arXiv:2601.08657v1 [cs.NE] 13 Jan 2026

NEVO-GSPT：使用膨胀与收缩算子的基于种群的神经网络进化

Davide Farinati[1,3],Frederico JBB Santos[2], Leonardo Vanneschi[1], 和 Mauro Castelli[1]

[1] 里斯本新大学 NOVA 信息管理学院（NOVA IMS），里斯本新大学，坎普利德校区，1070-312 里斯本，葡萄牙{dfarinati, lvanneschi, mcastelli}@novaims.unl.pt[2] 的里雅斯特大学工程与建筑系，意大利。fredericojose.jacomedebritosantos@phd.units.it[3] 圣拉斐尔生命健康大学，意大利米兰，综合癌症中心/泌尿科；URI；圣拉斐尔医院 IRCCS，意大利米兰 farinati.davide@hsr.it

摘要。 神经网络架构的进化是一个计算密集型过程。传统方法通常需要在庞大的架构空间中进行广泛搜索，而且难以深入了解结构修改如何影响模型行为。本文提出了基于几何语义扰动和基于种群训练的神经进化（NEVO-GSPT），这是一种建立在两项关键创新之上的新型神经进化算法。首先，我们将遗传编程中的几何语义算子（GSO）改造用于神经网络进化，确保架构变化在单峰误差曲面上对网络语义产生可预测的影响。其次，我们提出了一种新型算子（DGSM），可在保持 GSO 语义属性的同时，有效控制网络规模的缩减。与传统方法不同，NEVO-GSPT 的高效评估机制只需计算新增组件的语义，因此能够高效开展基于种群的训练，以极低的计算成本全面探索搜索空间。在四个回归基准测试上的实验结果表明，NEVO-GSPT 能够稳定地进化出紧凑型神经网络，其性能可与文献中已有方法相媲美或更优，包括标准神经网络、SLIM-GSGP、TensorNEAT 和 SLM。

关键词： 神经进化、进化机器学习、人工神经网络、基于种群的训练、几何语义变异

1 引言

设计有效的人工神经网络（人工神经网络）架构，是现代机器学习的核心挑战之一。尽管人工神经网络

在众多领域取得了成功，但要确定适用于特定任务的最优架构，通常仍需进行大量人工实验，或付出高昂的计算成本进行自动搜索 [11]。传统优化方法（如网格搜索和随机搜索）必须从头开始评估每种架构，且无法以有原则的方式将架构修改与其对网络行为的影响联系起来，因此既效率低下，也缺乏透明度。神经架构搜索（NAS） [9]能够自动执行这一过程，但计算代价高昂，往往需要数千 GPU 小时，因此只有少数机构能够负担。此外，神经架构搜索方法通常将架构优化视为黑盒问题，难以解释或理解某些设计为何优于其他设计。神经进化（NE） [24], 通过进化算法优化人工神经网络，是一种颇具吸引力的替代方案。早期的神经进化方法，例如增强拓扑神经进化（NEAT） [29],，通过有原则的遗传表示成功地同时进化拓扑结构和权重；但它们难以扩展到现代深层架构，而且由于必须对每个个体进行完整训练或评估，计算成本仍然很高。近期的神经进化变体 [7,14]试图降低这些成本，但核心挑战依然存在：如何在保持结构与行为之间关系可预测的同时，高效探索架构空间。

为解决这一问题，本文提出了基于几何语义扰动与种群训练的神经进化（NEVO-GSPT），这是一种兼顾计算效率与语义感知能力的新型神经进化方法。NEVO-GSPT 将几何语义遗传编程（GSGP） [17] 的基本原理及其近期发展——基于膨胀与收缩变异的语义学习算法（SLIM-GSGP） [32,33]——应用于人工神经网络的进化。NEVO-GSPT 扩展了一个名为语义学习机（SLM） [12],的既有框架，采用对个体语义产生可预测几何效应的算子，并为任何监督学习任务构造单峰误差曲面。与 SLM 相比，NEVO-GSPT 的一项关键创新是引入了收缩几何语义变异（DGSM）。传统的膨胀算子（IGSM）通过添加扰动组件来扩展网络，而 DGSM 则通过移除先前添加的元素来降低复杂度。交替进行膨胀与收缩可避免模型失控增长，从而在保留几何语义算子的优势的同时，进化出紧凑且可解释的网络。NEVO-GSPT 的计算效率也很高，因为它借鉴了 SLIM-GSGP 的思路，将网络表示为由扰动组件构成的链表。发生变异时，只需评估新增或移除的组件，并复用先前的评估结果。这种增量机制无需重新训练整个网络，大幅降低了适应度计算成本。我们在四个回归问题上开展了实验研究，旨在考察四个关键因素：（1）对初始种群进行先验反向传播训练的效果；（2）对进化模型进行事后微调的益处；（3）不同膨胀–收缩概率的影响；以及（4）新增组件复杂度的影响。

本文结构安排如下。第 2 节回顾神经进化领域的相关工作。第 3 节介绍背景知识，帮助读者了解本文的研究脉络。第 4 节阐述本文的方法，详细介绍用于神经进化的 IGSM 和 DGSM，以及所采用的基于种群的训练方法。第 5 节介绍实验研究。最后，第 6 节总结全文，并讨论未来的研究方向。

2 相关工作

2.1 神经进化方法

自 NEAT [29],等早期方法以来，神经进化领域取得了显著发展。这些方法通过物种分化和渐进式复杂度增长，实现了网络拓扑与权重的共同进化。HyperNEAT [28]和 ES-HyperNEAT [23] 等扩展方法采用基于组合模式生成网络（CPPN）的间接编码，以进化具有规律性的大规模网络，但主要针对浅层架构。近年来，更多研究开始关注深层架构：CoDeepNEAT [16] 引入两级进化过程来进化深度神经网络，其中一个种群进化模块（层的组合），另一个种群则进化这些模块如何组装成完整网络；CGP-CNN（用于卷积神经网络的笛卡尔遗传编程） [31,30] 使用笛卡尔遗传编程将卷积神经网络表示为有向无环图；DENSER [2] 等方法则采用动态结构化文法进化。尽管这些方法在 ImageNet 和 CIFAR 等基准测试上取得了具有竞争力的结果，但它们面临一个共同挑战：计算成本高昂，通常需要数百乃至数千个 GPU 日 [22]。这一计算障碍限制了它们在实际中的应用和可及性。

