arXiv:2304.01869v1 [cs.NE] 4 Apr 2023

DEEP-BIAS：使用可解释人工智能检测结构偏差

Bas van Stein
LIACS，莱顿大学 莱顿，荷兰
b.van.stein@liacs.leidenuniv.nl

Diederick Vermetten
LIACS，莱顿大学 莱顿，荷兰
d.l.vermetten@liacs.leidenuniv.nl

Fabio Caraffini
计算机科学系 斯旺西大学 斯旺西，英国
fabio.caraffini@swansea.ac.uk

Anna V. KononovaLIACS，莱顿大学 莱顿，荷兰
a.kononova@liacs.leidenuniv.nl

摘要

评估启发式优化算法的性能对于确定其在各种条件下的表现至关重要。近期，BIAS工具箱被引入作为一种行为基准测试，用于检测搜索算法中的结构偏差（SB）。该工具箱可用于识别现有算法中的偏差，也可用于检验新开发算法是否存在偏差。本文介绍了BIAS工具箱一种新颖且可解释的深度学习扩展，称为 Deep-BIAS。原始工具箱使用39项统计检验和一个随机森林模型来预测SB的存在及其类型，而 Deep-BIAS 方法则使用经过训练的深度学习模型，根据原始性能分布直接检测SB的强度和类型。通过针对各种具有结构偏差的场景开展一系列实验，我们展示了 Deep-BIAS 的有效性。我们还介绍了将该工具箱应用于336种最先进的优化算法所得的结果，结果显示这些算法存在多种类型的结构偏差，尤其表现为偏向目标空间中心或呈现离散化行为。Deep-BIAS 方法在偏差检测和SB类型分类方面均优于BIAS工具箱。此外，还可以利用可解释人工智能技术生成解释。

关键词 结构偏差，算法行为，可解释人工智能，优化

1 引言

随着数据量和优化问题复杂度不断增加，对有效启发式优化算法的需求也随之增长。由于没有任何一种启发式算法能够在所有情况下都表现最佳 [1], ，因此有必要对这些算法进行基准测试，以了解它们在特定条件下的表现。大多数连续优化基准测试都基于性能，例如黑盒优化基准测试（BBOB） [2]测试套件。这些基准测试提供算法性能及其在各种情况下与其他算法相比表现的信息。例如，一种算法可能擅长优化可分函数，而另一种算法则在单峰、高条件数函数上表现良好。基于资源的基准测试衡量特定条件下所需的资源（计算能力、内存和能量），但无法深入揭示算法在不同情况下的行为。另一方面，基于行为的基准测试则能提供更多信息，说明算法在不同条件下如何表现，例如群体优化算法中候选解种群的移动情况。

此前提出的 BIAS工具箱 [3] 是一种基于行为的基准测试工具。它可用于分析算法或算法组件是否会引发结构偏差(SB)。结构偏差是一种固有于迭代过程中的偏差。

影响其在目标空间中性能的启发式优化算法。通过检测启发式优化算法中结构偏差的存在、强度和类型，可以确定需要改进的方面，并了解在何种条件下结构偏差较不容易出现。这些信息可用于优化此类算法的性能。

在本研究中，我们针对 BIAS工具箱提出了一种改进的方法，用于检测结构偏差并分类 SB 的类型。我们不再使用 39 项统计检验及其 p值，而是提出使用单个卷积深度学习模型，根据在特殊测试函数 f_0上多次优化运行所得到的原始最终点分布，预测是否存在 SB 及其类型。在本研究中，我们旨在回答以下研究问题：

RQ1 如何利用机器学习确定某个启发式连续优化算法是否存在 SB？RQ2 如何预测某个连续启发式优化算法所遭受的 SB 类型？RQ3 我们能否生成对预测 SB 类型的可视化解释？RQ4 所提出的基于深度学习的方法是否优于统计检验方法，其优势在什么条件下体现?

为回答 RQ1 和 RQ2，我们提出了一种基于原始算法性能数据（在特殊测试函数上获得）的深度学习方法，用于识别给定优化器中最可能出现的 SB 类型（如果存在）。完整的 SB 基准测试深度学习测试套件（Deep-BIAS）以及针对不同 SB 情景的数据生成器均已开源 [4]。为回答 RQ3，我们提出采用针对所提出深度学习方法调整后的可解释人工智能技术，将 SB 结果可视化。为回答 RQ4，我们使用包含 11 种不同结构偏差场景的大量 189 个不同人工生成的参数化分布，对所提出的深度学习方法和先前的统计检验方法进行了评估。此外，我们还在广泛的最先进优化器集合上比较了两种方法的结果。

