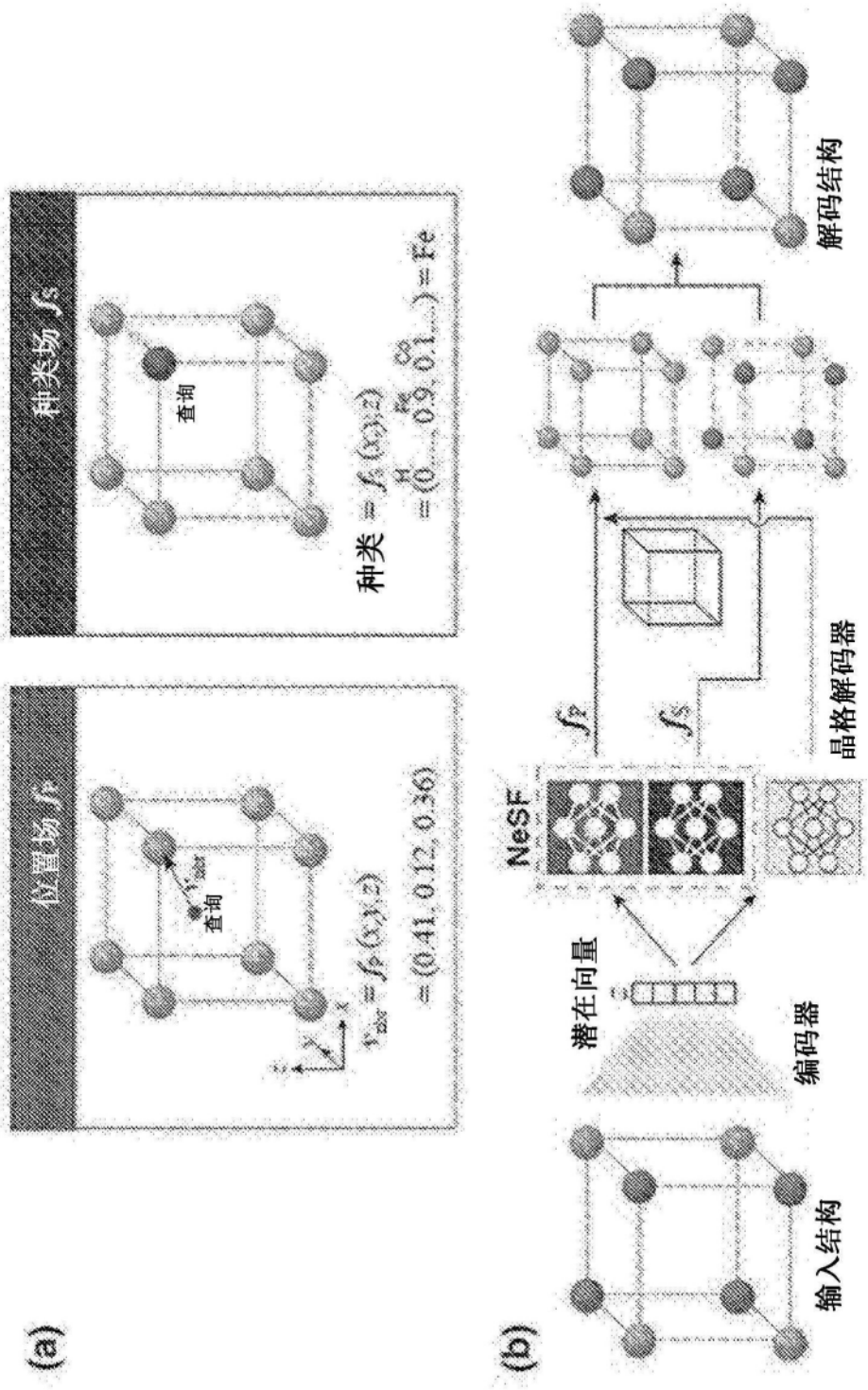

(a)
位置场 f_P
查询
r_{query}
r_{env}
$= f_P(x,y,z)$
$= (0.41, 0.12, 0.36)$
种类场 f_S
查询
种类 $= f_S(x,y,z)$
$= (0, \ldots, 0.9, \overset{C}{0.1}, \ldots) = \mathrm{Fe}$
(b)
输入结构
编码器
潜在向量
NeSF
f_P
f_S
晶格解码器
解码结构