2.2 基于语义的方法

一种截然不同的方法源自几何语义遗传编程（GSGP），该方法引入了对程序语义具有可预测影响的算子，从而产生单峰适应度景观。语义学习机（SLM） [12] 将这些原理应用于神经网络构建，通过几何语义变异逐步添加神经元，并限制其对输出的贡献。然而，SLM 存在以下局限性：（1）它不使用基于梯度的优化（反向传播），而是完全依赖进化进行参数优化；（2）尽管进化得到的网络比传统的几何语义遗传编程（GSGP）树更紧凑，但其规模仍可能显著增长。SLIM - GSGP [20] 通过在膨胀突变之外引入收缩突变，解决了规模增长问题，使程序规模能够扩展和缩减，同时保持单峰适应度景观的特性。与标准 GSGP 相比，该方法在缩减模型规模方面已展现出良好效果，但其主要应用于符号回归，而非神经网络进化。

总而言之，文献中仍存在一个显著的空白：亟需一种神经进化方法，能够高效地演化紧凑且可解释的神经网络架构，同时在现代基准测试中保持有竞争力的表现 [35]。这种方法需要：（1）利用基于梯度的优化来学习参

数，同时演化拓扑；（2）严格控制个体的规模；（3）形成有利的适应度景观特性，以促进高效搜索；（4）实现实用的训练时间，以 GPU 分钟而非 GPU 日计量。本文提出的方法正是结合这四项指导原则设计的。

3 背景

3.1 几何语义遗传编程

在监督学习任务中，遗传编程（GP）个体通常表示为将输入值转换为输出结果的计算机程序。对于给定个体 P 和包含 n 个适应度案例 $X = \{x_1, x_2, \ldots, x_n\}$ 的数据集，其生成的输出向量 $s_P = \{P(x_1)\ P(x_2) \ldots, P(x_n)\}$ 称为该个体的语义 [17]。传统 GP 算子（如交叉和变异）会在语法层面修改程序结构，因此其语义效应难以预测，还可能破坏搜索局部性 [18]。

GSGP [17] 通过用几何语义算子（GSO）取代语法算子来解决这一问题。这些算子会在语义空间中强制实现特定的几何行为，从而使任何监督学习任务都具有单峰误差曲面。对于父函数 $T : \mathbb{R}^n \to \mathbb{R}$，几何语义变异（GSM）将新函数定义为 $T_M = T + \text{ms} \cdot (T_{R1} - T_{R2})$，其中 ms 表示变异步长，$T_{R1}$ 和 T_{R2} 是输出值位于 $[0, 1]$ 内的随机函数。这确保变异后的个体位于以父代语义为中心、半径为 ms 的超球体内。

受 GSGP 成功的启发，近年来人们提出了各种扩展方法 [19,5,3]。然而，该方法的一项显著缺点是，子代往往比其父代更大，导致生成的解通常难以解释 [17]。

3.2 基于膨胀与收缩突变的语义学习算法

为了解决 GSGP 的膨胀问题，近期研究 [32] 提出了收缩几何语义变异（DGSM），该方法在保留语义属性的同时缩小个体规模。新提出的算子建立在两个核心思想之上。首先，上一节介绍的膨胀几何语义变异（IGSM）可通过代数运算改写为：

$$IGSM\,(T) = T + \text{ms} \cdot (T_{R1} - T_{R2}) = T - \text{ms} \cdot (T_{R2} - T_{R1})\,,$$

基于恒等式 $(T_{R1} - T_{R2}) = -(T_{R2} - T_{R1})$。其次，由于 T_{R1} 和 T_{R2} 是从同一分布中随机生成的，因此二者可以互换。因此，IGSM 的另一种有效表达式为：

$$IGSM\,(T) = T - \text{ms} \cdot (T_{R1} - T_{R2})\,.$$

考虑一个 T 经过三次 IGSM 操作后得到的个体：

$$T + \text{ms} \cdot (T_{R1} - T_{R2}) + \text{ms} \cdot (T_{R3} - T_{R4}) + \text{ms} \cdot (T_{R5} - T_{R6})\,.$$

使用 IGSM 的另一种基于减法的公式，我们可以应用第四次变异：

$$T + \mathrm{ms} \cdot (T_{R1} - T_{R2}) + \mathrm{ms} \cdot (T_{R3} - T_{R4}) + \mathrm{ms} \cdot (T_{R5} - T_{R6}) - \mathrm{ms} \cdot (T_{R7} - T_{R8}) .$$

这会继续扩展个体。不过，通过重用一对较早生成的随机程序（例如 T_{R3} 和 T_{R4}），可以简化该个体：

$$T + \mathrm{ms} \cdot (T_{R1} - T_{R2}) + \cancel{\mathrm{ms} \cdot (T_{R3} - T_{R4})} + \mathrm{ms} \cdot (T_{R5} - T_{R6}) \cancel{- \mathrm{ms} \cdot (T_{R3} - T_{R4})}.$$

中间项相互抵消，从而得到更紧凑的个体。DGSM 通过移除先前添加的项来实现这一点，同时保留 GSM 的所有语义属性，包括有界扰动和单峰误差曲面。顾名思义，基于膨胀与收缩变异的语义学习算法（SLIM-GSGP）结合了 IGSM 和 DGSM，并允许配置二者的使用概率；当只剩一项时，则限制使用 DGSM。

