SISSO 方法的最新进展及其在 SISSO++ 代码中的实现

Thomas A. R. Purcell,[1, a)] Matthias Scheffler,[1] 和 Luca M. Ghiringhelli[2, 1, b) 1)]
The NOMAD Laboratory at the FHI of the Max-Planck-Gesellschaft and IRIS-Adlershof of the Humboldt-Universität zu Berlin, Faradayweg 4–6, D-14195 Berlin,Germany[2)]
Physics Department and IRIS-Adlershof, Humboldt Universität zu Berlin, Zum Großen Windkanal 2 D-12489 Berlin,Germany

（日期：2023年5月3日）

准确且可解释的人工智能（AI）模型有望成为加速发现新材料或为现有材料寻找新应用的有力工具。近年来，符号回归已成为可解释人工智能领域越来越受欢迎的工具，因为它能够为目标属性生成相对简单的解析描述。确定性独立筛选与稀疏算子（SISSO）方法具有确定性这一特点，因此尤其适用于此类应用。本文介绍 SISSO 算法的最新进展，这些进展已在 SISSO++ 中实现。SISSO++ 是一款带有 Python 绑定的 C++ 代码。我们提出了 SISSO 所发现数学表达式的一种新表示方式，这是在特征构造步骤中引入“语法”规则的第一步。重要的是，通过在特征构造步骤中引入受控的非线性优化，我们扩展了该方法能够找到的描述符范围。最后，我们改进了回归和分类所用的求解器算法，大幅提高了 SISSO 的可靠性和效率。对于 SISSO 算法的所有这些基础性改进，我们不仅展示了它们可能产生的影响，还详细阐述了它们在数学和计算层面的运作方式。

arXiv:2305.01242v1 [physics.data-an] 2 May 2023

I. 引言

以数据为中心的方法和人工智能（AI）方法正成为描述物理和化学性质及过程的重要工具。人工智能的主要优势在于，无需事先了解哪些性质至关重要，就能发现不同性质集合之间的相关性。因此，人工智能在材料发现领域日益受到青睐，应用涵盖热输运性质[1,2], 催化[3–5]相稳定性[6], 以及量子材料[7]等方面。尽管这些方法取得了成功，如何构建可解释且具有物理相关性的人工智能模型，仍是该领域尚待解决的挑战[8–10]。

一种常见的可解释人工智能方法是符号回归[11–14]。符号回归算法从一组输入特征（即初级特征）中，为给定的目标属性找出最优的非线性解析表达式；这些输入特征可能与目标相关[15]。最初，人们使用基于（随机）遗传编程的方法来寻找这些表达式，这类方法至今仍在使用[15–18]；但近年来，求解器的种类更加多样[19–24]。sure-independence screening and sparsifying operator (SISSO) 方法将符号回归与压缩感知相结合[25–28]，从而提供了一种确定性方法来寻找这些解析表达式。该方法已用于描述多种属性，包括相稳定性[6,26,29]，催化[30]，和玻璃化转变温度——特征[31]。它也已被用于多任务[25] 和层级化[28]方式。

SISSO方法以一组初级特征以及数学上的一元和二元算子（例如，加法、乘法、 n次方根、对数函数等）为起点。第一步是特征创建步骤，在此过程中，通过穷尽地将这组数学算子应用于初级特征，构建生成特征池。随后，通过将这组算子应用于初级特征或上一步创建的生成特征，迭代重复该算法。特征创建步骤中的迭代次数称为层级。后续步骤是描述符识别，即通过在所有生成表达式的子空间 $\mathcal{S}$ 上执行 ℓ_0正则化优化，利用压缩感知来识别最佳的 n维线性模型。 $\mathcal{S}$通过确定性独立筛选[32], 进行选择，并采用适当的投影得分，具体取决于所求解的是回归问题还是分类问题（见下文）。对于回归问题，该方法根据皮尔逊相关值对所有生成特征进行排序，仅选择相关性最高的特征，本质上是在所有特征上执行一维 ℓ_0正则化线性回归。SISSO分析的结果是一个 $n-$维描述符，它是一个分量取自 $\mathcal{S}$的向量。对于回归问题，SISSO模型是所识别描述符与线性系数向量的标量积，该线性系数向量由ℓ_0正则化线性回归得到。对于分类问题，模型表示为一组超平面，这些超平面将点划分为不同类别；这些类别由所识别描述符与一组系数的标量积进行描述，而这组系数通过线性支持向量机（SVM）确定。

a)电子邮件：purcell@fhi-berlin.mpg.deb)电子邮件：ghiringhelli@fhi-berlin.mpg.de

本文介绍近期发布的 SISSO++代码[27] 中实现的新概念，并详细说明其实现方式。除了创建用于运行 SISSO 的模块化接口外，SISSO++ 还在多个方面改进了算法，包括特征创建和描述符识别步骤。该代码最重要的进展是将特征表示为二叉表达式树，而不是字符串，从而能够根据初级特征递归地定义生成表达式的所有方面。采用这种实现方式后，我们能够完整描述每个生成特征的单位，并初步表示其定义域和值域。这使得我们能够创建在物理单位一致性方面符合语法的表达式，并控制特征超出其物理意义范围所引发的数值问题。在特征创建步骤方面，我们还讨论了参数化 *SISSO*的实现。该方法引入了非线性参数的灵活性，同时结合使用算子，并根据基于压缩感知的特征选择指标所构建的损失函数进行优化。近期一篇出版物[33]使用了这一过程来描述一种材料的热导率。对于描述符识别步骤，我们涵盖两个组成部分：改进的分类算法和多残差方法。对于分类问题，我们将算法推广到适用于任意问题和任意维度，并明确纳入通过线性支持向量机进行的模型识别。多残差方法此前曾在文献[28], 中使用，它为识别具有多个维度的模型提供了更大的灵活性。本文将深入讨论其机制。