2 结构偏差

在许多复杂优化问题中，我们必须面对完全的黑箱设置。这意味着我们没有关于空间中哪些区域包含优质解的先验信息。在这种设置下，搜索必须在定义域边界内“从头开始”。随后，搜索应能够识别有希望的区域，并朝着目标函数值较优的区域推进。到目前为止采样得到的点应当是后续搜索步骤中影响算法逻辑和算子的唯一依据。这意味着算法不会天生偏向空间中的某个区域。在其他条件下，若要将算法应用于一般情形，无论高质量解位于问题可行域内的何处，算法都应能够找到它们。算法在寻找最优解时表现出的这种灵活性，是影响其成功与否的重要因素之一。

遗憾的是，由于函数的采样景观与算法的内部工作机制之间存在相互作用，因此很难检测迭代算法对定义域某些部分的偏向性 [5]。为了将这两个组成部分区分开来，测试函数 f_0 被定义如下：

$$f_0 : [0,1]^n \to [0,1], \text{ where } \forall x, f_0(x) \sim \mathcal{U}(0,1). \tag{1}$$

对于此函数，最优值在定义域内均匀随机地分布，因此不存在任何一个区域天生优于其他区域的情况。因此，预计无偏搜索会返回最终最优解的均匀分布。如果这些点的分布是非均匀的，则表明算法存在结构偏差，因为算法无法对定义域中的每个区域给予同等重要性。因此，结构偏差体现了算法的不灵活性。基于此，我们认为结构偏差是一种不理想的行为，可能会限制算法在通用场景下的性能。算法的算子与其结构偏差之间的确切关系，以及结构偏差可能以何种方式影响算法在不同函数景观上的性能，目前尚未得到充分理解。然而，通过在 f_0上的测试，可以识别出结构偏差，从而有可能通过谨慎选择算子将其消除。

2.1 测量结构偏差的现有方法

对在 f_0上找到的最终解分布进行视觉检查：这些结果是在多次独立运行待研究方法后收集的，并按分量显示 [6], ，这是检测结构偏差最直观的方法。虽然这种方法有助于理解结构偏差在明显有偏和无偏情况下的表现，但它容易受到个人解读的影响；当需要生成并检查大量图像时，也会耗费大量时间（参见 [5,7,8] 和代码仓库 [9,10]）。此外，面对轻微的结构偏差或图形渲染伪影时，这种方法无法给出可靠结果，因此难以得出正确结论。

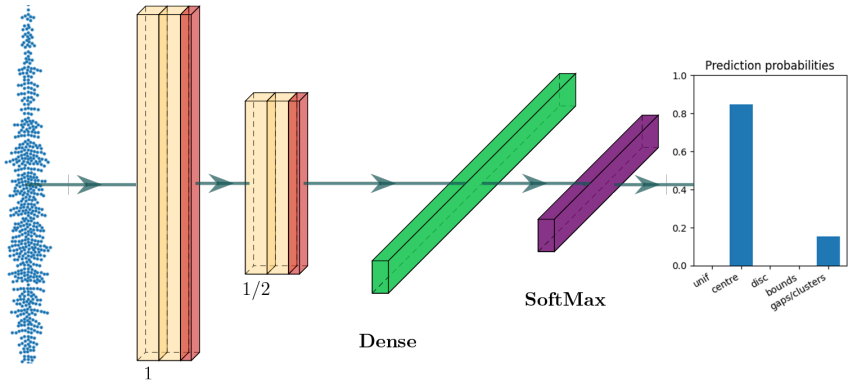


图1：具有针对每种样本量优化的超参数的通用一维CNN架构。网络以固定样本量的排序分布作为输入。黄色层为一维CNN层，红色层为最大池化层，绿色层为全连接层，最后是一个分类头，其SoftMax激活函数会为每个样本生成五个类别概率。

使用统计检验方法可以消除 SB 检查中的主观因素，并针对大量结果数据集实现自动化决策过程：对在 f_0 中通过多次运行获得的最终解的均匀性分布进行均匀性检验。然而，均匀性检验颇具挑战，尤其是在考察“实际”样本量 N时；在本例中，该样本量指需要执行的完整运行次数。实验表明，在部分算法类别中，当 N 不超过 100，并结合 Kolmogorov-Smirnov检验 $\alpha = 0.01$ 、Cramer-Von Mises检验 [11], 和 Anderson-Darling [13]检验以及 Benjamini-Hochberg校正方法 [14], 时，效果良好；但单独使用这些检验，在研究种类繁多的算法时表现不佳 [5, 15]。使用 $N = 600$可获得最佳结果，它有助于更频繁地检测到结构偏差，但无法保证在任何显著性水平下都能检测出所有不同类型的结构偏差 [8]——显著性水平越低、统计功效越高，就越需要增大 N [16]。要通过多种统计检验可靠地检测结构偏差及其类型，过程可能复杂而费力。从实践者或算法设计者的角度来看，这些流程应当实现自动化。