图1

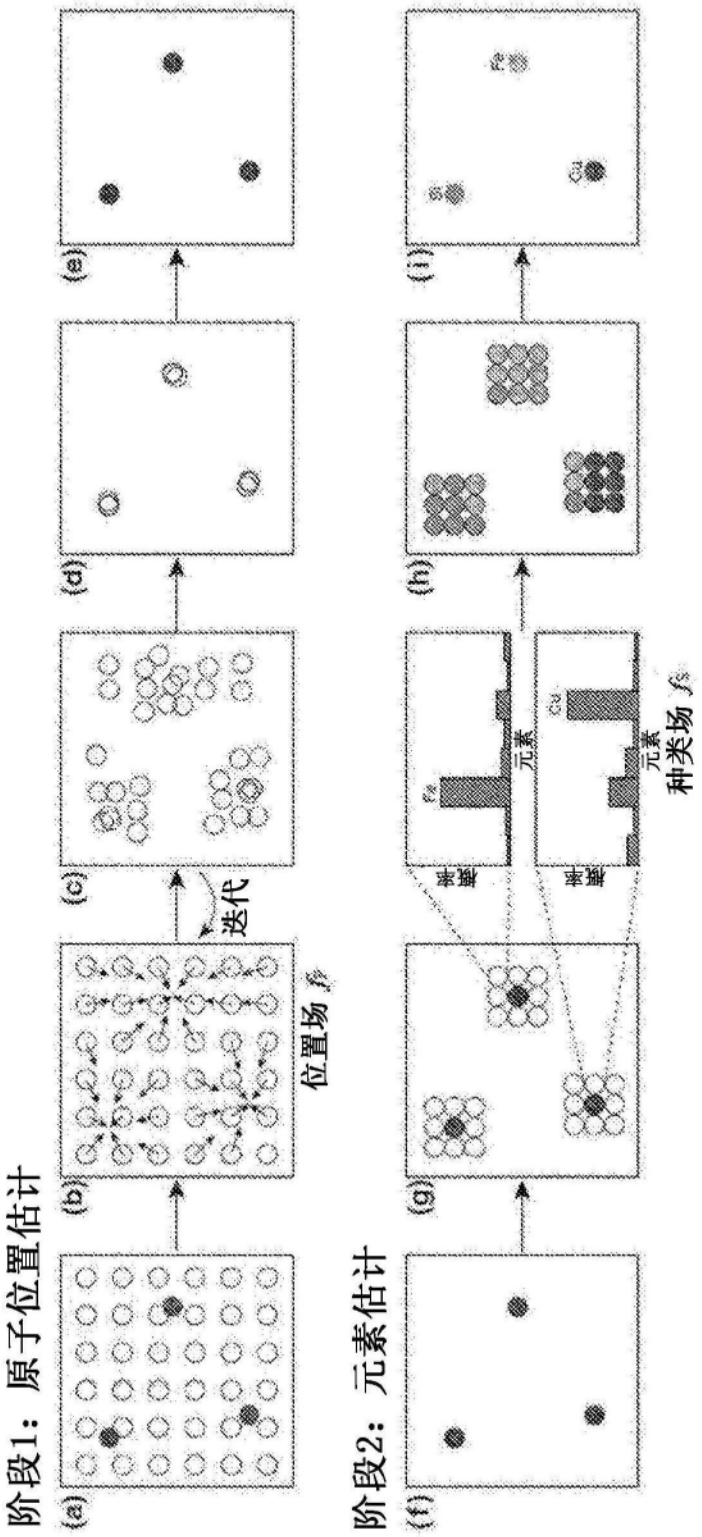


图2

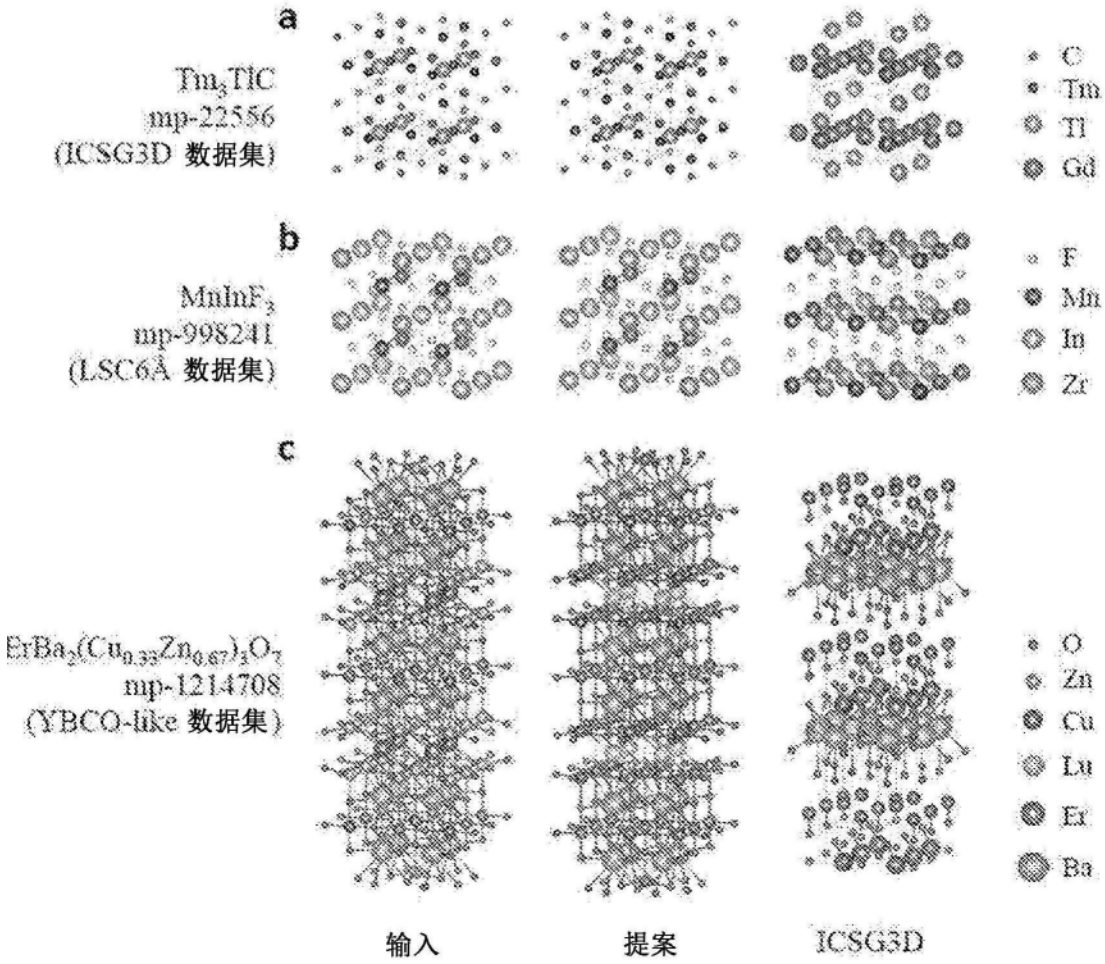

a
Tm_3TlC
mp-22556
(ICSG3D 数据集)
C
Tm
Tl
Gd
b
$MnInF_3$
mp-998241
(LSC6Å 数据集)
F
Mn
In
Zr
c
$ErBa_2(Cu_{0.33}Zn_{0.67})_3O_7$
mp-1214708
(YBCO-like 数据集)
O
Zn
Cu
Lu
Er
Ba
输入
提案
ICSG3D

图3

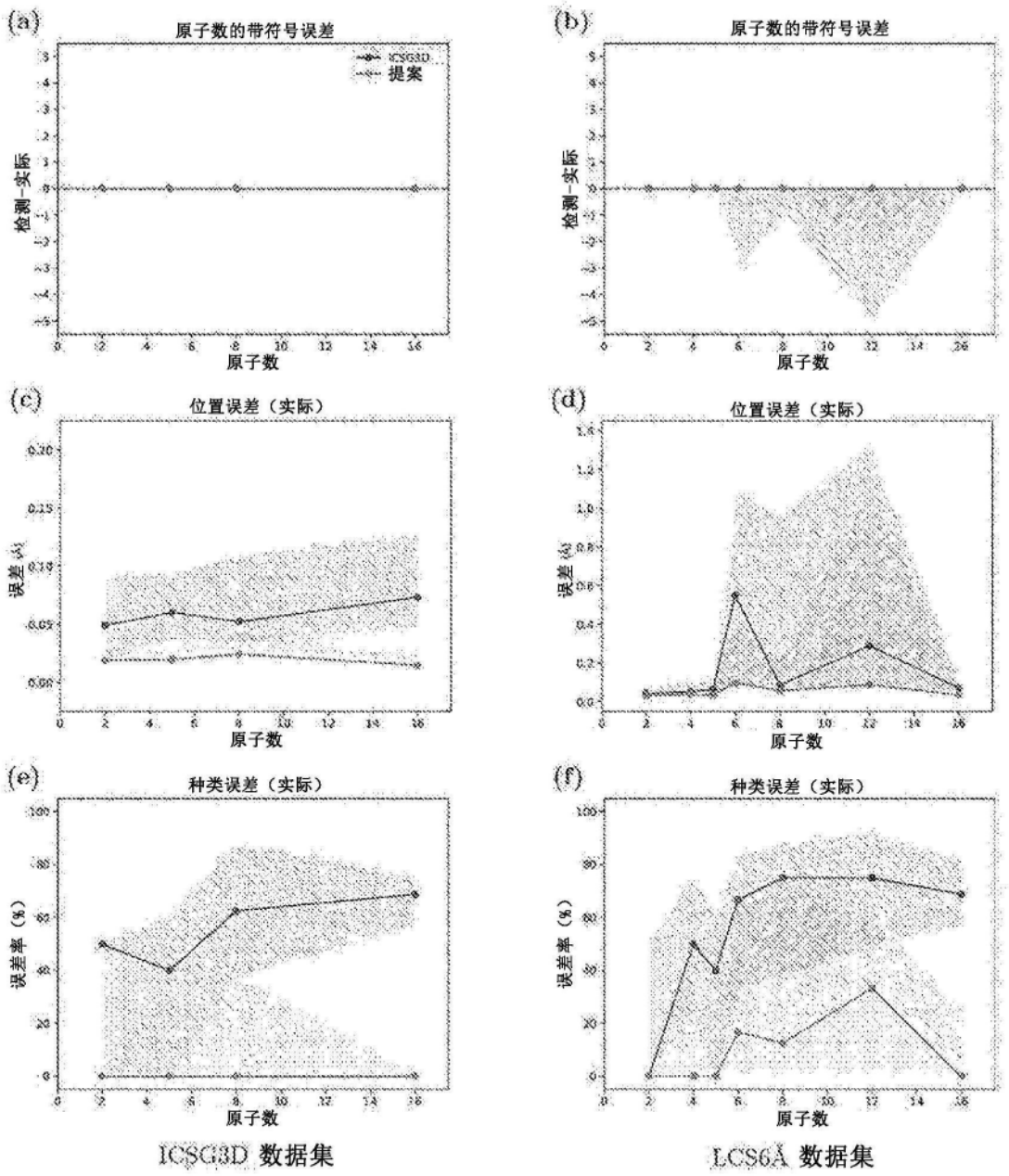

(a)
原子数的带符号误差
ICSG3D
提案
检测-实际
原子数
(b)
原子数的带符号误差
检测-实际
原子数
(c)
位置误差（实际）
误差 (Å)
原子数
(d)
位置误差（实际）
误差 (Å)
原子数
(e)
种类误差（实际）
误差率（%）
原子数
(f)
种类误差（实际）
误差率（%）
原子数
ICSG3D 数据集
LCS6A 数据集