3.3 语义学习机器

SLM [12]将 GSGP 原理应用于前馈神经网络。具体而言，在对父网络 T 进行变异时，SLM 会通过添加一个随机生成的网络来创建子代，并将该网络的输出乘以 ms进行缩放，从而确保语义变化始终限制在以父网络语义为中心、半径为 ms的球体内。这种方法保证子代处于受控的语义空间区域中，类似于 GSGP 中几何语义变异（GSM）的球形变异特性。SLM 的局限性包括：(1) 不使用反向传播进行参数优化（随机网络的参数在初始化后保持固定），只能依靠进化来提升网络性能；(2) 尽管比 GSGP 树更紧凑，网络仍可能大幅增长。

3.4 张量化拓扑增强神经进化

张量化拓扑增强神经进化（TensorNEAT） [34]解决了神经进化中的一个计算瓶颈：无法在 GPU 架构上高效并行执行 NEAT 的操作。TensorNEAT 将具有不同拓扑结构的网络张量化为形状统一的张量，以实现批量 GPU 计算，从而克服了这一限制。网络被编码为带填充的固定大小张量，使所有 NEAT 操作（变异、交叉、物种分化和网络推理）都能在整个种群范围内进行向量化，并在 GPU 上并行执行。与传统实现相比，TensorNEAT 显著提升了运行速度。实验结果表明，与使用最广泛的开源 NEAT 实现 NEAT-Python 相比，速度提升超过 $500\times$ 。

4 方法

在本研究中，我们提出了 NEVO-GSPT，这是一种新颖的神经进化方法，融合了 GSGP 和 SLIM-GSGP 的原理，用于演化神经网络（NN）的种群。该方法同时演化种群中神经网络的架构、权重和激活函数。该算法依赖两个核心算子：IGSM 和 DGSM，二者均受第 3 节所述 GSO 的启发。

本研究的目标是重现 GSGP 和 SLIM-GSGP 中所用算子对神经网络的作用效果。

4.1 用于神经进化的膨胀几何语义变异

IGSM 是一种神经进化算子，旨在向现有神经网络 N 引入可控变化，同时保留其原有功能。为此，算子会附加一个新的辅助随机网络 R，作为父模型的扰动。该网络 R 按照以下条件构建：

– 每层恰好包含一个神经元，因此对于一个有 n 层的网络，共有 n 个神经元。– 它仅接收来自父网络的连接；不允许向 N 添加反向连接。这可确保 N 的语义保持不变。– R 中的所有内部权重均从范围 $[-1, 1]$ 内均匀抽取，最后一个输出权重除外。– 输出权重从 $[0, ms]$ 中抽取，其中 $ms \in \mathbb{R}$ 是一个可调参数，称为变异步长。– R 的输出神经元使用 TanH 激活函数，以确保 $[-1, 1]$ 中的输出有界。– R 中的其他神经元可以使用任意激活函数。

构造完成后， R 的语义贡献会按变异步长缩放，并加到 N的输出上，从而生成最终的子代网络。IGSM 保证扰动的整体效果处于范围$[-ms, ms]$内，因此能够对父网络的行为进行有界的加性修改。此操作可简洁地表示为：

$$\mathrm{IGSM}(N) = N + ms \cdot R \tag{1}$$

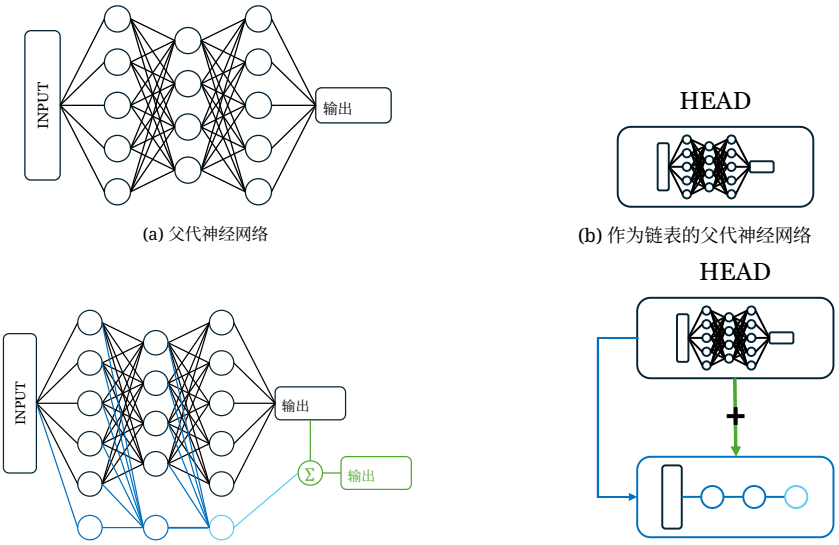
与 SLIM-GSGP 算法一样，子代生成的结构可以表示为链表。图 1 展示了应用 IGSM 变换前后，标准神经网络表示及其对应链表的情况。

4.2 用于神经进化的收缩几何语义变异

DGSM 是 IGSM 的一种互补算子，可在进化神经网络中实现语义简化。IGSM 会逐步向神经网络的输出添加新的扰动组件，而 DGSM 则通过移除先前添加的扰动来逆转这一过程。这样既能降低模型复杂度，又不损害语义保真度。

在神经进化中，DGSM 的实现方式是识别先前添加的辅助网络 R_i （通过 IGSM 引入），并从当前网络中减去其贡献。对于经过多次 IGSM 变换的神经网络 N，其输出可表示为：

$$N' = N + ms \cdot R_1 + ms \cdot R_2 + \cdots + ms \cdot R_k$$



(a) 父代神经网络 (b) 作为链表的父代神经网络

(c) 将 IGSM 应用于神经网络 (d) 以链表形式应用 IGSM图 1：使用神经网络和链表表示法展示 IGSM。

DGSM随机选择一个 $R_j \in \{R_1, R_2, \ldots, R_k\}$ 并移除其贡献：

$$DGSM(N') = N' - ms \cdot R_j$$

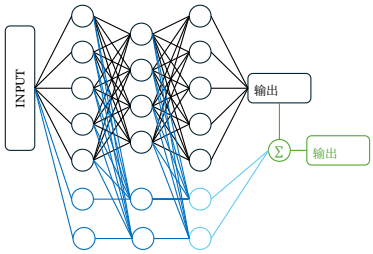
该操作保留了几何语义变异的核心语义特性：它会对网络输出施加有界的线性扰动，并保持误差景观的单峰结构。然而，它同时通过缩小进化模型的有效规模，抵消了重复加性扰动固有的膨胀问题。