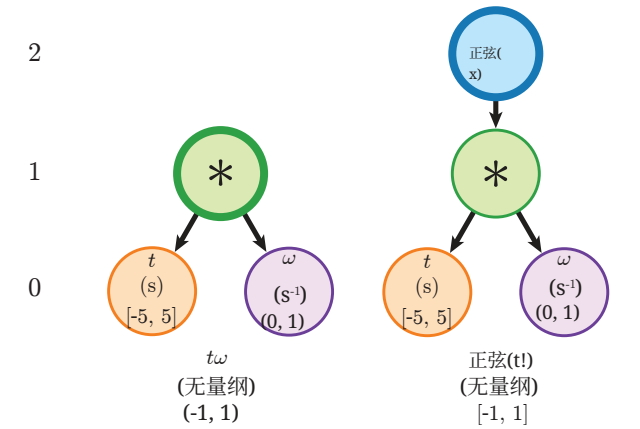

图 1. SISSO++ 代码中特征新表示方式的示意图。特征存储为树的根节点（以粗边框表示），初级特征为叶节点，而层级对应于树的高度，即每个叶节点与根节点之间的最长路径。特征的单位、范围、表达式和值仅需为初级特征存储；所有生成表达式的这些属性均以递归方式定义。

II. 特征构造

A. 特征的二叉表达式树表示

与参考文献[26], 中的原始实现相比，SISSO 在 SISSO++ 中的实现所取得的最大改进，是将特征表示为二叉表达式树，而非字符串。这种表示如图 1 所示，并且能够轻松地从初级特征中存储的数据出发，递归地即时计算生成特征的所有属性。对于某些应用，还可以存储更高层级特征的数据，以减少开销。

所有计算的计算成本。各个特征由二叉表达式树的根节点寻址，并在代码中存储为 C++标准库中的 SHARED PTR。这种表示方式降低了每次计算的总体内存占用，因为各个特征只需创建一次，而每生成一个新表达式，只需存储共享指针的副本。本节余下部分将介绍这种新表示方式的各个方面，包括特征的单位和范围说明

，以及如何利用它生成特征空间。

1. 单位

SISSO++的一项重要升级是对表达式的单位进行广义且精确的处理。在物理学中，量纲分析是生成具有物理意义的表达式时的重要工具，因此在使用符号回归解决科学问题时，必须将其纳入考虑。我们将这一功能引入 SISSO++：根据初级特征确定每个新表达式的单位，并明确检查新表达式在物理上是否成立。在代码中，UNITS 以字典的形式实现，其中键表示基本单位，值表示对应键的指数，例如，$\mathrm{m/s^2}$ 会存储为$\{\mathrm{m}:1,\ \mathrm{s}:-2\}$。代码中还提供了将 UNITS转换为字符串以及从字符串转换回来的函数，以便更轻松地表示相关信息。随后，我们为这些专用字典实现了乘法、除法和幂函数，从而能够按照表 I 中的规则递归推导生成特征的单位。需要特别注意的是，当前实现无法转换同一物理量的不同单位，例如纳米、皮米和玻尔半径之间的转换。

借助这种单位实现方式，代码可以进行最低限度的量纲分析。量纲分析侧重于检查每个表达式内部以及最终模型的单位是否一致。对于最终模型，这项检查用于

表 I. 每种运算的单位计算方式

运算	结果单位
$A + B$	单位(A)
$A - B$	单位(A)
$A\,(B)$	单位(A) * 单位(B)
A/B	单位(A) / 单位(B)
$\lvert A - B\rvert$	单位(A)
$\lvert A\rvert$	单位(A)
正弦 (A)	无量纲
余弦 (A)	无量纲
指数函数 (A)	无量纲
指数函数 $(-A)$	无量纲
对数函数 (A)	无量纲
$(A)^{-1}$	单位(A)$^{-1}$
$(A)^2$	单位(A)^2
$(A)^3$	单位(A)^3
$(A)^6$	单位(A)^6
$\sqrt{A}$	单位(A)$^{1/2}$
$\sqrt[3]{A}$	单位(A)$^{1/3}$

表 II. 每个单位的限制；如果某项运算未列出，则表示没有限制。

运算	单位限制
$A + B$	单位(A) == 单位(B)
$A - B$	单位(A) == 单位(B)
$\lvert A - B\rvert$	单位(A) == 单位(B)
正弦 (A)	单位(A) $== \emptyset$
余弦 (A)	单位(A) $== \emptyset$
指数函数 (A)	单位(A) $== \emptyset$
指数函数 $(-A)$	单位(A) $== \emptyset$
对数函数 (A)	单位(A) $== \emptyset$

确定末尾线性模型中拟合常数的单位；这些常数可以具有任意单位，因此能够原生地完成任何单位换算。用于通过量纲分析拒绝表达式的限制列于表 II，可概括为：加法和减法只能作用于单位相同的特征，而所有超越运算都必须作用于无量纲量。施加这两项限制后，只能找到物理上可能的特征，单位的选择也不应再影响所选特征。如果希望找到非物理表达式，可以为所有输入主要特征指定不带关联单位。

2. 范围

特征构造步骤的另一项重要进展，是为初级特征引入范围，这些范围在后续特征构造运算中充当定义域。符号回归面临的一项挑战，尤其是在数据集较小的情况下，是所选表达式有时

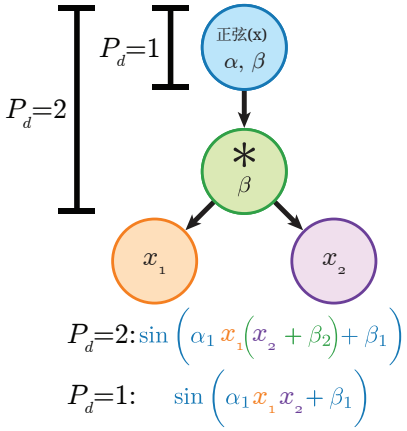

图 2. 参数化深度影响的示意图。如果 $P_d = 1$，则只有正弦算子被参数化；而如果 $P_d = 2$，则两个运算都被参数化。

会包含训练数据范围之外、但仍处于特定问题相关输入空间内的间断点。例如，这可能导致表达式对负数取对数函数，从而产生未定义的预测。SISSO++通过提供一种选项来解决这一问题：使用标准数学记法描述初级特征的范围，例如 $[0, \infty)$，然后依据表 III 中规定的规则，利用该范围计算所有使用此初级特征生成的公式的范围。在代码中，这些范围称为 Domain，因为层级为 $n-1$ 的特征的范围，是使用该特征构造层级为n 的表达式时可能采用的定义域。虽然表 III 中的所有范围都假定端点包含在内，但此实现也能处理不包含端点的范围，以及明确排除在范围之外的值列表，例如初级特征本身内部的点间断。