图4

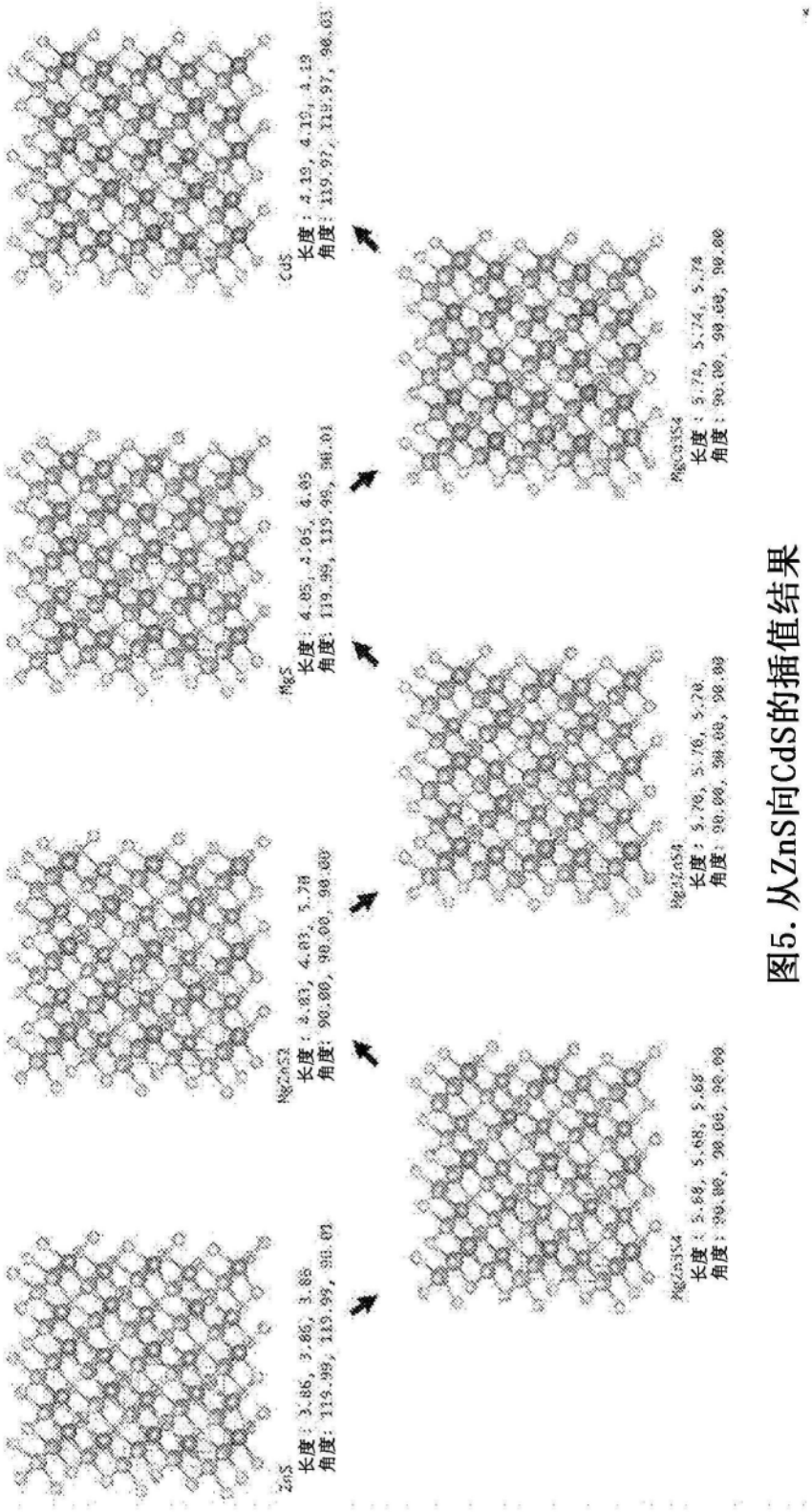


图5. 从ZnS向CdS的插值结果

图5

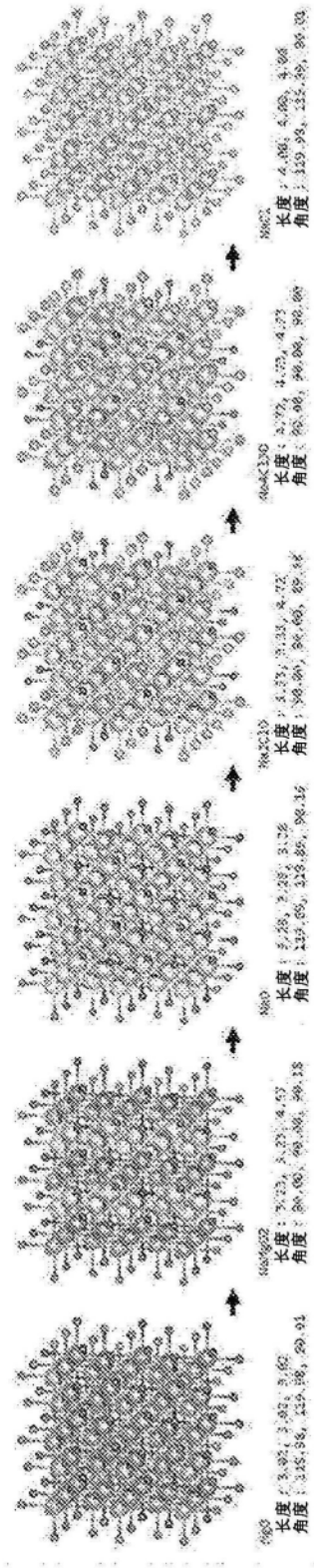


图6. 从MgO向NaCl的插值结果

图6

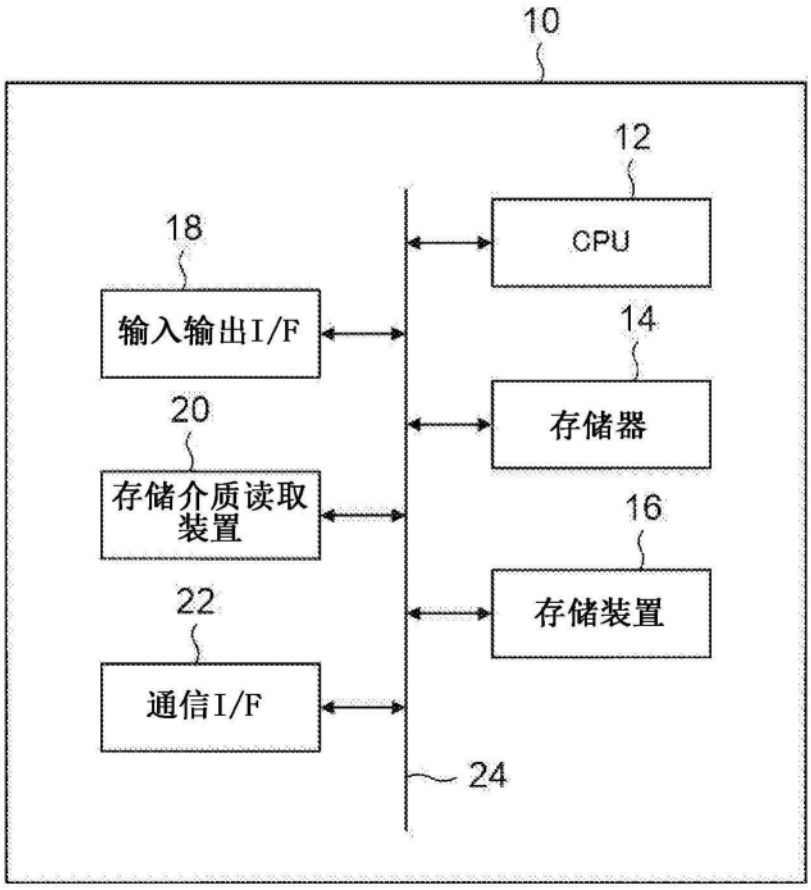


图7

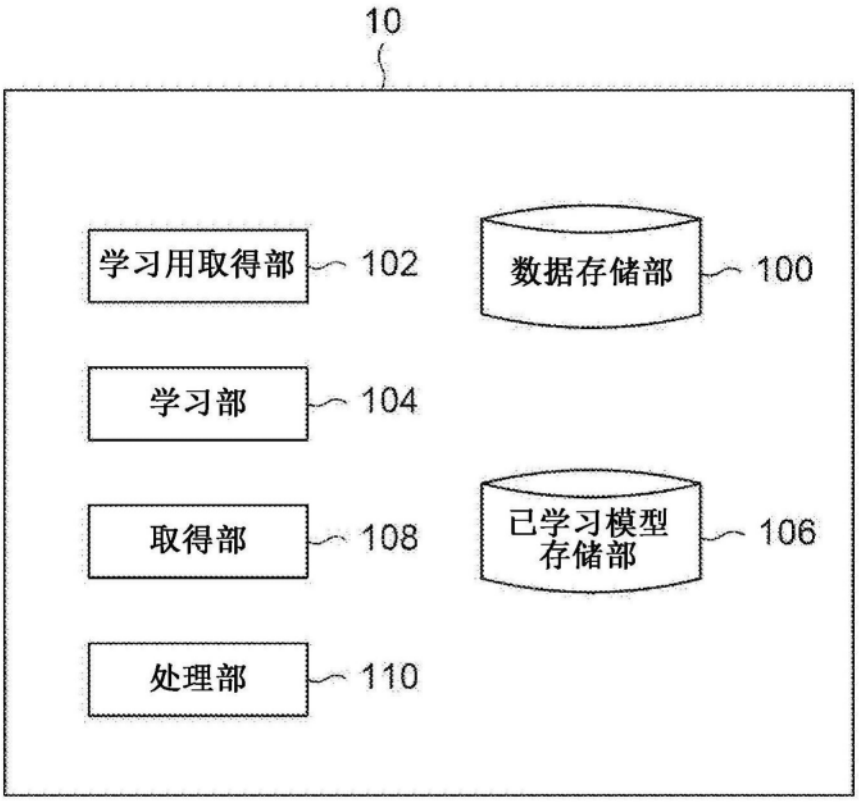


图8

晶体ID	位置数据	种类数据	晶格常数数据
00001	P1	S1	L1
00002	P2	S2	L2
00003	...	...	...
...	...	...	...