2.2 BIAS工具箱

BIAS [17] 是一个开源Python软件包，可从 [18], 获取，用于对连续域中的SB进行基准测试。该工具箱提供了一种SB检测机制，该机制基于汇总39项统计检验的结果，并利用随机森林模型识别结构偏差的类型。此外，它还包含一个数据生成器，可从一组场景中抽样生成合成结果；一个用于生成最终最优位置平行坐标图的组件，以展示SB；以及一个用于呈现通过统计分析检测SB时所作决策结果的组件。
BIAS工具箱易于使用、可扩展，能让SB检测流程变得快速而简单，即使面对大量最终最优位置也不例外。它还有助于对用于检测结构偏差的其他统计检验进行基准测试。

3 方法

这里提出的方法扩展了 BIAS 的功能。它使用基于深度学习的模型来预测结构偏差是否存在及其类型。该模型在大量偏差场景组成的场景组合上进行训练，并使用 AutoKeras 进行优化。本节首先概述这些场景以及用于训练模型的生成器，然后详细说明模型训练和测试流程。

3.1 场景组合

所提出的 Deep-BIAS 方法基于 [3] 中提出的参数化 SB 情景。这个场景组合包含了以往研究中观察到的最常见的结构偏差类型，包括：

- 偏向搜索空间中心，
- 偏向搜索空间边界，
- 倾向于搜索空间的某些区域，形成簇，

- 倾向于避开搜索空间的某些区域，从而形成空隙并
- 强离散化。

原论文中有 11 种不同的场景，每种场景采用不同的参数设置，共形成 194 个参数化场景。对这些场景的参数进行视觉和分析后，我们删除了其中 5 种参数设置，因为仅使用 600 个样本时，它们经常会生成在视觉和统计上都与随机均匀分布等效的分布。这样便剩下 189 个参数化分布生成器（仍对应 11 种场景），我们使用它们生成训练集和测试集。
为了训练深度学习模型，我们将 11 种场景分为四类：中心、边界、间隙/簇 和离散化。我们认为，间隙与簇偏差高度重叠，因为搜索空间中有簇时就会出现间隙，反之亦然。因此，我们将它们合并为一个类别标签。每个类别生成 20.000 个分布，并在属于该类别标签的参数设置和场景之间均匀分配。对于均匀（无偏差）标签，我们生成了 80.000 个分布，使偏差样本与无偏差样本之间的预测任务保持平衡。总计得到一个包含 160.000 个一维分布的数据集，其中 80% 用作训练数据，20% 用作最终验证数据（采用分层抽样）。

3.2 优化的卷积神经网络

在下一步中，使用 AutoKeras [19]算法训练并优化深度一维卷积神经网络（1d CNN）。首先将每个分布中的值按升序排列，然后将其作为 1d CNN 的输入。AutoKeras 设置为运行 100 次试验的评估预算，并且不使用图像增强和预处理（由于我们处理的不是图像，这在本研究情境下没有意义）。每个网络训练 50 个训练轮次，并使用一个随机选取的小型验证集（来自原始训练数据）来比较不同模型。性能最佳的神经网络实例将被保存并用于本研究。模型的总体架构如图 1 所示。其中包含两个模块，每个模块由两个卷积层和一个最大池化层组成。第二个模块的大小是第一个模块的一半。随后连接一个全连接层和一个 SoftMax 分类头。总共有四个模型，分别对应 30 、 50 、 100 和 600 的样本量。在 NVIDIA T4 GPU 上，每种样本量对应的网络训练和优化大约需要一天。

3.3 预测 SBand SBtype

现在可以使用优化后的 CNN 模型，直接预测任意分布是否存在结构偏差及其类型。首先，我们使用留出的最终验证集来检验这些模型。

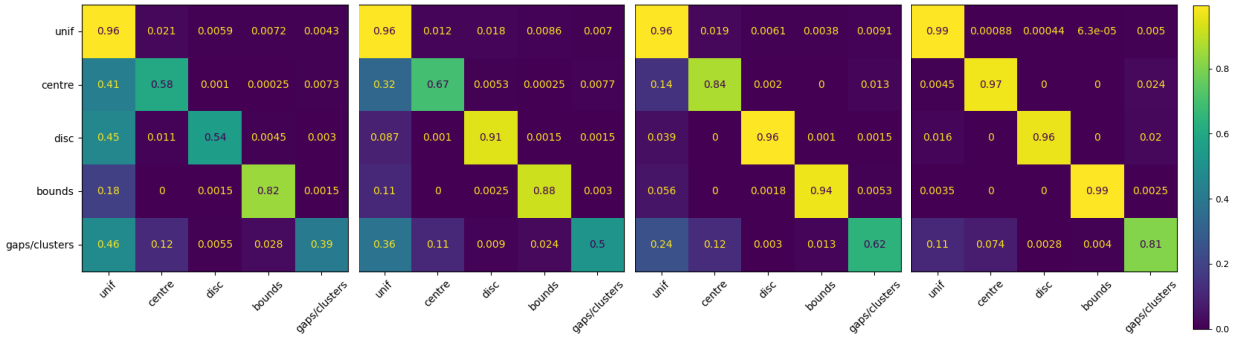


图2：不同样本量下模型的混淆矩阵，从左到右分别为：30、50、100 和 600。横轴表示预测标签，纵轴表示真实标签。模型使用包含 128.000 个一维分布的数据集进行训练，并在包含 32.000 个分布的数据集上进行验证。样本量为 600 的模型宏平均 F1 分数为 0.95。