表 IV 列出了使用特征范围来防止生成新表达式的情况。在所有情况下，这都会阻止出现数学运算无定义的运算，例如对负数求平方根。如果未定义主要特征的取值范围，则不会执行这些检查，而是采用原先的假设，即所有运算都是安全的。

B. 参数化 SISSO

参数化 SISSO 扩展了 SISSO 的特征构造步骤，自动为每个运算加入缩放项和偏置项，正如 Purcell等人[33]所采用的那样。对于一般算子 $\hat{h}(\mathbf{x}) \in \hat{\mathcal{H}}$，给定一组缩放项和偏置项 $\hat{}$参数 $\mathcal{P}$，参数化方案将算子更新为，

$$\hat{h}\left(\mathbf{x}\right) \rightarrow \hat{h}^{\hat{\mathcal{P}}}\left(\alpha_1 \mathbf{x} + \beta_1\right), \tag{1}$$

表 III. 各运算范围的计算方法

运算	结果范围
$A + B$	[最小值（A）+最小值（B），最大值（A）+最大值（B）]
$A - B$	[最小值（A）−最大值（B），最大值（A）−最小值（B）]
A（B）	[最小值（最小值（A）∗最小值（B，最小值（A）∗最大值（B，最大值（A）∗最小值（B，最大值（A）∗最大值（B）），最大值（最小值（A）∗最小值（B，最小值（A）∗最大值（B，最大值（A）∗最小值（B，最大值（A）∗最大值（B））
A/B	$\left[\text{范围}（A）* \text{范围}\left(B^{-1}\right)\right]$
正弦（A）	$[-1, 1]$
余弦（A）	$[-1, 1]$
指数函数（A）	[指数函数 (min (A)) , 指数函数 (max (A))]
指数函数 ($-A$)	[指数函数 ($-$ max (A)) , 指数函数 ($-$ min (A))]
对数函数 (A)	[对数函数 (min (A)) , 对数函数 (max (A))]
$(A)^{-1}$	如果（$0 \in$ 范围（A））： $(-\infty,\ 0) \cup (0,\ \infty)$ 否则： $\left[（\text{最大值}（A））^{-1},\ （\text{最小值}（A））^{-1}\right]$
$(A)^3$	$\left[（\text{最小值}（A））^3,（\text{最大值}（A））^3\right]$
$\sqrt{A}$	$\left[\sqrt{\text{最小值}（A）}\ \sqrt{\text{最大值}（A）}\right]$
$\sqrt[3]{A}$	$\left[\sqrt[3]{\text{最小值}（A）}\ \sqrt[3]{\text{最大值}（A）}\right]$
$\lvert A\rvert$	[最大值（0，最小值（A））， 最大值（|最大值（A）|，|最小值（A）|）]
$(A)^2$	[最大值（0，最小值（A））^{2}, 最大值（|最大值（A）|，|最小值（A）|）^{2}]
$(A)^6$	[最大值（0，最小值（A））^{6}, 最大值（|最大值（A）|，|最小值（A）|）^{6}]
$\lvert A - B\rvert$	[最大值（0，最小值（A）−最大值（B））， 最大值（|最大值（最大值（A）−最小值（B））|，|最小值（最大值（A）−最小值（B））|）]

表 IV. 各项运算的定义域限制；未列出的运算均无限制。

运算	定义域限制
A/B	$0 \notin$ 范围 (B)
$(A)^{-1}$	$0 \notin$ 范围 (A)
对数函数（A）	最小（范围（A））> 0
$\sqrt{A}$	最小值（范围（A））≥ 0

其中，α_1 是缩放参数，β_1 是偏置项，$\mathbf{x}$ 是包含所有输入数据的向量。随后，这些新算子可用于构造新特征 $\hat{}\phi^{\hat{\mathcal{P}}}(\mathbf{x})$，与 SISSO 中的常规做法相同，每个特征都有自己的一组参数。不过，这确实为特征构造步骤引入了一个新的超参数——参数化深度 P_d，它定义了参数化在二叉表达式树中的层级。图 2 对此作了最清晰的说明，其中 P_d控制哪些运算会纳入参数化。在此示例中，如果 β_2 先前已通过另一项优化设定，且 $P_d = 2$，那么该先前设定的值将仅对这一特征失效；但是，如果 $P_d = 1$，则会保留现有的 β_2。为避免不同运算与线性回归中设定的常数之间存在线性依赖，部分缩放项和偏置项分别设为 1 和 0。所有算子的汇总见表 V。需要注意的是，对于对数函数算子，α 始终设为 ± 1，以避免与其他参数存在线性依赖。尽管这会留下一个单位依赖关系，但它可以

表 V. SISSO 更新后的特征生成步骤中使用的每种运算的自由参数和更新后方程式

运算	参数化运算
$A + B$	$A + \alpha B$
$A - B$	$A - \alpha B$
$A\,(B)$	$A(B + \beta)$
A/B	$A/(B + \beta)$
$\lvert A - B\rvert$	$\lvert A - (\alpha B + \beta)\rvert$
$\lvert A\rvert$	$\lvert A + \beta\rvert$
正弦 (A)	正弦 ($\alpha A + \beta$)
余弦 (A)	余弦 ($\alpha A + \beta$)
指数函数 (A)	指数函数 $(\alpha A)^{\mathrm{a}}$
指数函数 ($-A$)	指数函数 $(-\alpha A)^{\mathrm{a}}$
对数函数 (A)	对数函数 $(A + \beta)^{\mathrm{b}}$
$(A)^{-1}$	$(A + \beta)^{-1}$
$(A)^2$	$(A + \beta)^2$
$(A)^3$	$(A + \beta)^3$
$(A)^6$	$(A + \beta)^6$
$\sqrt{A}$	$\sqrt{\alpha A + \beta}^{\mathrm{b}}$
$\sqrt[3]{A}$	$\sqrt[3]{A + \beta}$

[a] $\alpha > 0$ [b] α 只能为 1.0 或 -1.0

可通过以下方式移除

$$\ln(x + \beta) \to \ln(\alpha_{\text{unit}}(x + \beta)) - \ln(\alpha_{\text{unit}}), \qquad (2)$$