为实现这一行为，必须使用类似链表的结构维护扰动组件的历史记录，其中每个元素都对应于此前通过 IGSM 添加的特定辅助网络。DGSM 的作用是从该列表中删除一个元素。

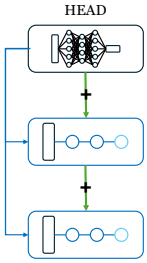
图 2 展示了以神经网络和链表形式应用 DGSM 的情况。

4.3 基于种群的训练

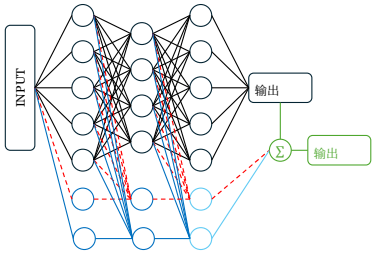
新个体的评估通过对链表中所有元素的语义求和来完成，正如先前工作 [33] 中所提出的那样。因此，当应用 IGSM 时，只需评估一个宽度为 1 的 NN，因为所有其他组件的语义都已经计算完毕。相比之下，当使用 DGSM 时，可以通过简单求和来确定新个体的语义和适应度。这使评估过程非常高效，从而能够演化出一个个体种群。维护一个种群可以更全面地探索与给定问题相关的适应度景观。



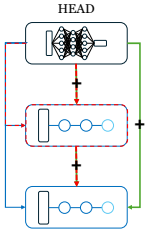
（a）经过两次 IGSM 后的父代神经网络



（b）以链表形式表示的经过两次 IGSM 后的父代神经网络



（c）应用于网络的 DGSM



（d）以链表形式应用的 DGSM图 2：使用神经网络和链表表示展示 DGSM。

5 实验分析

本节介绍了为评估 NEVO-GSPT 而开展的实验。内容分为三个部分。第 5.1 节介绍用于评估所提出算法性能的主要基准测试。第 5.2 节深入分析 NEVO-GSPT 的各种超参数和配置。最后，第 5.3 节报告了 NEVO-GSPT 与来自进化计算（EC）、神经进化（NE）和机器学习（ML）的其他基线算法相比的性能。

5.1 实验设置

为评估 NEVO-GSPT，我们选择了四个常用于评估 GP [1,27,6,4,13,15,21]、NE [12] 和 NN [8]的基准回归数据集。所用数据集及其特征见表 1。这些数据集呈现出不同的过拟合特征 [15,10], ，因此可以用于研究 NEVO-GSPT 的泛化能力。

数据集	**观测数量 特征数量**	
翼型	1502	5
混凝土强度	1029	8
生物利用度	359	241
ld50	234	626

表 1：用于模型比较的数据集特征。

我们分析了若干超参数，并在第 5.2 节确定了它们的最优值。其他超参数在所有实验中均保持固定，其中种群规模设为 100，变异步长设为 2。所有比较均采用 80-20 训练—测试划分的蒙特卡洛交叉验证，并确保各方法在每次运行中使用相同的数据划分。第 5.2 节所述实验进行了十次独立运行，而第 5.3 节所述实验进行了三十次运行。后者实验中成对 Wilcoxon 符号秩检验所得的相应 p 值见补充材料。本研究使用的 Python 代码可在 NEVO-GSPT[4] 中找到。

5.2 消融研究

本节深入分析算法的配置和超参数，重点在于确定每个数据集的最佳设置，并解释其有效性的根本原因。

先验训练在种群初始化期间，$|P|$ 神经网络以不同的结构和权重随机生成，其中 $|P|$ 表示种群规模。每个神经网络都可以通过反向传播进行训练，也可以直接进入进化过程。本节研究先验训练 （AprT）对进化过程的影响。对于每个数据集，我们评估了三种配置：不进行 AprT、对一半神经网络（随机选取）进行 AprT，以及对所有神经网络进行 AprT。图 3 展示了各数据集在训练集和测试集上的均方根误差（RMSE）结果。

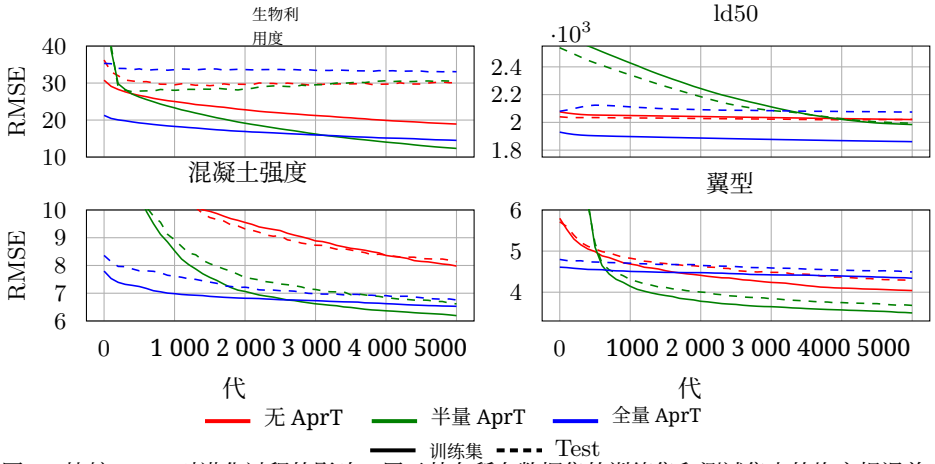


图 3：比较 AprT 对进化过程的影响，展示其在所有数据集的训练集和测试集上的均方根误差表现。

图 3 中的结果表明，在所有

[4] https://github.com/Fari98/NEVO-GSPT

数据集上，Half AprT 配置始终能在训练集和测试集均方根误差之间取得最佳平衡。No AprT 收敛较慢且误差较高，而 Full AprT 在某些情况下容易过拟合；相比之下，Half AprT 收敛更快，泛化性能更优。这表明，适度集成 AprT 能够最有效地引导进化过程。