图2展示了不同样本量下的分类结果。可以看出，随着样本量增加，分类结果明显改善；对于 [3]中提出的统计方法，情况也同样如此。值得注意的是，即使是使用 50 个样本的网络，也能很好地将均匀分布预测为均匀分布。大多数误分类都发生在模型将某个分布预测为均匀分布、但实际并非如此的情况下。

为进一步分析各模型的误分类情况，我们采用可解释人工智能（XAI）方法 SHAP [20], , 该方法基于训练数据中的背景样本集 [21] 近似计算 Shapley值。利用分布中每个点的 Shapley值，可以突出显示特定预测的感兴趣区域。

在图3中，展示了四个与真实标签不符的预测示例。这些示例来自样本量为600的模型。值得注意的是，许多场景生成器具有很强的随机性。因此，簇可能高度重叠，形成均匀分布的样本；簇也可能位于中心或边界处，从而误导分类器。在所示示例中还可以看到，一些生成器能够生成细微的偏差（均匀分布中的微小间隙），导致其难以被正确分类，因为对于600个样本而言，这些偏差也可能偶然出现在随机均匀分布中。

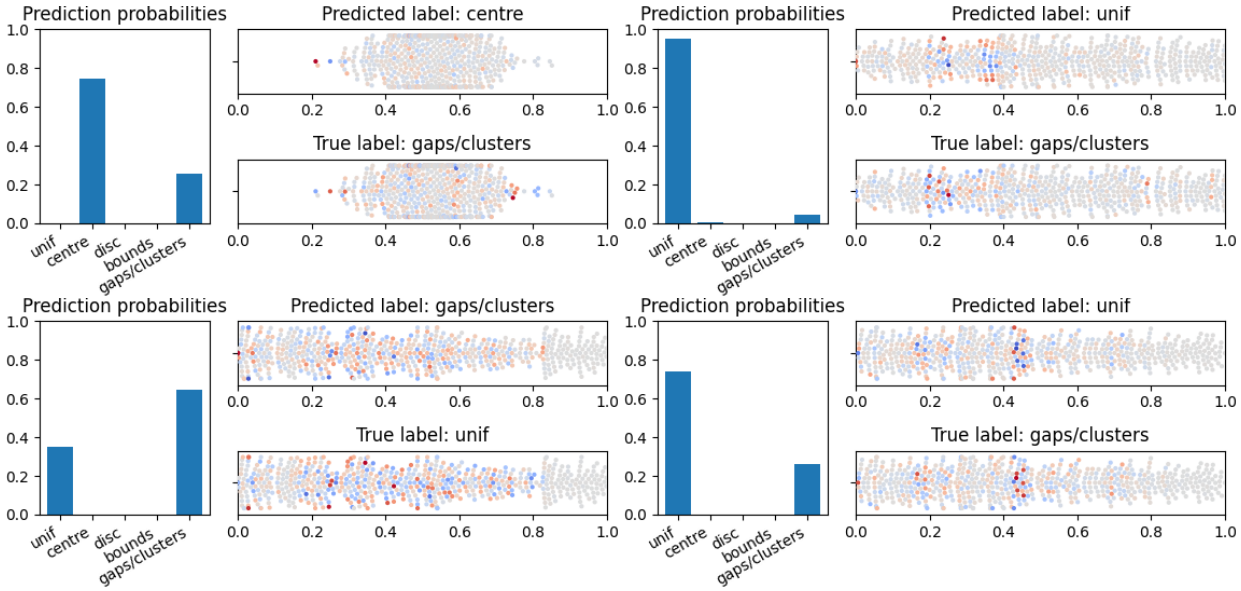


图3：由于类别重叠而被（错误）分类的样本示例。簇可能偶然位于空间的边界或中心；在均匀采样中，也可能偶然出现小间隙。颜色表示 Shapley值，其中深红色表示对类别标签（预测标签（上方）或真实标签（下方））的正向贡献，深蓝色表示负向贡献。数值相近的样本点会彼此堆叠。