其中，α_{unit} 是单位换算因子。

Once $\hat{\phi}$ 定义后，所有参数 $\hat{}p \in \hat{}\mathcal{P}$ 均使用非线性优化库 NLopt[34] 进行优化。

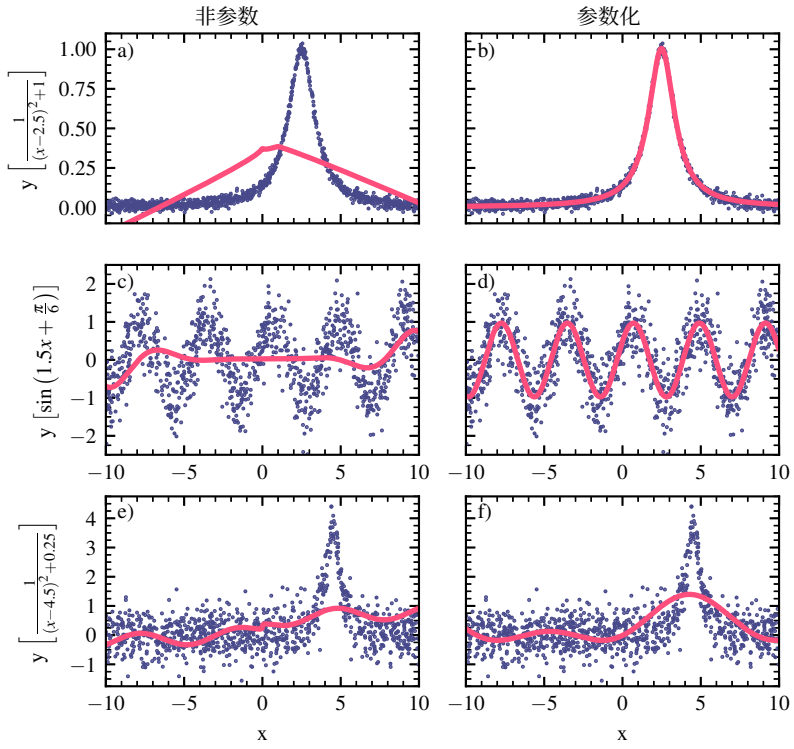

图 3. 对比通过非参数 SISSO（a、c 和 e）与参数化 SISSO（b、d 和 f）为洛伦兹函数（a、b、e 和 f）及正弦函数（c、d）找到的表达式。蓝点表示训练数据，红线表示 SISSO 找到的表达式。即使高噪声或参数的初始猜测不佳导致得到未优化解，参数化方案仍能找到正确的模型（b 和 d），或比非参数函数更好的模型（f）。

我们将柯西损失函数用作优化目标

$$\min_{\hat{\mathcal{P}}} f\left(\mathbf{P}, \hat{\phi}^{\hat{\mathcal{P}}}\right) \tag{3a}$$

$$f\left(\mathbf{P}, \hat{\phi}^{\hat{\mathcal{P}}}\right) = \sum_{i}^{n_{\text{samp}}} \frac{c^2}{n_{\text{samp}}} \log\left(1 + \left(\frac{P_i - \hat{\phi}^{\hat{\mathcal{P}}}\left(\mathbf{x}_i\right)}{c}\right)^2\right), \tag{3b}$$

其中，$\mathbf{P}$ 是性质向量，c 是缩放因子，所有计算中均设为 0.5，而 n_{samp} 是样本数。我们选择使用柯西损失函数而非均方误差，以提高非线性优化对数据集中离群值的鲁棒性。由于方程 3b 对尺度或偏置并非不变，因此引入额外的外部参数 α_{ext} 和 β_{ext}，分别用于考虑这些影响。对于多任务 SISSO[25],，每个任务都有自己的外部偏置和缩放参数，以考虑各自的线性回归解。以图 2（$P_d = 2$）所示的特征为例，待优化的函数为

$$\hat{\phi}^{\hat{\mathcal{P}}}\left(\mathbf{x}\right) = \alpha_{ext} \sin\left(\alpha_1 \frac{x_1}{x_2 + \beta_2} + \beta_1\right) + \beta_{ext}. \tag{4}$$

ˆ 为初始化 $\mathcal{P}$ 中的参数，我们将所有内部 α 和β 项分别设为 1.0 和 0.0，并将 α_{ext} 和β_{ext} 设为每个任务的最小二乘回归问题的解。在某些情况下，如果将 β 设为零会导致包含算子定义域之外的值，则可以将其设为非零值。在这些情况下，β 设为 min (x)+10^{-10}。每次优化都遵循此处概述的两步或三步流程。首先执行局部优化，以找到与初始参数对应的局部最小值。在找到局部最小值后，可选择执行全局优化，以查找优于最初找到的最小值的其他最小值。在前两个步骤中，参数分别以 10^{-3} 和 10^{-2} 的相对容差进行优化，函数评估次数最多为五千次。最后，执行精度更高的局部优化，将相对容差设为 10^{-6}，以找到最佳参数集。对于这次最终优化，允许进行一万次函数评估。此外，在初始优化和全局优化步骤中，为提高优化效率，参数的取值范围限制在 -100 到 100 之间，但在最终优化中会取消此限制。对于所有局部优化，均使用 Subplex 算法[35],，它是 Nelder-Mead 单纯形法的一种更快、更稳健的变体[36]，。The