后验训练NEVO-GSPT 生成的最终解是神经网络，这些网络还可以进一步训练，以优化其权重。本节研究了后验训练 （ApoT）对最后一代所获得最终解的影响。结果展示了各数据集的测试集均方根误差，见图 4。

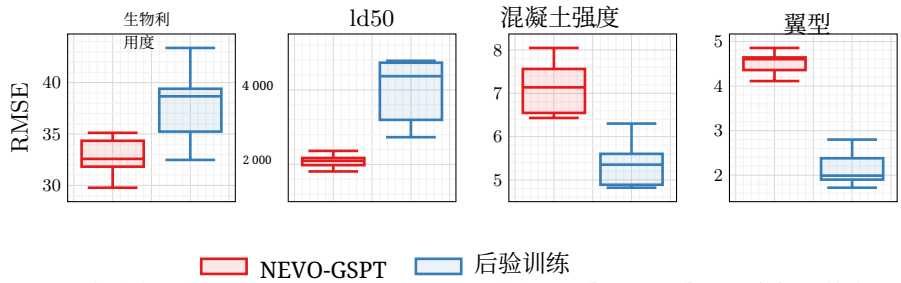


图 4：比较后验训练对进化过程的影响，展示了所有数据集上测试集的均方根误差表现。

如结果所示，后验训练对不易过拟合的数据集（即混凝土强度 和翼型）有积极影响，改善了测试集均方根误差。-然而，对于容易过拟合的数据集（即生物利用度 和半数致死量），则观察到相反趋势：后验训练会降低测试表现，并削弱算法的泛化能力。

收缩与膨胀概率膨胀概率和收缩概率的不同组合可用于平衡收敛速度与程序规模之间的权衡。在本实验中，我们测试了四种膨胀概率值（0.3、0.5、0.7 和 1），对应的收缩概率始终设为 1 减去膨胀概率。此项分析的结果见图 5 和图 6，其中展示了训练集和测试集上的均方根误差表现，以及由此得到的程序规模。

正如预期，较低的膨胀概率会产生更小的程序。然而，其对性能的影响则不那么直观。对于不易过拟合的数据集（即，混凝土强度 和 翼型），较低的膨胀概率会导致收敛速度变慢。相反，对于容易过拟合的数据集（即，生物利用度 和 半数致死量），较低的膨胀概率会提升算法的泛化能力。这些结果进一步印证了 [10] 中报告的观察：对于这类问题，程序规模往往与过拟合相关。