图9

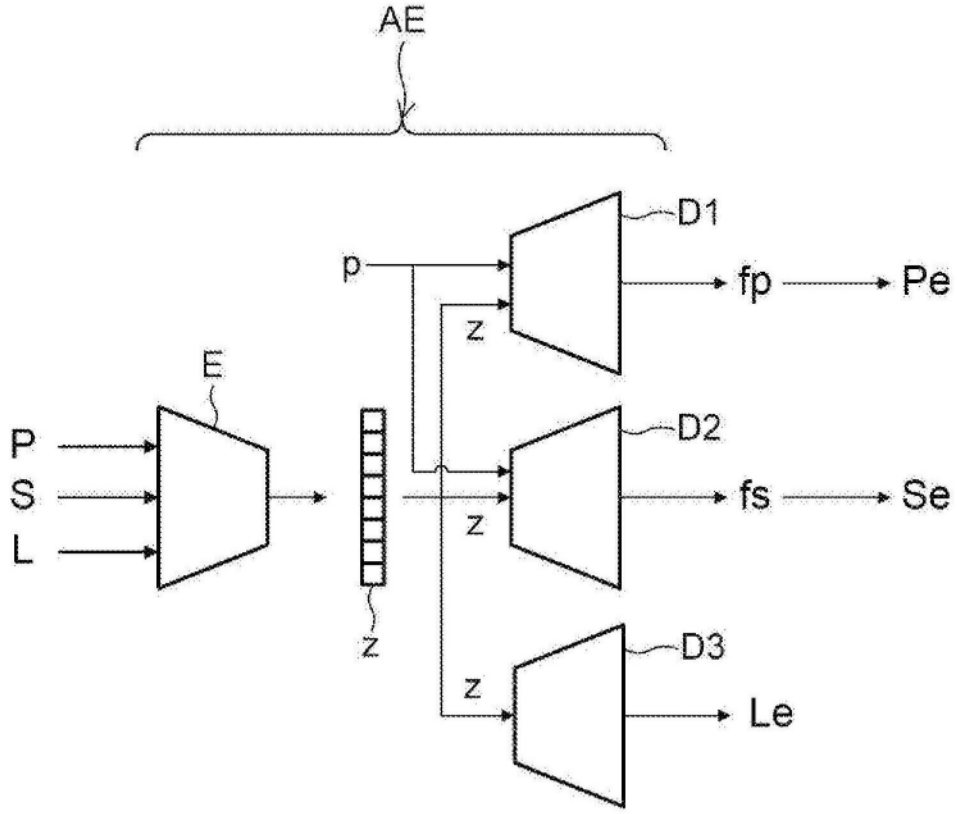


图10

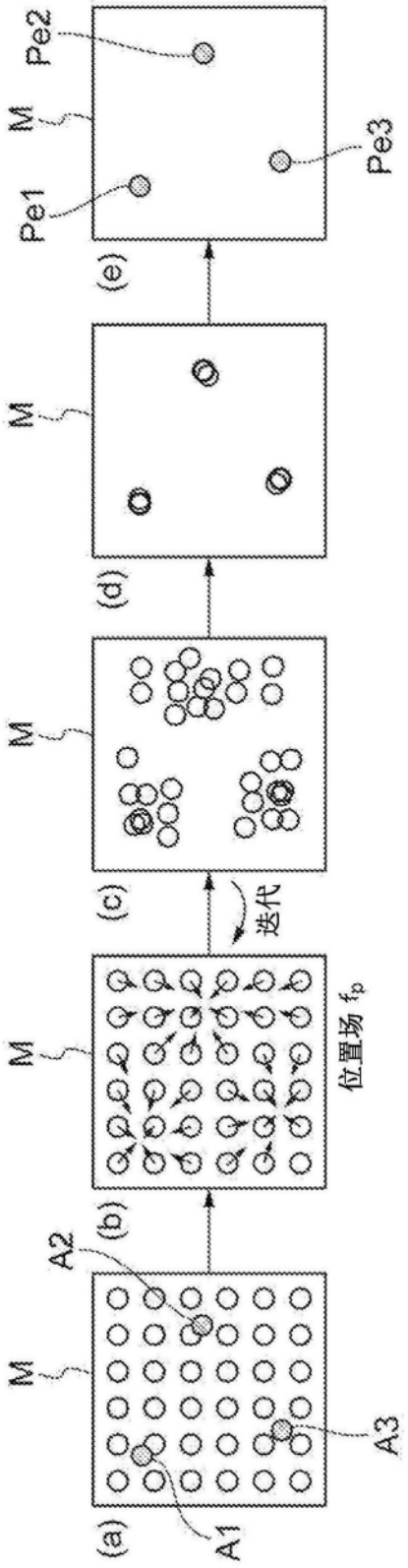


图11

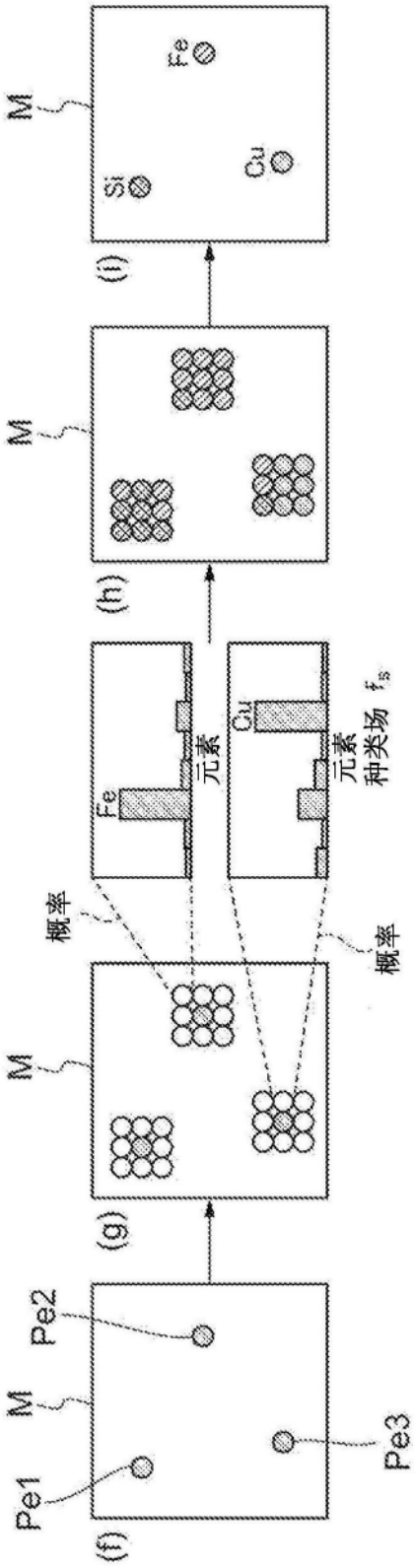


图12

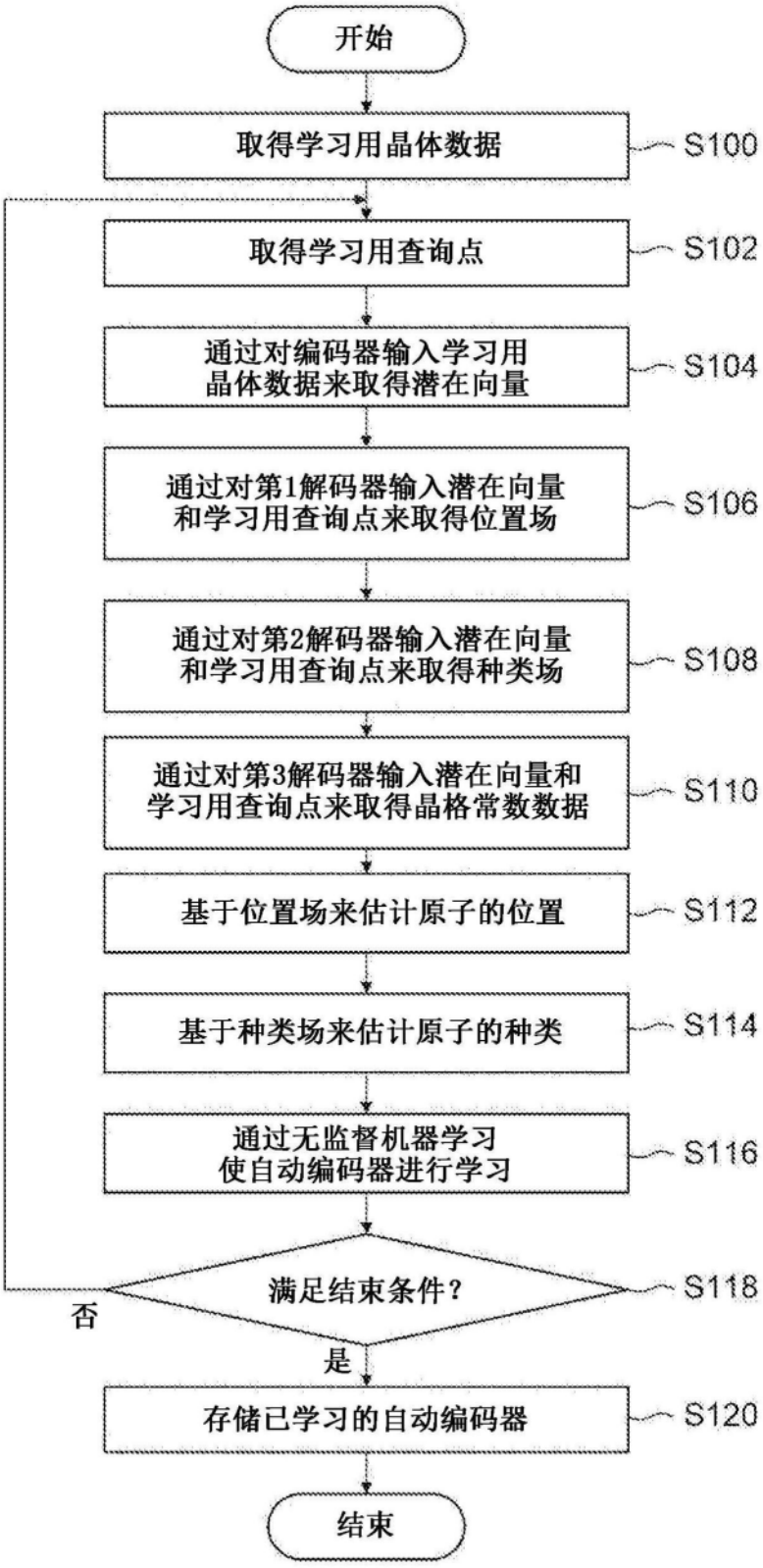

开始
取得学习用晶体数据
S100
取得学习用查询点
S102
通过对编码器输入学习用
晶体数据来取得潜在向量
S104
通过对第1解码器输入潜在向量
和学习用查询点来取得位置场
S106
通过对第2解码器输入潜在向量
和学习用查询点来取得种类场
S108
通过对第3解码器输入潜在向量和
学习用查询点来取得晶格常数数据
S110
基于位置场来估计原子的位置
S112
基于种类场来估计原子的种类
S114
通过无监督机器学习
使自动编码器进行学习
S116
满足结束条件？
S118
否
是
存储已学习的自动编码器
S120
结束

图13

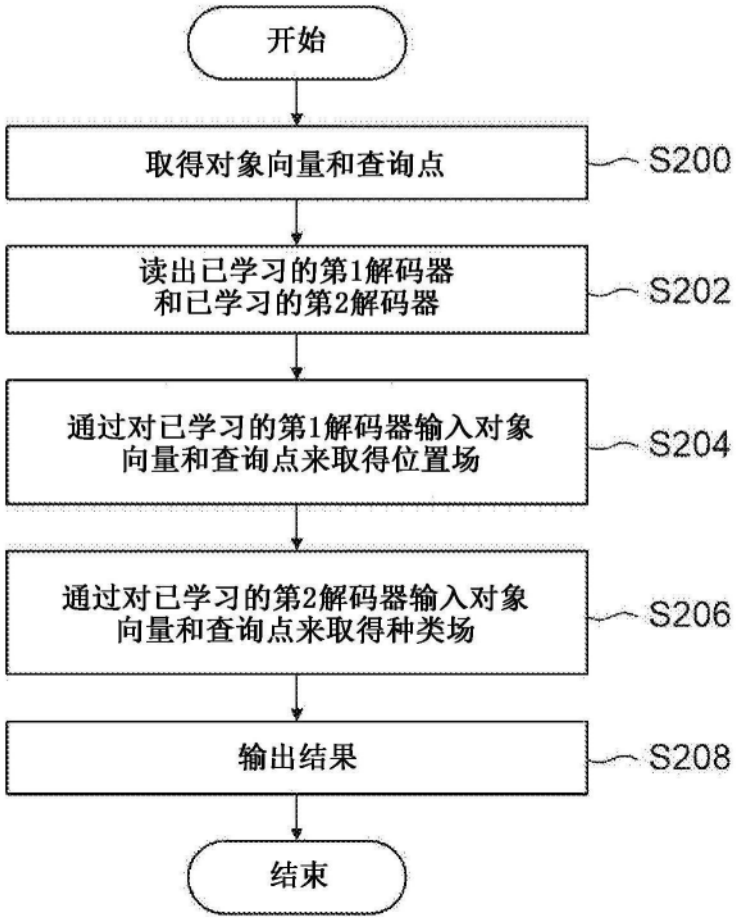

开始
取得对象向量和查询点
S200
读出已学习的第1解码器
和已学习的第2解码器
S202
通过对已学习的第1解码器输入对象
向量和查询点来取得位置场
S204
通过对已学习的第2解码器输入对象
向量和查询点来取得种类场
S206
输出结果
S208
结束

图14

ABSTRACT

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 next-generation methods that explore materials with desired properties without relying on "luck" or "serendipity." 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. To evaluate the NeSF, 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 Introduction

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)[3–9]. 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[8]. Applications of MI include the prediction of material properties from material characteristic data such as crystal structures[8] and compositions[8–11], automated analyses of experimental data[12–14], and natural language processing for knowledge retrieval from scientific literature[15].

Many MI studies have been focused on predicting properties of given materials, such as the bandgap, Seebeck coefficient, and elastic modulus[10–13]. 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[19–23].

1

图15

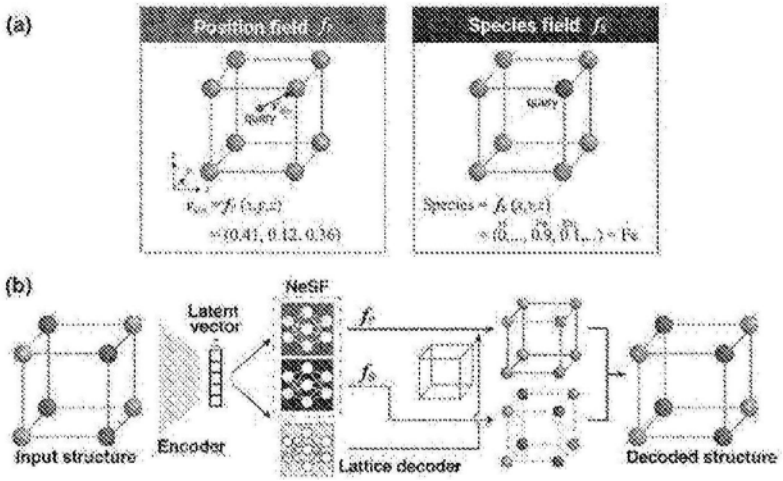


Figure 1. Overview of NeSF and crystal structure autoencoder using the NeSF. a, The NeSF consists of two vector fields, namely, position field f_p and species field f_s, which are defined in 3D space. Given a 3D point as a query, the position field is trained to represent the 3D vector pointing to the 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 $\mathbf{z}$ via a PointNet-based encoder[28,29]. 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 $\mathbf{z}$. The lattice constants are also estimated via the lattice decoder. These encoder and decoder are implemented as simple MLPs. See the Appendix for detailed definitions.

Despite the great potential utility of this inverse approach in developing materials, few studies have addressed this[19–23] or its underlying problem[24–26], 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[10,16,17,27], a technical bottleneck persists for 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[16], 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

图16

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:

$$s = f(p, z) \tag{1}$$

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[30–34] 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[31–33] and free-viewpoint image synthesis[30,34]. 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[31] 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[31–33].

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[18,20,22,24–26], 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[24] 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.

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

3/17

图17

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[46].

2 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 $\boldsymbol{z}$ (called *latent vectors*) via an encoder, and then the NeSF acts as a decoder to recover the input crystal structures from $\boldsymbol{z}$. 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 $\boldsymbol{z}$ 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.