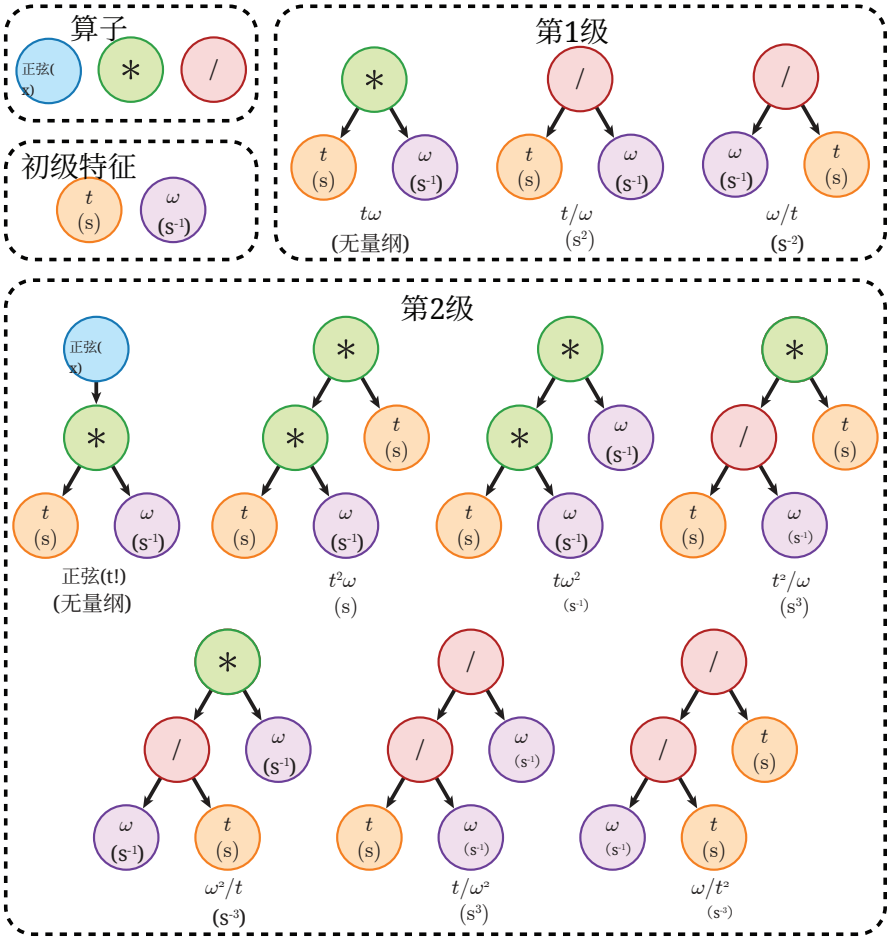

图4. SISSO 特征空间构建方式示意图。在此示例中，用户选择了两个初级特征 ω （紫色）和 t （橙色），以及三个算子：正弦（蓝色）、乘法（绿色）和除法（红色）。随后，SISSO 通过提高二叉表达式树的高度，对现有特征应用这些运算，逐步构建出更复杂的表达式空间。在此过程中，每种运算的单位和范围都得到遵循。

改进的随机排序进化策略算法[37] 用于所有全局优化。优化完成后，仅将内部 α 和 β 参数存储在$\hat{}$ $\mathcal{P}$中。

图3展示了新参数化方案的强大之处。对于由解析洛伦兹函数和带有一定白噪声的正弦函数表示的两个玩具问题，SISSO 的非参数版本无法找到准确的模型，因为它无法妥善处理非线性。采用这一新的参数化方案后，SISSO 现在能够准确地找到模型，如图3c和3d所示。不过，值得注意的是，更强大的特征构造能力需要付出代价：生成特征空间所需的时间显著增加，因为参数化过程成了计算瓶颈。此外，也可能出现参数化方案无法找到最优解的情况，例如噪声过大，或最优参数与初始猜测相距太远，如图3e和3f所示。

C. 构建完整特征集

随着特征表示方面的所有新改进到位，SISSO++ 实现了完全并行化的特征集构建，结合使用线程和 MPI进程等级，以高效构建特征集。创建新特征的基本过程如图 4 所示：每个新层级都在现有特征的基础上，通过在上一层级的二叉表达式树上添加新的运算来构建。在整个过程中，都会进行检查，以确保单位正确，并遵守每种新运算的定义域。此外，代码还会检查无效值，例如 NaN 或 inf，并对所有特征进行一些基本化简，例如会舍弃$\frac{t\omega}{\omega}$ 这类特征。算子分为彼此对应的参数化和非参数化版本，以便按需采用参数化SISSO概念。

III. 描述符识别

A. 分类问题的线性规划实现

SISSO 方法的一项重大更新，是提出了一种求解分类问题的全新通用方法。在以往的实现中，SISSO 在寻找分类方案时，会显式构建凸包，然后计算不同类别之间重叠区域内的点数，并计算归一化重叠体积或分离距离，以找出最优解。该方法适用于二维情形，但在三维及更高维度下，求解重叠体积或分离距离会变得难以处理；而对于四维及更高维度的分类问题，连定义凸包也会变得难以处理。SISSO++采用一种算法取代这些条件，通过线性规划确定凸包重叠区域内的点数，并使用线性支持向量机构建模型。线性规划算法会检查其可行性

$$\min 0 \tag{5}$$

$$\text{s.t.} \sum_{i\in I}^{\min} \alpha_i x_i = x_j, \quad \sum_{i\in I} \alpha_i = 1, \quad \alpha_i \geq 0,\ \forall i \in I, \tag{6}$$

其中， x_i 是类别 I所有点集中的 i^{th} 点， α_i 是 x_i的系数，而 x_j 是要检查其是否位于凸包内的点。当且仅当 x_j 位于表示问题中某个类别的点集 I内时，上述问题才有可行解。此处我们优化一个零函数，因为该优化问题的实际解并不重要，重要的是能找到这样的解，即满足约束条件。可行性问题和线性规划问题通过 Coin-CLP 库[38]定义。确定重叠区域中的点数后，我们会为最佳候选项计算线性支持向量机模型，并将其用作新的平局判定程序。首先比较 SVM 模型误分类的点数，其次比较间隔距离。SVM 模型使用 LIBSVM[39]计算。