接下来，将 Deep-BIAS 方法与原始 BIAS工具箱进行比较。通过将问题转换为二元问题来检测结构偏差（SB）。由于原始 BIAS工具箱也以这种方式在不同维度数（而不只是针对一维分布）下进行了验证，因此这里重复进行相同的实验。评估的维度数为：1、10、20 和 30。对于这些维度数（d），生成一个测试集，其中包含 1890 个有偏和 1890 个无偏的 d 维分布。随后，使用原始 BIAS工具箱和新提出的 Deep-BIAS 方法对这些分布进行预测。对于所提出的方法，对所有维度的类别概率取平均值，从而为每个 d 维分布给出最终预测。我们将配置定义为有偏配置，当且仅当其至少 10% 的维度被分类为非均匀。选择这一阈值是为了与原始 BIAS工具箱保持一致 [3]。

本实验的结果如图4所示，其中比较了不同维度和样本量下的假阳性、假阴性和F1分数。可以看出，虽然 Deep-BIAS 的假阳性率略高，但其假阴性率显著较低，弥补了这一不足。总体而言，Deep-BIAS 的 F1 分数略高，表明其性能略优于统计方法。

4 真实算法数据的结构偏差基准测试

本节通过应用 BIAS 工具箱，对大量启发式优化算法进行基准测试，以回答 RQ3。

4.1 数据收集设置

我们使用来自异构启发式算法集合的数据，这些算法在维度为 $n = 30$ 时运行于 f_0，最多进行 $10000 \cdot n$ 次适应度函数调用。我们总共考虑 336 种优化启发式算法，它们分为以下几类（均使用 $N = 100$）：

- 差分进化的变体（195 种配置），

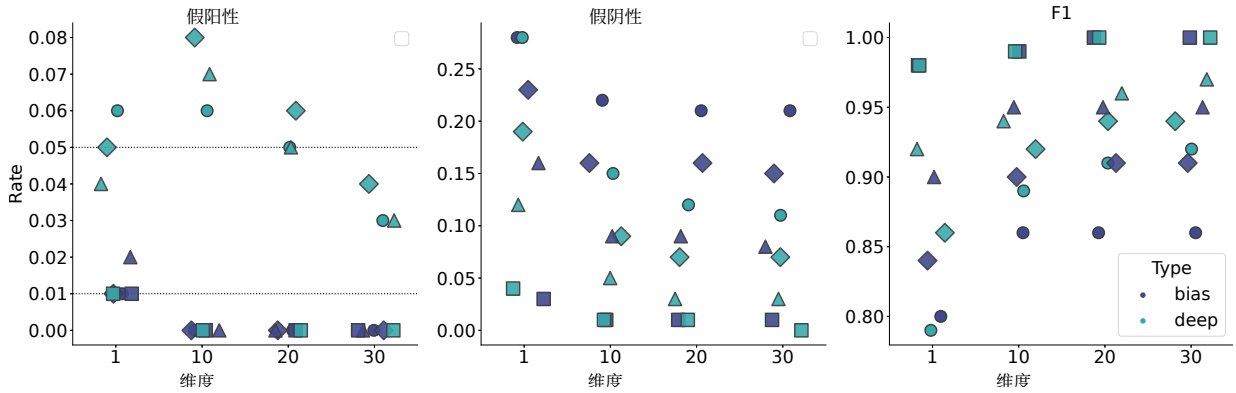


图4：原始 BIAS 工具箱（蓝色）与 Deep-BIAS（青绿色）在假阳性（左）、假阴性（中）和 F1 分数（右）方面的比较（含 $\alpha = 0.01$）。所有图中，标记表示所使用的样本量：○、◇、△ 和 □分别为 30、50、100 和 600。