2.1 Crystal structure estimation with NeSF

Given the target material information as vector $\boldsymbol{z}$, the estimation of the crystal structure from $\boldsymbol{z}$ 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 $\boldsymbol{z}$. On the other hand, the atomic positions and species are estimated by position field $\boldsymbol{f}_\mathrm{p}$ and species field $\boldsymbol{f}_\mathrm{s}$ of the NeSF, respectively, as described in the previous section. These fields are also implemented as simple MLPs, each taking query position $\boldsymbol{p}$ and vector $\boldsymbol{z}$ 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 $\boldsymbol{z}$ 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 $\{\boldsymbol{p}_i^0\}$, 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 $\boldsymbol{p}_i^t$ using the position field according to $\boldsymbol{p}_i^{t+1} = \boldsymbol{p}_i^t + \boldsymbol{f}_\mathrm{p}(\boldsymbol{p}_i^t, \boldsymbol{z})$. We iterate this process for all particles to obtain $\{\boldsymbol{p}_i\}$ 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 $\boldsymbol{p}_i$ and filter outliers. As the norm of the output of the position field, $\|\boldsymbol{f}_\mathrm{p}(\boldsymbol{p}_i, \boldsymbol{z})\|$, indicates an estimated distance from $\boldsymbol{p}_i$ to its nearest atom, we score each particle $\boldsymbol{p}_i$ by $\|\boldsymbol{f}_\mathrm{p}(\boldsymbol{p}_i, \boldsymbol{z})\|$ and discard the particle if the score is above a certain threshold (set to 0.9 Å in this study).

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

图18

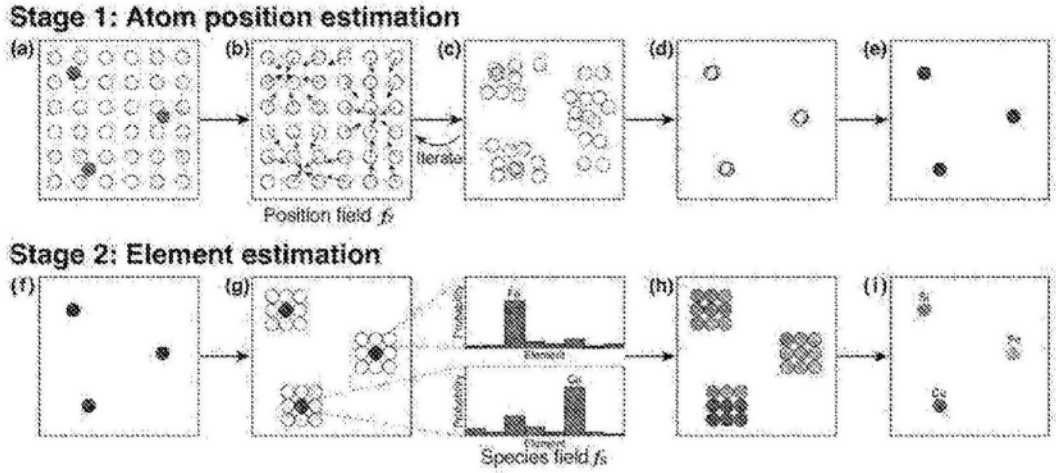


Figure 2. Algorithm for crystal structure estimation with the NeSF. The algorithm consists of two stages: 1) atomic position estimation (corresponding to steps 1–4 in Section 2.1) and 2) element estimation (corresponding to step 5 in Section 2.1). **a**, To detect true atomic positions (shown as red 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.

$B_c = \{\boldsymbol{p}_i\}$ 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. (The spherical radius was set to 0.5 Å in this study.) By repeating these steps until B_c becomes empty, we obtain atomic positions $\{\boldsymbol{a}_i\}$ as the selected particles stored in B_a.

5. **Estimate species.** Finally, we use species field $\boldsymbol{f}_s(\boldsymbol{p}, z)$ to estimate the atomic species at each atomic position $\boldsymbol{a}_i$. For robust estimation against errors in $\boldsymbol{a}_i$, we spread new particles intensively around each $\boldsymbol{a}_i$ as queries to the species field instead of using $\boldsymbol{a}_i$ directly as a query. Hence, we obtain multiple probability distributions, each predicting the species of atom $\boldsymbol{a}_i$. We select the most frequent atomic species among them as the final estimate.

2.2 Training of NeSF

The training of the NeSF differs from the above estimation algorithm and is much simpler. We randomly sample 3D query points $\{\boldsymbol{p}'_i\}$ in the unit cell and compute loss values for the field outputs at these points, thus supervising $\boldsymbol{f}_p(\boldsymbol{p}'_i, z)$ and $\boldsymbol{f}_s(\boldsymbol{p}'_i, z)$ 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[31], sample the training query points near the surface. Curriculum DeepSDF[35] 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 to the position field toward their nearest

图19

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 abovementioned 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 centered 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 neighborhood of atoms. The parameters used for sampling are provided in the Supplementary Information (SI).

2.3 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 $\boldsymbol{z}$. The decoder, for which we use the NeSF, reconstructs the input crystal structure from latent vector $\boldsymbol{z}$. 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 ML. 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*[28,29]. Among them, the family of graph neural networks[10,18,17,27] 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 ensure the invariance to coordinate systems, its information loss in input may unintentionally hinder the reconstruction performance. Thus, we adopt the basic encoder architecture from PointNet[28] and DeepSets[29]. 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 Section 4.2 and in the SI.

Training and evaluation procedures

We trained and evaluated the autoencoder and ICSG3D[24] (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 the Materials Project and are designed to have different difficulty levels. The ICSG3D dataset is a materials collection used by Court et al.[24] 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 Section 4.1.

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

图20

Table 1. Reconstruction results

Method	Proposed	ICSG3D	Proposed	ICSG3D	Proposed	ICSG3D
Dataset	ICSG3D		LCS6Å		YBCO-like	
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	**0.1101 ± 0.0073**	0.2086 ± 0.0132	**0.2631**	(0.6311)
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) [%]	**4.30 ± 0.39**	65.05 ± 1.85	**16.11 ± 0.66**	58.32 ± 1.45	**19.89**	54.34

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 the displayed value.

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 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 but evaluated it once instead of 10 times. 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 Section 4.3.

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. 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. 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. Fig. 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 (approximately 65% in both actual and detected metrics). By contrast, the proposed method achieves slightly better position error and error in 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

图21

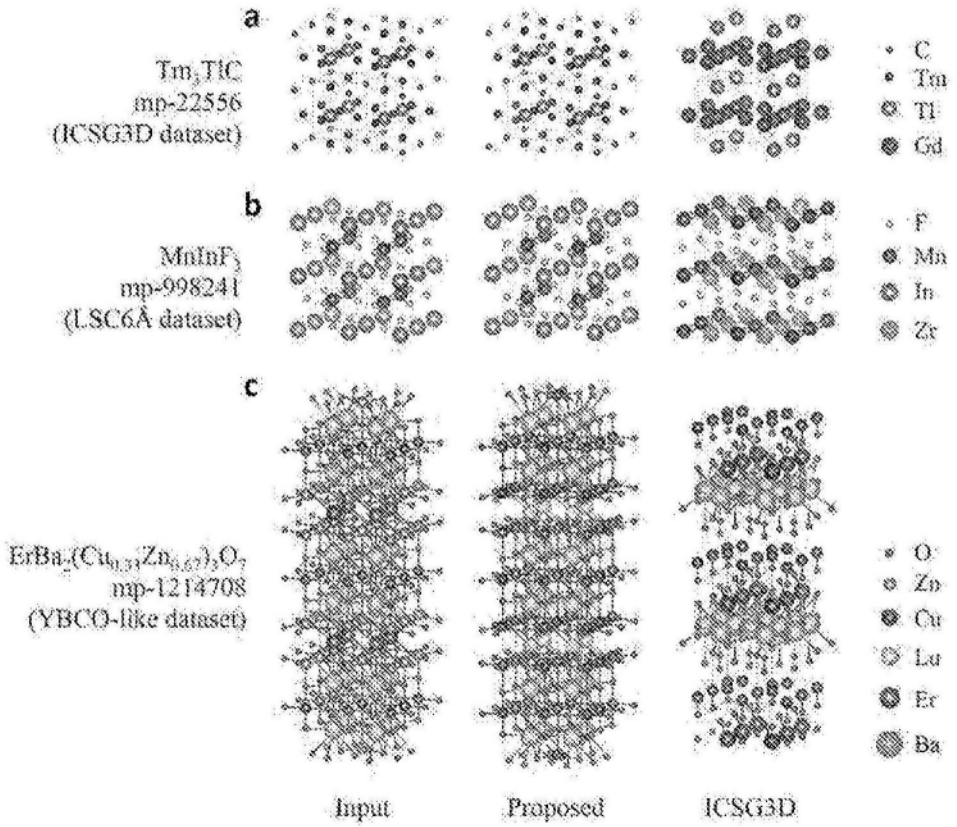


Figure 3. Reconstruction examples. For the three evaluated datasets (rows), we show test samples (first column) and reconstruction results by the proposed (second column) and ICSG3D (third column) methods. The full results are available online.