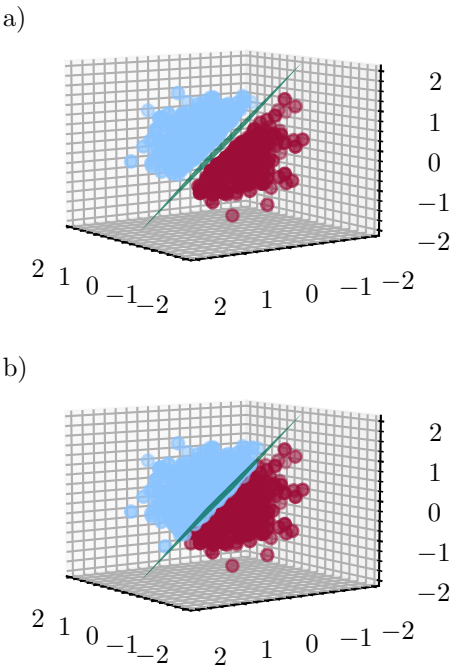

图 5. 分类算法示例。对三个特征 x_0、 x_1和 x_2，从以原点为中心、标准差为 0.5 的高斯分布中抽取 1000 个点。平面 $x_0 + x_1 + x_2 = 0$（绿色表面）将这些点分成两个类别（红色和浅蓝色圆点）。a) 将所有满足 $|x_1 + x_2 + x_3| < 0.4$ 的点移动到各自类别内距平面 $0.2\sqrt{2}$ 至 $0.35\sqrt{2}$ 的一个随机点；b) 原始数据存储为 x_3、 x_4和 x_5。更新后的分类算法正确判定 x_0、 x_1和 x_2 是更优的分类器；而按照原始定义，对于三维及更高维度中仅存在凸重叠区域的情况，它们会被视为同样出色。投影的仰角为 7.5 度，方位角为 145 度。

图 5 展示了新分类算法的能力。在这个问题中，我们从以原点为中心、标准差为 0.5 的高斯分布中随机抽样，为三个特征 x_0、 x_1和 x_2生成一千个样本。这三个特征被复制并重新标记为 x_3、 x_4和 x_5。随后，根据 $(x_1 + x_2 + x_3)$ 大于零（浅蓝色）还是小于零（红色），将样本分成两个类别。对于 x_0、 x_1和 x_2，我们将所有点
其中 $|x_1 + x_2 + x_3| < 0.4$ 移动到各自类别内远离分割平面的一个随机点，以确保形成明显的分隔间隔。为简便起见，我们仅使用图 5a 和 b 坐标轴上所示的六个初级特征，不生成任何表达式；

然而，特征 $(x_0 + x_1) + x_2$ 的第2阶特征能够在一维空间中将类别完全分开。使用更新后的算法，SISSO 现在可以轻松识别出集合 $\{x_0, x_1, x_2\}$ 比集合 $\{x_3, x_4, x_5\}$ 更适合作为分类器，因为平局判定规则适用于所有维度，而不仅仅是前两个维度。更重要的是，SISSO 现在可以给出由线性支持向量机找到的 n 维分割平面，从而自动构建出一个实际的分类器，此处以绿色平面表示。需要注意的是，同样的过程可以应用于任意维度，但在三维以上时，可视化会变得不切实际。

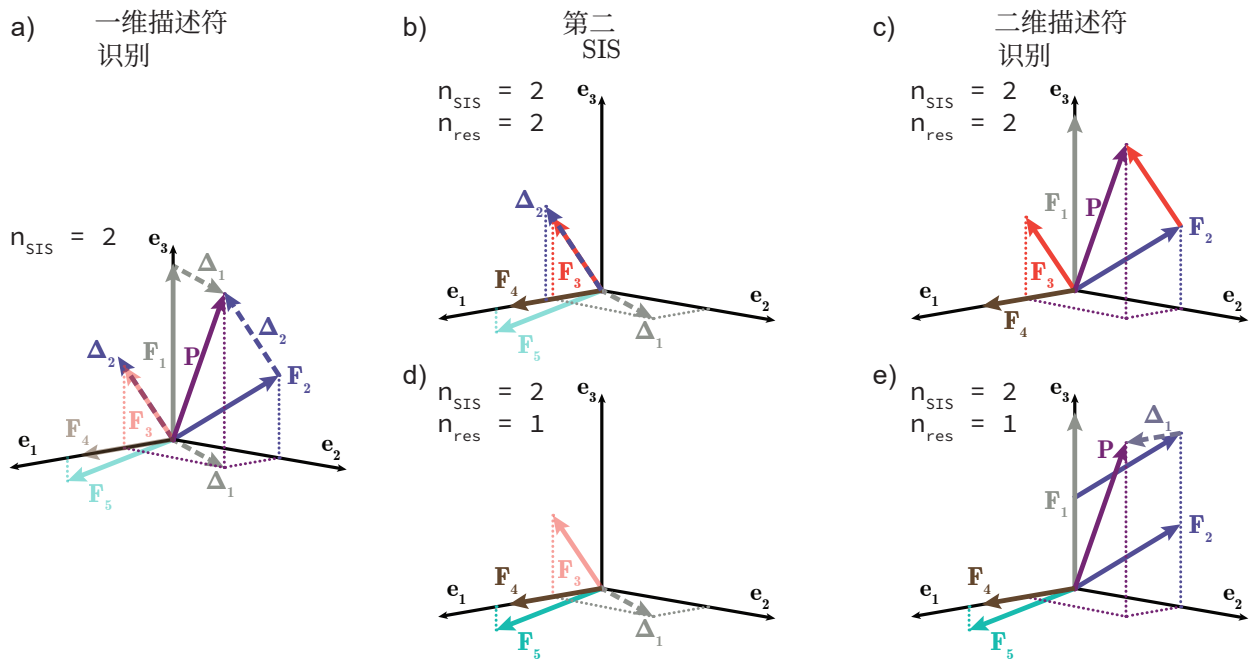

图6. 示意图说明跟踪多个残差如何提升 SISSO 的性能。由 $\mathbf{e_1}, \mathbf{e_2}$，和 $\mathbf{e_3}$ 定义的三维问题空间（包含三个训练样本），具有五个特征向量 $\mathbf{F_1}$（灰色）、$\mathbf{F_2}$（蓝色）、$\mathbf{F_3}$（红色）、$\mathbf{F_4}$（棕色）和 $\mathbf{F_5}$（青绿色），以及一个性质向量 $\mathbf{P}$（紫色）。在此示例中，SIS 子空间的大小 n_{sis} 为 2。图中以虚线表示最佳（$\mathbf{\Delta_1}$，灰色）和次佳（$\mathbf{\Delta_2}$，蓝色）一维模型的残差。若残差数 n_{res} 为 1，则只使用 $\mathbf{\Delta_1}$（d、e），而 $\mathbf{F_3}$ 不会在第二个 SIS 步骤中被选中。这意味着无法找到最佳二维模型。然而，若 $n_{res} = 2$（b、c），则会选中 $\mathbf{F_3}$，并且在第二个 ℓ_0 步骤中可以找到由 $\mathbf{F_2}$ 和 $\mathbf{F_3}$，组合而成的最佳二维模型。