- 紧凑型优化算法（81 种配置），
- 单解算法（60 种配置），

为确保清晰性和可复现性，这些算法配置的具体组成和设置均在一份专门文档中作了完整说明，该文档可从 [18]获取。

4.2 结果

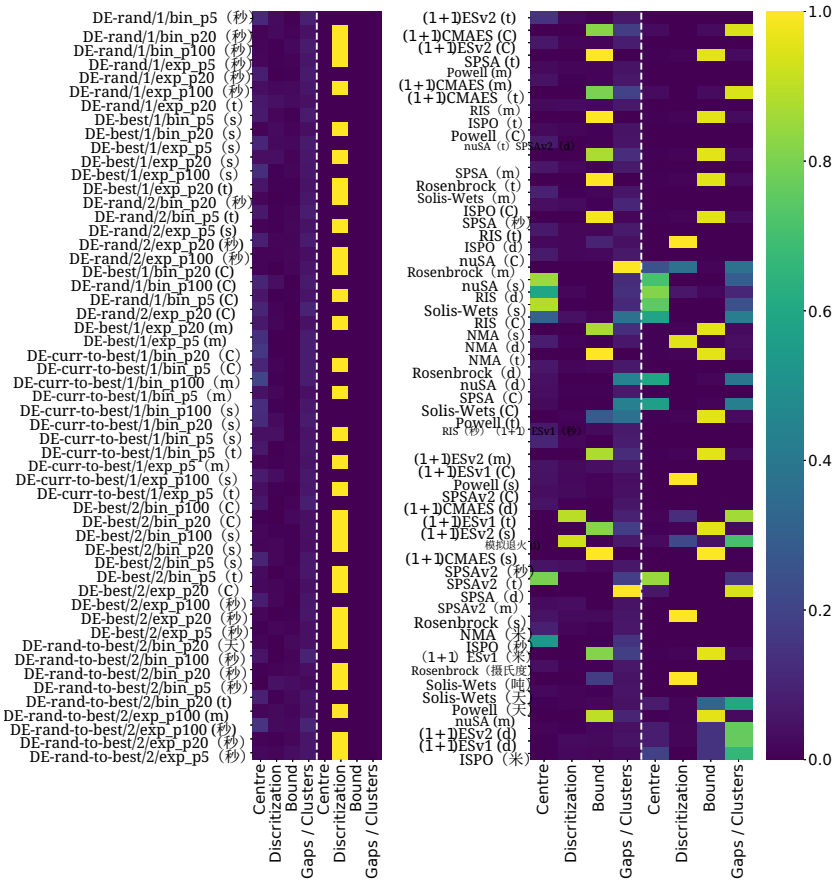
首先，我们在单解算法集合上，将 Deep-BIAS 方法所作的判断与原始工具箱的判断进行比较。具体做法是，在图5(b)中展示两种方法针对每种算法给出的类别概率。从图中可以看出，在大多数情况下，两种方法对是否有偏的判断相同。然而，对于相当一部分算法，两种方法检测到的偏差类型并不相同。当原始工具箱的随机森林（RF）模型预测为“簇”时，通常会出现这种情况。这可能部分归因于两个模型训练数据的差异；此外，在某些簇设置中，簇可能位于边界附近，使这两个类别之间的区分变得有些模糊。

除了比较单解算法之外，我们还可以研究差分进化不同版本所产生的影响，从而深入了解算法配置与检测到的偏差类型之间的关系。在这种情况下，我们只重点展示两个模型对有偏/无偏作出不同判断的算法，并在图 5(a) 中展示由此得到的类别概率。显然，这里有两种截然不同的情况：对于大多数被 RF 判定为有偏的模型，RF 预测的偏差类型是“离散化”。这与这些配置均采用“饱和”边界修正方法这一事实相符；该方法会将任何不可行解重新置于边界上。因此，如果某个算法生成了一定比例的不可行解，那么至少部分坐标恰好落在边界上的概率就很高，导致某些坐标出现多个相同值。对于均匀分布而言，这种情况极不可能发生，因此统计检验很容易检测到这种差异。

对于图 5(b) 和图 5(a)，还需注意，对于 RF 模型，类别概率按设计总和为 1；而深度模型并非如此，因为均匀类别仍会获得一部分概率质量。因此，Deep-BIAS 的结果间接反映了偏差的强度。

为了更深入地了解 Deep-BIAS 模型与原始工具箱结果不同的场景，我们可以聚焦于两个预测结果不一致的配置。图 6 展示了两个场景，突显了这两种偏差检测方法在行为上的差异。图 6(a) 展示了一种 DE 变体：统计检验明确检测到非均匀性，而 Deep-BIAS 模型则未发现结构偏差的有力证据。观察点的分布可以发现，许多维度上有多个点恰好位于边界处。这是“饱和”边界修正方法产生的自然结果，因为该方法会将定义域之外生成的点置于最近的边界上。由于均匀分布生成的点恰好落在边界上的概率为 0，因此，多个点位于边界的情况似乎很容易被各种检验检测出来。然而，Deep-BIAS 模型并不具备这一背景信息，因此对边界饱和现象不那么敏感。

在图 6(b) 中，我们展示了相反的情况：统计检验未被拒绝，但 Deep-BIAS 模型始终将这些点分类为非均匀分布。有趣的是，从视觉上看，该分布似乎在空间中心有稍多的



(a) 差分进化配置　　(b) 单解算法

图 5：Deep-BIAS（白色虚线左侧）和原始偏差方法（白色虚线右侧）对差分进化配置（左侧）和单解算法（右侧）集合中的非均匀性类型所作的预测；在这些集合中，两种方法给出的有偏／无偏判定不同。对于原始方法，当配置被判定为有偏时，概率由随机森林模型确定；否则设为 0。对于 Deep-BIAS，预测值是逐维预测的平均值。右侧的色条适用于两个子图。