待添加的神经元如第4.1节所述，应用 IGSM 算子时，会为父代神经网络的每一层添加一个神经元。不过，也可以生成更复杂或更简单的网络。本节将探讨

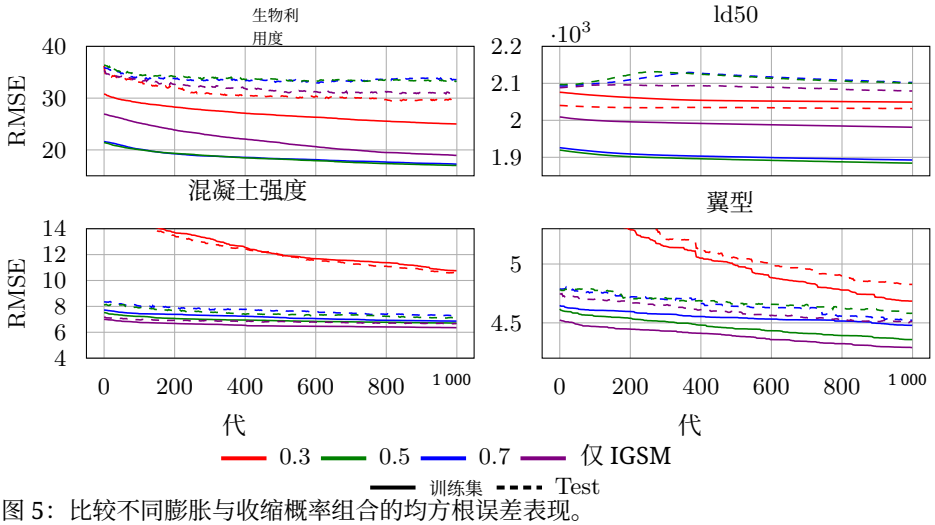


图 5：比较不同膨胀与收缩概率组合的均方根误差表现。

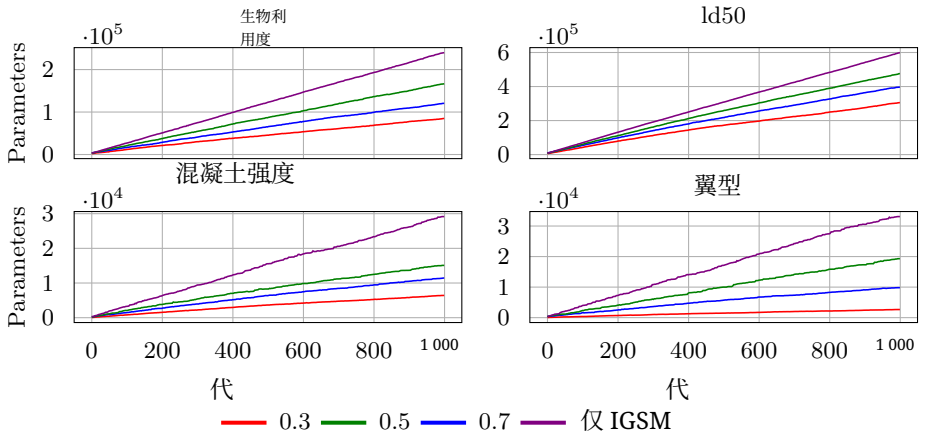


图6：比较不同膨胀与收缩概率组合的大小。

通过将不同的比例设为超参数（0.3、0.5、0.7 和 1），测试每层添加较少神经元的影响。图7展示了这些配置各自的训练集和测试集均方根误差表现。

如图7所示，每层添加少于一个神经元会持续导致算法收敛速度变慢，泛化能力下降。

5.3 实验结果

在探索 NEVO-GSPT 的超参数并为每个数据集选出最佳配置后，我们明确区分了容易过拟合与不易过拟合的配置，并将该算法与一组

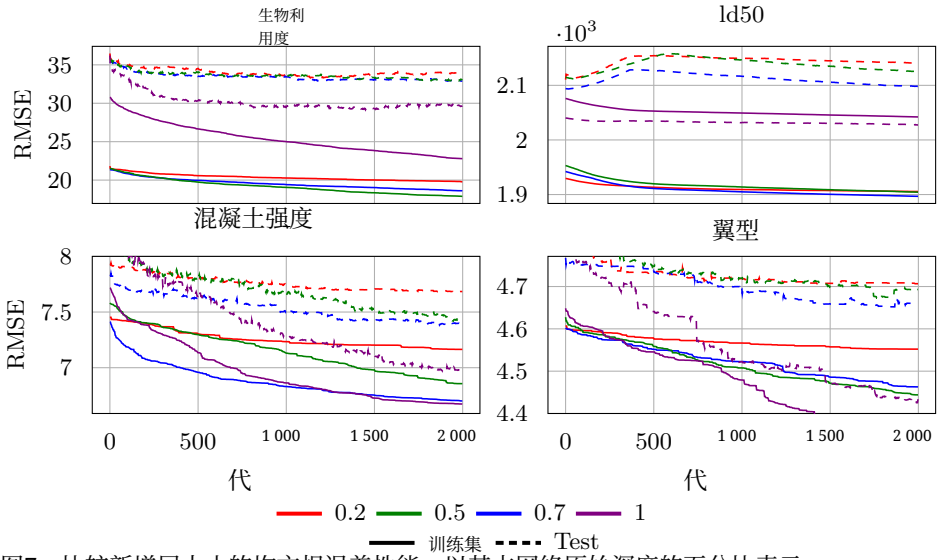


图7：比较新增层大小的均方根误差性能，以其占网络原始深度的百分比表示。

来自进化计算和神经进化的基线方法进行比较。其中包括一个与 NEVO-GSPT 所演化出的模型具有相同架构的神经网络，其权重通过反向传播优化；还包括 SLM [12], TensorNEAT [34],和 SLIM-GSGP [25]（变体 *SLIM*SIG1*）。为确保公平比较，所有进化算法均采用一致的超参数配置，使其参数数量与 NEVO-GSPT 演化出的模型相匹配。TensorNEAT 运行 1000 代，种群规模为 100，最多使用 627 个节点和 626 个连接，每个物种包含 10 个个体，生存阈值为 0.2。对于 TensorNEAT 的初始种群，连接和节点的数量必须至少与输入特征数量相同，因此其数值高于 NEVO-GSPT。SLM 运行 100 代，使用最多包含 3 层的网络，初始神经元数量为 32，并在每次 GSM 突变时为每层增加 4 个神经元。请注意，由于 SLM 的实现（PyTorch Genesis [26]）存在限制，当每代每层仅增加一个神经元时会产生显著的计算开销，因此必须调整参数，增加每次变异时添加的神经元数量，同时减少代数。我们据此调整参数，以确保比较公平。图8展示了不同算法在所有数据集上的测试集 RMSE。Wilcoxon 符号秩检验的 p 值见补充材料。

如图8所示，NEVO-GSPT 是唯一一种在所有任务中始终名列前茅的算法。这凸显了基于种群的训练与几何突变相结合，在实现强劲收敛能力和稳健泛化方面的有效性。

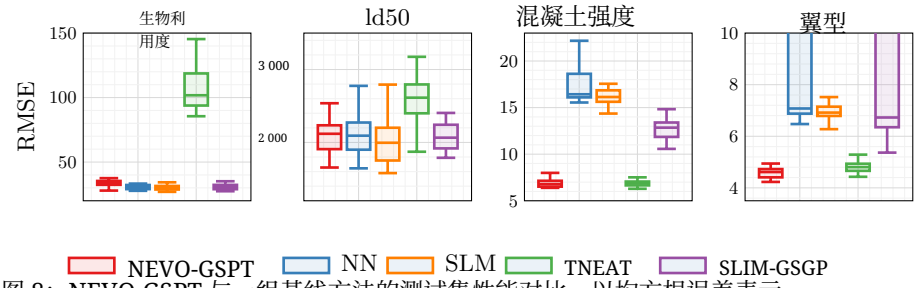


图 8：NEVO-GSPT 与一组基线方法的测试集性能对比，以均方根误差表示。

程序规模比较这项工作的创新之处还在于为神经进化引入了一种 DGSM 算子，旨在生成更紧凑的神经网络。图9比较了 NEVO-GSPT 与另外两种神经进化技术 TensorNEAT 和 SLM 在所有数据集上生成的神经元数量。

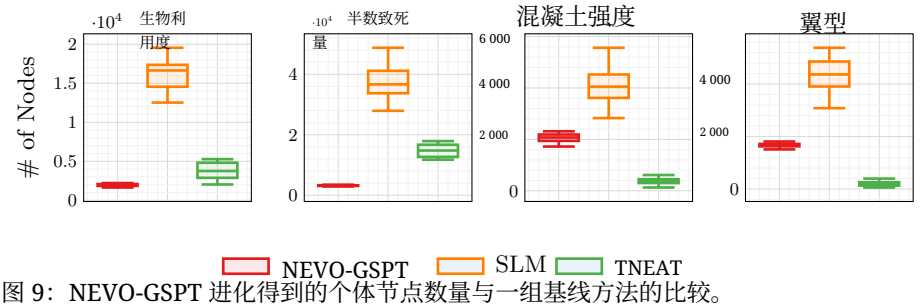


图 9：NEVO-GSPT 进化得到的个体节点数量与一组基线方法的比较。

如所有测试用例所示，NEVO-GSPT 始终生成比 SLM 更小的神经网络，证明了所提出的 DGSM 算子的有效性。此外，在四个数据集中有两个数据集上，NEVO-GSPT 生成的网络也比 TensorNEAT 更小；TensorNEAT 是神经进化领域的一种最先进算法。