B. 多个残差

SISSO 算法描述符识别步骤的第二项改进，是引入多残差方法，为维度高于一的模型选择表达式（特征）。在原始SISSO算法[40], 中，先前找到的模型 $\vec{\Delta}^0_{D-1}$ 的残差，即存储每个样本属性值的向量 P 与（$D-1$）维模型（$\vec{\Delta}^0_{D-1} = \vec{P}_{D-1} - \vec{P}$）预测值之间的差值，用于在 SIS 步骤中计算候选特征的投影得分，以确定最佳 D维模型 $s^0_j = R^2\left(\vec{\Delta}^0_{D-1}, \vec{d}_j\right)$。这里，$R$ 是皮尔逊相关系数，对应一个回归问题；j 则对应于 SISSO 特征创建步骤中生成的每个表达式。在 SISSO++[27]中，我们扩展了残差的定义，并使用最佳的 r 个残差来计算投影得分：$\max\left(s^0_j, s^1_j, \ldots, s^{r-1}_j\right)$。多残差概念扩展了 SISSO 的描述符识别步骤，利用模型集成的信息来确定要添加到选定子空间中的特征。一次计算的r 值可通过交叉验证设定为任意超参数。

这一过程在图6中针对一个三元.....展示。

由$\mathbf{e_1}$，$\mathbf{e_2}$，和$\mathbf{e_3}$ 定义的维度问题空间（三个训练样本）中，使用五个候选特征$\mathbf{F_1}, \cdots, \mathbf{F_5}$来描述性质$\mathbf{P}$。原则上，特征向量的数量可以任意多，但为清晰起见，我们仅展示五个。如果仅使用单个残差，那么选出的二维模型将由$\mathbf{F_1}$ 和$\mathbf{F_4}$，组成，因为$\mathbf{F_3}$ 从使用 F_1找到的模型的残差中获得的投影得分最小，因此从未被选中。然而，由于 $\mathbf{F_2}$ 沿$\mathbf{e_2}$，具有一个分量，因此存在一个由$\mathbf{F_2}$ 和$\mathbf{F_3}$ 的线性组合构成的更优二维模型，尽管$\mathbf{F_2}$ 并不是与性质$\mathbf{P}$相关性最高的特征。当进入更高维的特征空间时，相关性相似的特征向量包含正交信息的可能性会增加，因此在 SISSO 中需要使用多个残差。

为了展示在多个残差上进行学习的效果，并估计最优残差数量和 SIS 子空间大小（n_{sis}），我们绘制了函数 $y = 5.5 + 0.4158d_1 - 0.0974d_2 + \delta$对应二维模型的训练均方根误差，其中$d_1 = x_0^2\sqrt[3]{x_1}$、$d_2 = \left|x_2^3\right|$和 δ 是从标准差为 0.05 的分布中抽取的高斯白噪声项，如图 7 所示。对于这一问题，最佳一维描述符是 $\frac{1}{\sqrt[3]{x_3}}$，其中 x_3 被显式设定为（$y + \Delta$）$^{-3}$，而 Δ 是标准差为 20.0 的高斯白噪声。该主要特征为

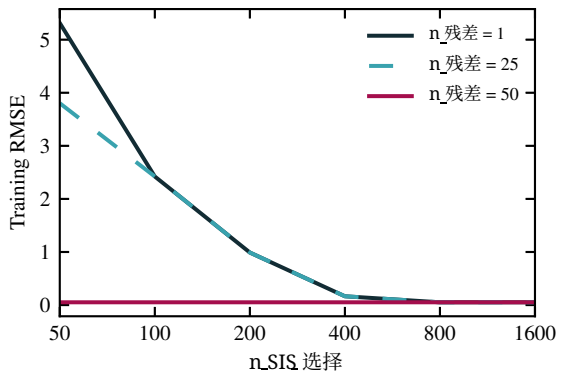

图 7。二维模型的训练均方根误差与 SIS 子空间大小的关系（ $y = 5.5 +$ ），$0.4158d_1 - 0.0974d_2 + \delta$ $d_1 = x_0^2 \sqrt[3]{x_1 \quad} d_2 = |x_2^3|$ 其中 、 和δ 是从标准差为 0.05 的分布中抽取的高斯白噪声项。深蓝色实线表示使用一个残差进行学习时的误差，浅蓝色虚线表示使用二十五个残差进行学习时的误差，红色实线表示使用五十个残差进行学习时的误差。

明确添加这一项，是为了确保在这个合成问题中，最佳一维模型既不是 d_1 ，也不是 d_2 。因此，使用单个残差时，必须将 SIS 子空间的大小增加到 400 以上，SISSO 才能复现 y 。然而，将残差数量增加到 50 后，SISSO 现在能够找到与 d_1 的残差相关性最高的特征，并立即找到 y。在我们最近发表的一篇论文中，我们进一步展示了这种方法在学习立方钙钛矿体积模量模型方面的有效性，[28]。

IV. 结论

本文从数学和计算细节两方面，介绍了近期对 SISSO 方法所作的改进及其在 SISSO++code 中的实现，这些改进大幅提升了 SISSO 方法的表达能力。利用这些特性，可以更灵活地控制 SISSO 找到的表达式，也为在 SISSO 和符号回归中引入“语法”规则开了个头。特别是，公式的单位和范围等概念可以用于缩减最终模型可能表达式的搜索空间。我们还介绍了参数化 *SISSO*的实现，它大幅拓宽了 SISSO 能找到的表达式范围。最后，我们讨论了与 SISSO 求解器相关的两项改进，即用于分类问题的线性规划实现和多残差技术；两者都提升了 SISSO 所找到的描述符和模型的灵活性。

V. 致谢

T.A.R.P. 感谢 Christian Carbogno 就参数化 SISSO 方案进行的宝贵讨论，并感谢他校对了手稿中的相关部分。T.P. 感谢 Lucas Foppa 就多残差方法进行的讨论，并感谢他校对了手稿中的相关部分。本研究由 NOMAD 卓越中心（欧盟地平线 2020 研究与创新计划，资助协议编号 951786）、ERC 高级资助 TEC1p（欧洲研究委员会，资助协议编号 740233）、BigMax（马克斯·普朗克学会大数据驱动材料科学研究网络）以及 FAIR-mat 项目（面向凝聚态物理和固体化学物理的 FAIR 数据基础设施，德国研究基金会，项目编号 460197019）资助。T.P. 感谢亚历山大·冯·洪堡基金会通过洪堡博士后研究奖学金计划给予的支持。