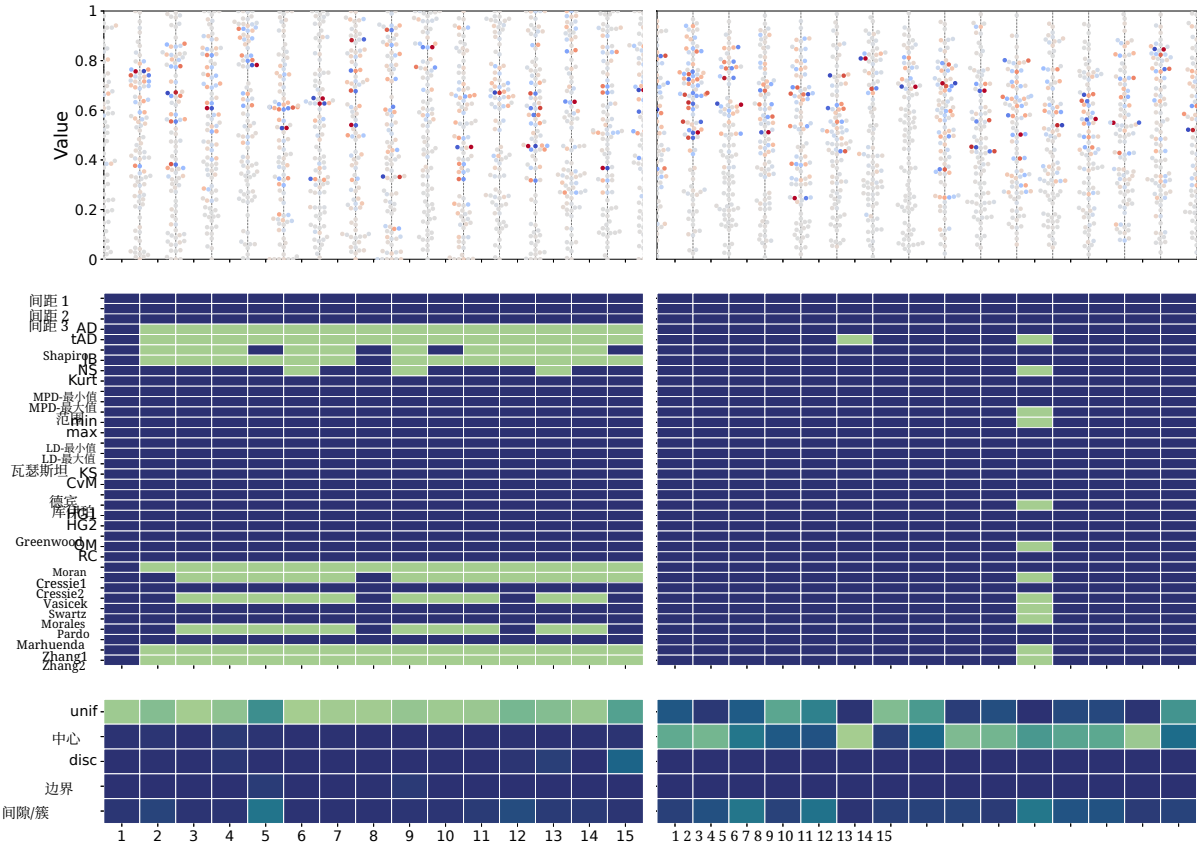
质量。然而，偏差非常小，因此在应用必要的 p 值校正后，这些检验的把握不足以拒绝零假设。由于这种情况没有足够的检验拒绝，我们或许可以认为它不具有偏差。不过，由于深度模型不局限于二元决策，我们可以利用其输出的概率，将这种情况标示为偏差程度非常低。这说明，结合使用这两种方法，可以更充分地判断结构偏差是否存在、其强度及类型。

5 可复现性声明

我们在 Github 上提供了 Deep-BIAS 软件包的开源、附有文档的实现版本 [4]。实验、更多图表和文档也可在该处找到。预训练模型可在同一代码库的“models”目录下获取。所有模型均使用单块 T4 GPU 进行训练，并通过使用太阳能实现碳中和计算（无 CO_2）。

6 结论

根据本研究所进行的分析，我们得出结论：使用深度学习检测结构偏差（SB）是可行的，并能取得令人满意的结果。仅用 50 个样本，我们就能正确检测出大多数均匀分布的点，而且



(a) DE-best/1/bin_p20 (秒)　　(b) Solis-Wets（C）

图6：两种算法的位置（上）、统计检验结果（中）和深度预测（下）概览（展示了30个维度中的第1至第15个）。上方部分的颜色基于 SHAP 值，具体说明见图3的图注。左侧图例适用于两个子图。

样本量越大，性能越好。我们发现，用于检测结构偏差（SB）的最优网络架构并不像文献中其他深度学习应用常设计的架构那样复杂，但其表现与原始 BIAS工具箱相似。原始工具箱基于统计检验，并且在 F1 分数这一评估指标上表现出色。

与 BIAS 相比，所提出的 Deep-BIAS 替代方案展现出一些有趣的特征和有利表现。

- 在对结构偏差的类型进行分类方面，它优于基于统计检验的方法。
- 它通过使用类别概率，更好地衡量结构偏差的强度。
- 通过使用可解释人工智能（XAI），该方法提供了更多见解，并可进一步分析分布中的感兴趣区域，以了解分类背后的机制。