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 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 number of atoms given as medians (points) and 68-percent 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 atoms in their unit cells. Figs. 4a and 4b show the signed errors between the

8/17

图22

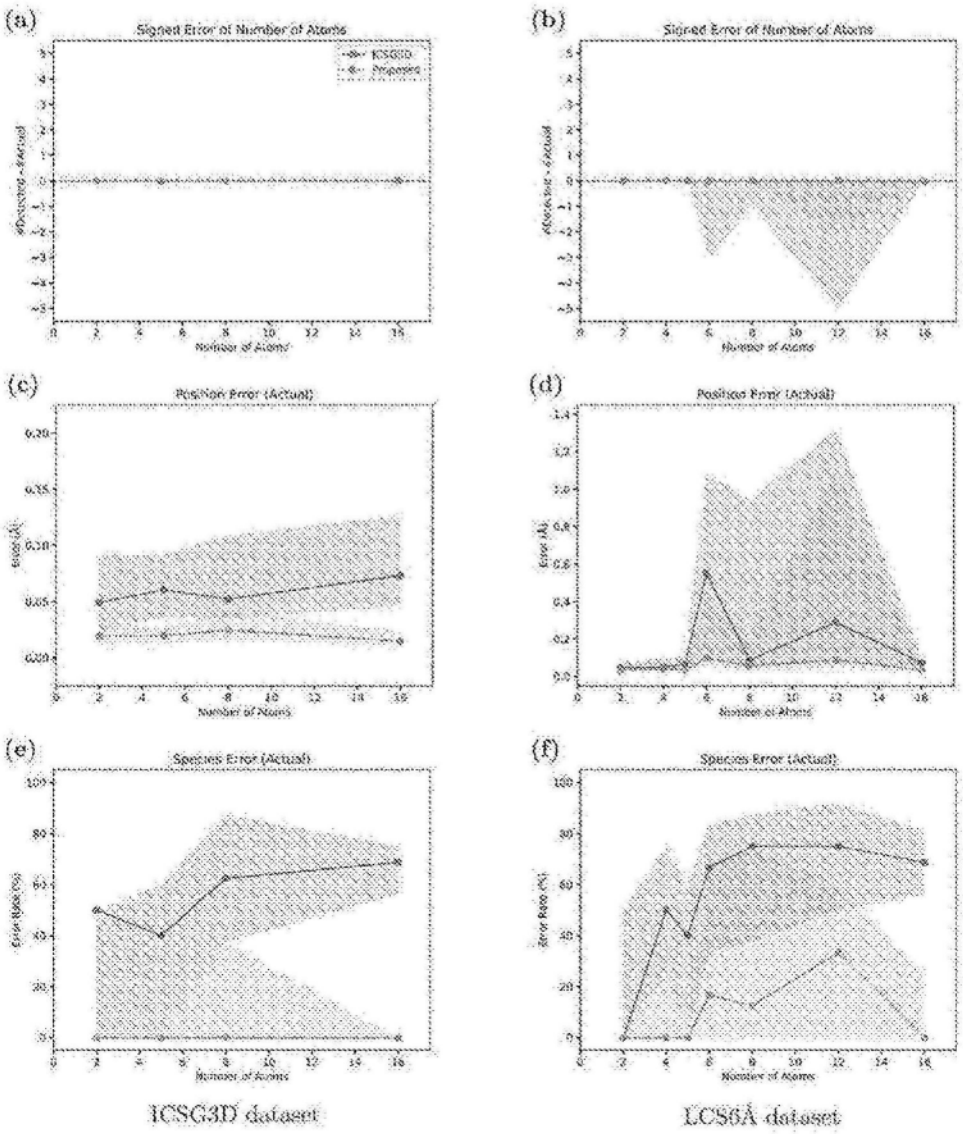


Figure 4. Relationships between reconstruction performance and number of atoms in conventional unit cell. Three kinds of error distributions (rows) are shown for the ICSG3D (first column) and LCS6Å (second column) datasets described by medians (points) and 68-percent ranges (colored regions) similar to means $\pm\sigma$ ranges. **a, b**, Distributions of signed errors between detected and actual numbers of atoms. **c, d**, Distributions of position errors according to 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 less than 10 materials samples from the visualization (see Section 4.1 for detailed dataset statistics).

9/17

图23

numbers of detected and actual atoms. These results indicate that both methods correctly estimate the number of atoms for most (*i.e.*, > 68 percent) 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, Figs. 4c and 4d show the distributions of position errors and Figs. 4e and 4f 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 more than 68 percent 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, being 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 indirectly describe the 3D space 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.

2.4 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 $\mathbf{z}_{\mathrm{src}}$ and $\mathbf{z}_{\mathrm{dst}}$ via the trained encoder.
2. Interpolate between $\mathbf{z}_{\mathrm{src}}$ and $\mathbf{z}_{\mathrm{dst}}$ 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.

图24

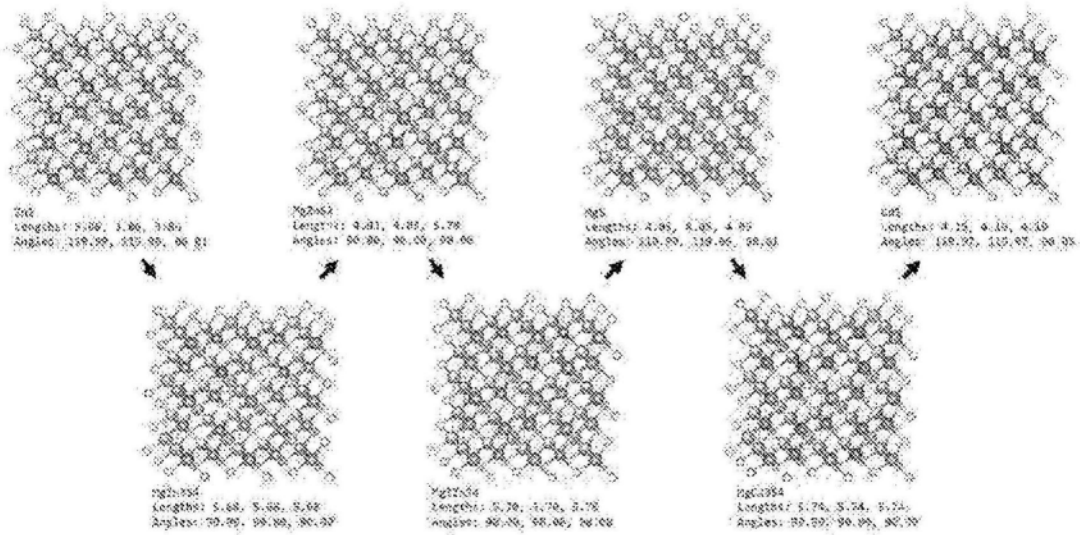


Figure 5. Results of interpolation from ZnS to CdS.

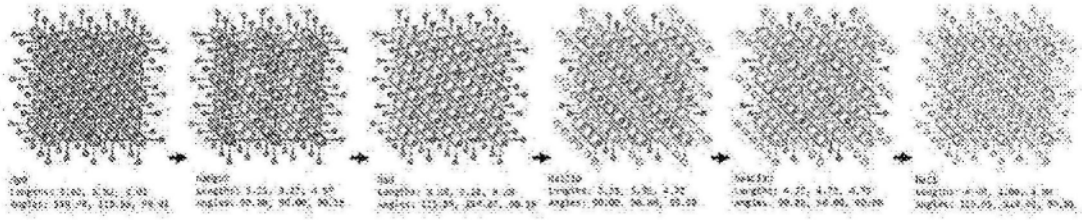


Figure 6. Results of interpolation from MgO to NaCl.

To facilitate 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 → $MgZn_3S_4$ ($Mg_{0.25}Zn_{0.75}S$) → $MgZnS_2$ ($Mg_{0.5}Zn_{0.5}S$) → Mg_3ZnS_4 ($Mg_{0.75}Zn_{0.25}S$) → MgS → $MgCd_3S_4$ ($Mg_{0.25}Cd_{0.75}S$) → 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 → $NaMgO_2$ ($Na_{0.5}Mg_{0.5}O$) → NaO → Na_2ClO ($NaCl_{0.5}O_{0.5}$) → Na_4Cl_3O ($NaCl_{0.75}O_{0.25}$) → NaCl.