运行时间的实证分析为了评估 NEVO-GSPT 的计算效率，我们在英特尔酷睿 i5-1235U（1.30 GHz，16GB 随机存取存储器）上进行了 30 次独立运行，并将其运行时间与标准反向传播进行了比较。在 100 代/轮次内，NEVO-GSPT 的平均耗时为 14.58s± 0.60s，而反向传播耗时为 5.12s ± 0.42s。不过，这项比较需要结合重要背景来理解：NEVO-GSPT 的每一代都会评估种群中的 100 个个体，而反向传播则代表一次单模型训练运行。考察单个个体的评估时间时，NEVO-GSPT 的膨胀/收缩操作耗时为 0.0006s ± 0.00008s，相比之下，反向传播耗时为 0.0031s ± 0.0006s，快约 5× 倍

个体评估。值得注意的是，直接将运行时间与 GPU 加速实现（如 PyTorch Genesis（SLM 所用）和 TensorNEAT）进行比较可能并不完全公平，因为我们的 CPU 实现受到不同计算条件的限制。尽管如此，NEVO-GSPT 能够在 CPU 硬件上约 15 秒内同时探索 100 种架构方案，这对于神经进化而言是一项进步，尤其适用于 GPU 访问可能受限的环境。

6 结论

本文介绍了 NEVO-GSPT，这是一种新颖的神经进化算法，旨在应对神经架构搜索中的两项根本挑战：探索架构空间所需的计算开销，以及结构变化与其对网络行为的影响之间缺乏明确映射。我们工作的一个关键创新是将 DGSM 引入神经进化。该新颖算子能够通过移除先前添加的组件来有控制地降低网络复杂度，从而解决影响几何语义方法的膨胀问题，同时保持有利于高效搜索的适应度景观特性。NEVO-GSPT 的计算效率是另一项重要贡献。通过将网络表示为由扰动组件构成的链表，并且仅评估新添加或移除的组件，我们的方法降低了适应度评估的成本，相较于必须从头训练整个网络的方法尤为如此。这种效率使得以分钟而非 GPU 日为单位的计算成本演化大型种群成为可能。

在四个回归基准测试上的实验结果表明，与标准神经网络、SLM、TensorNEAT 和 SLIM-GSGP 相比，NEVO-GSPT 始终名列前茅。在取得这一竞争力的同时，它生成的网络比 SLM 小得多，并且在若干情况下也比 TensorNEAT 更紧凑。

致谢

本研究得到国家资金的支持，该资金通过 FCT（科学技术基金会）提供，用于项目 - UID/04152/2025 - 信息管理研究中心（MagIC）/NOVA IMS -（2025-01-01/2028-12-31）以及 UID/PRR/04152/2025[6] （2025-01-01/2026-06-30）。

[5] https://doi.org/10 2025 6 54499 04152 ./UID//
https://doi.org/10.54499/UID/PRR/04152/2025

参考文献。

1 Archetti, F. , Lanzeni, S., Messina, E., Vanneschi, L.：用于药物发现与开发中计算药代动力学的遗传编程。《遗传编程与可进化机器》**8**(4)，413–432（2007） 2. Assunc˜ao, F., Lourenco, N., Machado, P., Ribeiro, B.：DENSER：深度进化网络结构化表示。《遗传编程与可进化机器》**20**(1)，5–35（2019） 3. Bonin, L., Rovito, L., De Lorenzo, A., Manzoni, L.：细胞几何语义遗传编程。《遗传编程与可进化机器》**25**(1)，8（2024） 4. Castelli, M., Manzoni, L., Goncalves, I., Vanneschi, L., Trujillo, L., Silva, S.：几何语义交叉分析：一种计算几何方法。载于：第八届国际进化计算理论与应用会议。第201–208页（2016年1月） 5. Castelli, M., Trujillo, L., Vanneschi, L., Silva, S., Z-Flores, E., Legrand, P.：结合局部搜索的几何语义遗传编程。载于：2015年遗传与进化计算年会论文集。第999–1006页（2015） 6. Castelli, M., Vanneschi, L., Silva, S.：使用几何语义遗传算子的遗传编程预测高性能混凝土强度。《专家系统与应用》**40**(17)，6856–6862（2013） 7. Custode, L.L., Iacca, G., De Falco, I., Scafuri, U., Della Cioppa, A.：无模型通信的联邦神经进化。《ACM进化学习与优化汇刊》（2025）。https://doi.org/10 3745032 8 1145 3745032, 10 1145 . / https://doi.org/ . / . Dong, X., Liu, Y., Dai, J.：基于全连接神经网络的PyTorch在混凝土抗压强度预测中的应用。《土木工程进展》**2024**(1)（2024年1月）。https://doi.org/https://doi.org/10.1155/2024/8048645，8048645 9 10 1155 2024 https://onlinelibrary.wiley.com/doi/ . / / . Elsken, T., Metzen, J.H., Hutter, F.：神经架构搜索：综述。《机器学习研究杂志》**20**(1)，1997–2017（2019年1月） 10. Farinati, D., Pietropolli, G., Vanneschi, L.：收缩几何语义变异的动态性与有效性研究。《电气电子工程师学会进化计算汇刊》第1–页，3611226 11 1 2025 10 1109 2025–（）。https://doi.org/ . /TEVC. . . Ghosh, A., Jana, N.D., Mallik, S., Zhao, Z.：使用差分进化算法设计最优卷积神经网络架构。《Patterns》**3**(9)，100567（2022）。https://doi.org/https://doi.org/10.1016/j.patter.2022.100567，https://www.sciencedirect.com/science/article/pii/S2666389922001787 12. Goncalves, I., Silva, S., Fonseca, C.M.：语义学习机：一种受几何语义遗传编程启发的前馈神经网络构建算法。载于：葡萄牙人工智能会议。第280–285页。施普林格（2015） 13. Goncalves, I., Silva, S., Fonseca, C.M., Castelli, M.：不确定何时停止？载于：遗传与进化计算会议论文集。ACM（2017年7月 3071328 14 10 1145 3071178）。https://doi.org/ . / . . Huang, H., Peng, J., Li, K., Zhou, J., Sun, Z.：采用解析梯度进行高效GPU加速的神经进化势训练（2025年7月）。https://doi.org/10.48550/arXiv.2507.00528 15. McDermott, J., White, D.R., Luke, S., Manzoni, L., Castelli, M., Vanneschi, L., Ja´skowski, W., Krawiec, K., Harper, R., Jong, K.D., O'Reilly, U.M.：遗传编程需要更好的基准测试。载于：第14届年会论文集