使用神经网络模型的一个缺点是，与采用统计检验方法得到的SB检测系统相比，由此得到的SB检测系统泛化能力较弱，因为网络可能会漏检训练数据中未包含的SB类型。不过，为Deep-BIAS准备的训练数据集似乎足够充分，而且网络未能发现偏差的情况主要出现在使用饱和约束处理方法时。这对BIAS而言并不是问题。因此，本研究的总体结论是，联合使用这两个系统能够实现最佳的SB检测。我们建议将BIAS用作偏差/无偏差二元分类的主要方法，并使用Deep-BIAS检查SB的类型并确定其强度。

参考文献

[1] D. Wolpert 和 W. Macready。优化的无免费午餐定理。*IEEE* 进化计算汇刊，1:67–82，1997。[2] Nikolaus Hansen、Anne Auger、Raymond Ros、Olaf Mersmann、Tea Tušar 和 Dimo Brockhoff。Coco：一个用于在黑箱设置下比较连续优化器的平台。优化方法与软件，36(1):114–144，2021。[3] Diederick Vermetten、Bas van Stein、Fabio Caraffini、Leandro L. Minku 和 Anna V. Kononova。Bias：用于在连续域中对结构偏差进行基准测试的工具箱。*IEEE* 进化计算汇刊，26(6):1380–1393，2022。[4] 匿名作者。Deep-bias v1.0.0，2023年1月。[5] Anna V. Kononova、David W. Corne、Philippe De Wilde、Vsevolod Shneer 和 Fabio Caraffini。基于种群的算法中的结构偏差。信息科学，298:468–490，2015。[6] Alfred Inselberg。具有平行坐标的平面。视觉计算机，1(2):69–91，1985。[7] Fabio Caraffini、Anna V. Kononova 和 David W. Corne。差分进化中的不可行性与结构偏差。信息科学，496:161–179，2019。[8] Bas van Stein、Fabio Caraffini 和 Anna V. Kononova。差分进化中结构偏差的涌现。载于*2021*年遗传与进化计算会议论文集（伴随卷），GECCO '21 伴随卷，美国纽约州纽约市，2021年7月。计算机协会。[9] Fabio Caraffini 和 Anna V. Kononova。优化算法中的结构偏差：扩展结果，2021。[10] Bas van Stein、Fabio Caraffini 和 Anna V. Kononova。差分进化中结构偏差的涌现——源代码与扩展图形结果，2021。[11] Andrey Nikolaevich Kolmogorov。关于分布律的经验确定。意大利精算师协会杂志，4:83–91，1933。[12] Sandor Csorgo 和 Julian J. Faraway。Cramer-von Mises统计量的精确分布与渐近分布。英国皇家统计学会杂志*B*辑（方法论），58(1):221–234，1996。[13] Theodore W Anderson 和 Donald A Darling。基于随机过程的某些"拟合优度"准则的渐近理论。数学统计年刊，第193–212页，1952。[14] Yoav Benjamini。发现假发现率。英国皇家统计学会杂志*B*辑（统计方法），72(4):405–416，2010。[15] Anna V. Kononova、Fabio Caraffini、Hao Wang 和 Thomas Bäck。紧凑型优化算法是否存在结构偏差？载于 T. Bäck、M. Preuss、A. Deutz、H. Wang、C. Doerr、M. Emmerich 和 H. Trautmann 编，自然中的并行问题求解——*PPSN XVI*，第229–242页，瑞士 Cham，2020。Springer International Publishing。[16] S. S. Kar 和 A. Ramalingam。30是神奇数字吗？样本量估计中的问题。国家社区医学杂志，4(1):175–179，2013。[17] Diederick Vermetten、Bas van Stein、Fabio Caraffini、Leandro L. Minku 和 Anna V. Kononova。Bias：用于在连续域中对结构偏差进行基准测试的工具箱。*IEEE* 进化计算汇刊，26(6):1380–1393，2022。[18] Diederick Vermetten、Anna V. Kononova、Fabio Caraffini、Bas van Stein 和 Leandro Minku。Bias：用于在连续域中对结构偏差进行基准测试的工具箱——代码，2021年9月。[19] Haifeng Jin、Qingquan Song 和 Xia Hu。Auto-keras：一种高效的神经网络架构搜索系统。载于第*25*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，第1946–1956页。ACM，2019。[20] Tjeerd van Campen、Herbert Hamers、Bart Husslage 和 Roy Lindelauf。一种应用于世贸中心9·11恐怖袭击事件的Shapley值新近似方法。社交网络分析与挖掘，8(1):1–12，2018。[21] L. Shapley。n人博弈的一个值。载于 H. Kuhn 和 A. Tucker 编，博弈理论文集*II*，第307–317页。普林斯顿大学出版社，1953。