Additionally, we provide the results of interpolation from NaCl (mp-22862) to PbS, from MgO (mp-1265) to CaO (mp-2605), and from PbS (mp-21276) to CaO (mp-2605) in the SI.

In the interpolation examples, 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.

11/17

图25

2.5 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 NeSF adopts a relatively simple network architecture, and architectural design choices were not thoroughly explored. For example, the NeSF treats the elements as mutually independent categorical (one-hot) vectors. Thus, we do not explicitly train our model with the physical characteristics of elements or their similarities. Meanwhile, other developments have attempted to explicitly inject physical characteristics and features of elements into their models. For instance, existing crystal-structure encoders[18] represent input elements using their fingerprints, such as their groups and periods, electron densities, atomic radii, and electronegativities, instead of using one-hot vectors. In ICSG3D[24], the outputs of atomic species are trained using the mean squared error of the atomic numbers rather than a categorical loss, which is considered in our method. Incorporating these schemes into our model may allow to reflect the characteristics and similarities between elements in the latent space and reconstructed crystal structures. Another architectural choice regarding the NeSF is whether to use the conventional (current choice) or primitive unit cell for expressing crystal structures. Both formats have their advantages and drawbacks depending on the application. The conventional unit cell offers more intuitive visualizations that may facilitate manual analysis, whereas the primitive unit cell provides more compact structure representations that may facilitate computer processing. A thorough analysis of the network architecture design should be performed in future work.

Another limitation is that 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[24] 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.

Finally, for an evaluation purpose, we adopted the autoencoder architecture rather than generative models, such as variational autoencoders[16,25,37] and generative adversarial networks[26,37–39]. 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 analysis. The NeSF should be applied to generative models in future work with appropriate performance analysis.

3 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 representations 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. A 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.

图26

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 next-generation development of materials. For instance, the NeSF can be easily incorporated into 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 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 next-generation development of materials.

4 Methods

4.1 Datasets

We evaluated the proposed NeSF-based autoencoder using datasets from the Materials Project.

- ICSG3D dataset: To compare with the ICSG3D baseline, we reproduced the dataset used in the original paper[23]. 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. 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 the Materials Project, 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, 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 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 of 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.

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

No. atoms per cell	2	3	6	8	12	15	16	20	32	40
Training	381	1413	7	372	1	12	4616	17	18	36
Validation	18	66		32			223	1		2
Test	24	61		33			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	1335	244	69	1886	58	137	9	892	8	23	4	882	5	6	1	3	9	1
Validation		35	2	80	61	14	3	91	4	5		13		2		20					1	
Test		33	3	83	68	17	2	85		5		19		1		28						

13/17

图27

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

No. atoms per cell	13
Training	89
Validation	5
Test	5

4.2 Neural network architecture

The proposed NeSF-based autoencoder consists of simple MLPs. Spacifically, 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 the SI.

4.3 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:

$$L = \lambda_{\mathrm{pos}} L_{\mathrm{pos}} + \lambda_{\mathrm{spe}} L_{\mathrm{spe}} + \lambda_{\mathrm{len}} L_{\mathrm{len}} + \lambda_{\mathrm{ang}} L_{\mathrm{ang}} \tag{2}$$

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 $(\lambda_{\mathrm{pos}}, \lambda_{\mathrm{spe}}, \lambda_{\mathrm{len}}, \lambda_{\mathrm{ang}}) = (10, 0.1, 1, 1)$. 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 epochs for all the materials 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 on the LCS6Å dataset took approximately 9 h using a computer equipped with a single NVIDIA 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 the SI.

References

1. De Graef, M. & McHenry, M. E. *Structure of Materials.* An Introduction to Crystallography, Diffraction and Symmetry (Cambridge University Press, 2012).
2. Callister, W. D. & Rethwisch, D. G. *Materials Science and Engeneering* (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. DOI: 10.1038/s41524-019-0221-0 (2019)

图28

5. Szymanski, N. J. *et al.* Toward autonomous design and synthesis of novel inorganic materials. *Mater. Horiz.* **8**, 2169–2198, DOI: 10.1039/D1MH00495F (2021).

6. Choudhary, K. *et al.* Recent advances and applications of deep learning methods in materials science. *npj Comput. Mater* **8**, 1–26, DOI: 10.1038/s41524-022-00734-6 (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, DOI: 10.1063/1.4946894 (2016).

8. Jha, D. *et al.* ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition. *Sci. Reports* **8**, 17593, DOI: 10.1038/s41598-018-35934-y (2018).

9. Goodall, R. E. A. & Lee, A. A. Predicting materials properties without crystal structure: Deep representation learning from stoichiometry. *Nat Commun* **11**, 6280, DOI: 10.1038/s41467-020-19964-7 (2020).

10. Park, C. W. & Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. *Phys. Rev. Mater.* **4**, 063801, DOI: 10.1103/PhysRevMaterials.4.063801 (6 1, 2020).

11. 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, DOI: 10.1038/s41524-021-00545-1 (2021).

12. Park, W. B. *et al.* Classification of crystal structure using a convolutional neural network. *IUCrJ* **4**, 486–494, DOI: 10.1107/S205225251700714X (2017).

13. Oviedo, F. *et al.* Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. *npj Comput. Mater.* **5**, 60 (2019).

14. 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, DOI: 10.1021/acs.chemmater.1c01071 (2021).

15. Tshitoyan, V. *et al.* Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98, DOI: 10.1038/s41586-019-1335-8 (2019).

16. Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **120**, 145301, DOI: 10.1103/PhysRevLett.120.145301 (2018).

17. Cheng, J., Zhang, C. & Dong, L. A geometric-information-enhanced crystal graph network for predicting properties of materials. *Commun Mater* **2**, 1–11, DOI: 10.1038/s43246-021-00194-3 (2021).

18. Choudhary, K. & DeCost, B. Atomistic Line Graph Neural Network for improved materials property predictions. *npj Comput. Mater* **7**, 1–8, DOI: 10.1038/s41524-021-00650-1 (2021).

19. Noh, J. *et al.* Inverse Design of Solid-State Materials via a Continuous Representation. *Matter* **1**, 1370–1384, DOI: 10.1016/j.matt.2019.08.017 (2019).

20. 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, DOI: 10.1039/D0SC00594K (2020).

21. Yao, Z. *et al.* Inverse design of nanoporous crystalline reticular materials with deep generative models. *Nat. Mach. Intell.* **3**, 76–86, DOI: 10.1038/s42256-020-00271-1 (2021).

22. Long, T. *et al.* Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. *npj Comput. Mater* **7**, 1–7, DOI: 10.1038/s41524-021-00526-4 (2021).

15/17

图29

23. 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, DOI: 10.1038/s41524-021-00670-z (2021).

24. 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, DOI: 10.1021/acs.jcim.0c00464 (2020).

25. Hoffmann, J. *et al.* Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949* (2019).

26. 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, DOI: 10.1021/acscentsci.0c00426 (2020).

27. 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, DOI: 10.1021/acs.chemmater.9b01294 (2019).

28. Charles, R. Q., Su, H., Kaichun, M. & Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

29. Zaheer, M. *et al.* Deep sets. In *Neural Information Processing Systems* (2017).

30. Mildenhall, B. *et al.* NeRF: Representing scenes as neural radiance fields for view synthesis. *arXiv* (2020). 2003.08934.

31. Park, J. J., Florence, P., Straub, J., Newcombe, R. & Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

32. Chen, Z. & Zhang, H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

33. 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 (CVPR)* (2019).

34. Xie, Y. *et al.* Neural fields in visual computing and beyond. *Comput. Graph. Forum* DOI: 10.1111/cgf.14505 (2022).

35. Duan, Y. *et al.* Curriculum deepsdf. In Vedaldi, A., Bischof, H., Brox, T. & Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020*, 51–67 (Springer International Publishing, Cham, 2020).

36. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)* (2015).

37. Goodfellow, I. J., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, Cambridge, MA, USA, 2016). http://www.deeplearningbook.org.

38. Goodfellow, I. *et al.* Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates, Inc., 2014).

39. Nouira, A., Sokolovska, N. & Crivello, J.-C. CrystalGAN: Learning to Discover Crystallographic Structures with Generative Adversarial Networks. *arXiv:1810.11203 [cs, stat]* (2019). 1810.11203.

Data availability

The codes, documents, and supplementary resources that support the findings of this study are available at https://hogehoge.

图30