遗传与进化计算会议。第791–798页。GECCO '12，ACM，纽约，纽约州，美国（2012 2330273 16 10 1145 2330163）。https://doi.org/ . / . . Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Hodjat, B.：在神经网络与脑计算时代于人工智能中演化深度神经网络（第293–312页）（2018） 17. Moraglio, A., Krawiec, K., Johnson, C.G.：几何语义遗传编程。载于：et al., C.A.C.（编）《自然界并行问题求解》。计算机科学讲义，第7491卷，第21–31页。施普林格（2012） 18. Nguyen, Q.U.：考察遗传编程中算子的语义多样性与语义局部性。博士论文，都柏林大学，爱尔兰（2011年7月18日），http://ncra.ucd.ie/papers/Thesis_Uy_Corrected.pdf 19. Pietropolli, G., Camerota Verd`u, F.J., Manzoni, L., Castelli, M.：通过梯度下降对遗传编程树进行参数化，以提升符号回归性能。载于：遗传与进化计算伴随会议论文集。第619–622页（2023） 20. Pietropolli, G., Farinati, D., Manzoni, L., Castelli, M., Silva, S., Vanneschi, L.：在SLIM-GSGP中引入交叉。载于：欧洲遗传编程会议（EvoStar的一部分）。第103–119页。施普林格（2025） 21. Pietropolli, G., Manzoni, L., Paoletti, A., Castelli, M.：几何语义遗传编程与基于梯度的优化器的混合。《遗传编程与可进化机器》**24**(2)，16（2023） 22. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.：用于图像分类器架构搜索的正则化进化。载于：AAAI人工智能会议论文集。第33卷，第4780–4789页（2019） 23. Risi, S., Stanley, K.O.：演化可塑性与神经几何的统一方法。载于：2012年国际神经网络联合会议（IJCNN）。第1–8页。电气电子工程师学会（2012） 24. Risi, S., Tang, Y., Ha, D., Miikkulainen, R.：《神经进化：利用人工智能智能体设计中的创造力》。麻省理工学院出版社，剑桥，马萨诸塞州（2025），http://www.cs.utexas. edu/users/ai-lab?risi:book25 25. Rosenfeld, L., Farinati, D., Rasteiro, D., Pietropolli, G., Rebuli, K.B., Silva, S., Vanneschi, L.：SLIM GSGP：用于非膨胀GSGP的Python库。载于：遗传与进化计算会议（GECCO '25）。第9页。ACM，马拉加，西班牙（2025年7月）。https://doi.org/10.1145/3712256.3726398，https://doi.org/10.1145/3712256.3726398-26. Santos, F.J., Gon.calves, I., Castelli, M.：采用盒式变异的神经进化：用于演化深度神经网络的自适应模块化框架。《应用软计算》**147**，110767（2023） 27. Silva, S., Vanneschi, L.：算子均衡、膨胀与过拟合：人体口服生物利用度预测研究。载于：Rothlauf, F.（编）《遗传与进化计算会议，GECCO 2009》。第1115–1122页。ACM（2009） 28. Stanley, K.O., D'Ambrosio, D.B., Gauci, J.：用于演化大规模神经网络的基于超立方体的编码。《人工生命》**15**(2)，185–212（2009） 29. Stanley, K.O., Miikkulainen, R.：通过拓扑增广演化神经网络。《进化计算》**10**(2)，99–127（2002） 30. Suganuma, M., Kobayashi, M., Shirakawa, S., Nagao, T.：使用笛卡尔遗传编程演化深度卷积神经网络。《进化计算》**28**(1)，141–163（2020） 31. Suganuma, M., Shirakawa, S., Nagao, T.：设计卷积神经网络架构的遗传编程方法。载于：遗传与进化计算会议论文集。第497–504页（2017）

32. Vanneschi, L.：SLIM GSGP：非膨胀几何语义遗传编程。载于：Giacobini, M., Xue, B., Manzoni, L.（编）《遗传编程》。第 125–141 页。Springer Nature Switzerland，沙姆（2024） 33. Vanneschi, L., Farinati, D., Rasteiro, D., Rosenfeld, L., Pietropolli, G., Silva, S.：探索非膨胀几何语义遗传编程，第 237–258 页。Springer Nature Singapore，新加坡（2025）。https://doi.org/10.1007/ 978 981 96 0077 9_12 10 1007 978 981 96 0077 9_12 - - - - , 34 https://doi.org/ . / - - - - 。Wang, L., Zhao, M., Liu, E., Sun, K., Cheng, R.：用于 GPU 加速的张量化拓扑增强神经进化。载于：《遗传与进化计算会议论文集》。第 1156–1164 页（2024） 35. White, C., Safari, M., Sukthanker, R., Ru, B., Elsken, T., Zela, A., Dey, D., Hutter, F.：神经架构搜索：来自 1000 篇论文的见解。arXiv 预印本 arXiv:2301.08727（2023）