VI. 利益冲突声明

作者声明不存在需要披露的利益冲突。

VII. 作者贡献

T.A.R.P. 实现了所有方法并完成了所有计算。T.A.R.P. 在 LMG 的协助下构思了所有方法。MS 和 LMG 负责监督该项目。所有作者均撰写了稿件。

VIII. 数据可用性声明

支持本研究发现的数据以及用于生成图表的所有脚本均已在 FigShare 上公开，数字对象标识符为：10.6084/m9.figshare.22691857。

[1]朱等，Energy Environ. Sci. **14**, 3559 (2021). [2]S. A. Miller 等，Chem. Mater. **29**, 2494 (2017). [3]K. Tran 和 Z. W. Ulissi，Nature Catalysis**1**, 696 (2018). [4]L. Foppa 等，《MRS 公报》 **46**, 1016 (2021). [5] L. Foppa 等，《ACS 催化》 **12**, 2223 (2022). [6]C. J. Bartel 等，《科学进展》 **5** (2019). [7]V. Stanev、K. Choudhary、A. G. Kusne、J. Paglione 和 I. Takeuchi，Commun. Mater. **2**, 105 (2021). [8]P. P. Angelov 、 E. A. Soares、R. Jiang、N. I. Arnold 和 P. M. Atkinson，《WIREs 数据挖掘与知识发现》 **11**, e1424 (2021). [9]D. Gunning 等，《科学·机器人学》 **4**, eaay7120 (2019). [10]A. Das 和 P. Rad，arXiv 预印本 arXiv:2006.11371 (2020). [11]F. Xu 等，《可解释人工智能：历史、研究领域、方法与挑战简述》，载于 *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 11839 563 574, 2019 ，

LNAI 卷，页码 – Springer，。[12]G. S. I. Aldeia 和 F. O. De Fran.ca，《利用部分效应衡量符号回归模型的特征重要性》，载于 *GECCO 2021 - Proc. 2021 Genet. Evol. Comput. Conf.*，页码 750–758，2021。

[13]A. Holzinger、A. Saranti、C. Molnar、P. Biecek 和 W. Samek，《可解释人工智能方法——简要概述》，载于 *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 13200 13 38，LNAI 卷，页码 – 施普林格科学与商业媒体德国有限公司，2022。[14]Z. Li、J. Ji 和 Y. Zhang，arXiv: 2111.12210 (2021). [15]Y. Wang、N. Wagner 和 J. M. Rondinelli，MRS Communica- tions **9**, 793 (2019). [16]J. R. Koza, Statistics and Computing **4**, 87 (1994). [17]T. Mueller、E. Johlin 和 J. C. Grossman，Physical Review B **89**, 115202 (2014). [18]F. Yuan 和 T. Mueller，Scientific Reports **7**, 17594 (2017). [19]S.-M. Udrescu 和 M. Tegmark，arXiv 预印本：1905.11481 (2019). [20]S. Kim 等，IEEE Transactions on Neural Networks and Learn- ing Systems **32**, 4166 (2021). [21]M. D. Cranmer、R. Xu、P. Battaglia 和 S. Ho，利用图网络学习符号物理，2019。[22]M. Valipour、B. You、M. Panju 和 A. Ghodsi, SymbolicGPT：用于符号回归的生成式Transformer模型，2021。[23]B. K. Petersen 等，深度符号回归：通过风险寻求策略梯度从数据中恢复数学表达式，2021。[24]W. Tenachi、R. Ibata 和 F. I. Diakogiannis，受单位约束引导的物理学深度符号回归：迈向物理定律的自动发现，2023。[25] R. Ouyang、E. Ahmetcik、C. Carbogno、M. Scheffler 和 L. M. Ghiringhelli，J. Phys. Mater. **2**, 24002 (2019). [26]R. Ouyang、S. Curtarolo、E. Ahmetcik、M. Scheffler 和 L. M. Ghiringhelli，Phys. Rev. Mater. **2**, 83802 (2018). [27]T. A. R. Purcell、M. Scheffler、C. Carbogno 和 L. M. Ghir- inghelli, J. Open Source Softw. **7**，3960（2022）。[28]L. Foppa、T. A. Purcell、S. V. Levchenko、M. Scheffler 和 L. M. Ghringhelli，Phys. Rev. Lett. **129**，55301（2022）。[29]G. R. Schleder、C. M. Acosta 和 A. Fazzio, ACS Appl. Mater. Interfaces **12**，20149（2020）。[30]Z.-K. Han 等, Nat. Commun. **12**，1833（2021）。[31]G. Pilania、C. N. Iverson、T. Lookman 和 B. L. Marrone，J. Chem. Inf. Model. **59**，5013（2019）。[32]J. Fan 和 J. Lv，J. R. Stat. Soc. Ser. B Stat. Methodol. **70**，849（2008）。[33]T. A. R. Purcell、M. Scheffler、L. M. Ghiringhelli 和 C. Car- bogno, 已提交至 npj Comput. Mater.（2022）。[34]S. G. Johnson，NLopt 非线性优化软件包，http://github.com/stevengj/nlopt，2021。[35]T. H. Rowan, *Functional stability analysis of numerical algorithms* 1990，博士学位论文，德克萨斯大学奥斯汀分校，。[36]J. A. Nelder 和 R. Mead, Comput. J. **7**，308（1965）。[37]T. Runarsson 和 X. Yao，IEEE Trans. Syst. Man Cybern. Part C (Applications Rev. **35**，233（2005）。[38]J. J. Forrest 等，coin-or/clp：版本 1.17.6。[39]C.-C. Chang 和 C.-J. Lin, ACM Transactions on Intelligent Systems and Technology **2**，27:1（2011），软件可通过 `http://www.csie.ntu.edu.tw/~cjlin/libsvm` 获取。[40]R. Ouyang、S. Curtarolo、E. Ahmetcik、M. Scheffler 和 L. M. Ghiringhelli，Physical Review Materials **2**，083802（2018）。