神经符号集成与可进化策略

Marios Thoma[1,2] 0000-0001-7364-5799, **VassilisVassiliades**[1] 0000-0002-1336-5629和 **Loizos Michael**[2,1]

arXiv:2601.04799v1 [cs.LG] 8 Jan 2026

摘要
神经符号人工智能（NeSy）已成为一种前景广阔的方法，可将神经网络的学习能力与符号系统可解释的推理能力结合起来。然而，现有的神经符号框架通常要求使用预定义的符号策略或可微策略，这限制了它们在缺乏领域专业知识或策略本质上不可微时的适用性。我们提出了一种框架，通过进化过程同时学习不可微的符号策略和神经网络权重，从而解决这一局限。我们将神经符号系统视为种群中的有机体，通过突变（包括符号规则添加和神经权重变化）不断进化，并以基于适应度的选择引导其向隐藏的目标策略收敛。该框架扩展了 NEUROLOG 架构，使符号策略可训练；将 Valiant 的可进化性框架调整至神经符号上下文；并采用 Machine Coaching 语义处理可变的符号表示。神经网络通过从符号组件进行溯因推理来训练，无需满足可微性要求。通过广泛的实验，我们证明，从空策略和随机神经权重开始的神经符号系统能够成功逼近隐藏的不可微目标策略，正确性能中位数接近 100%。这项工作推动了神经符号研究的发展，使其能够应用于难以或无法从专家处获取符号知识的领域。

关键词
Ne神经符号集成、进化学习、溯因推理、符号策略归纳 n.

1 引言

近年来，人工智能已迅速融入日常生活，计算机视觉、个性化推荐系统、人脸识别和大型语言模型等技术日益普及。然而，尽管深度学习和神经网络具有诸多优势，其“黑箱”特性（Lianget al. 2021）给开发可信且可解释的人工智能系统带来了重大挑战。

神经符号人工智能（NeSy）（Besoldet al.2021；Hitzlerand Sarker2021）通过将神经网络的学习能力与符号人工智能固有的可解释性相结合，应对这一挑战。其目标是开发能够进行基于感知的学习和逻辑推理的稳健人工智能系统，从而提升其可解释性、泛化能力和透明度（d’Avila GarcezandLamb2023）。此类集成的一个近期典型案例是 DeepMind 的 AlphaGeometry（Trinh etal. 2024），它将大型语言模型与符号推理引擎相结合，以解决国际数学奥林匹克竞赛中的几何问题，取得了接近人类金牌得主的成绩。

神经符号人工智能面临的一项核心挑战，是如何获取符号策略[1]，尤其是在缺乏领域专业知识或背景知识，或难以获取这些知识的情况下。现有方法通过各种策略应对这一挑战：有些方法通过基于梯度的优化，学习符号策略的可微近似（例如，Dai 等人，2019；Serafini 和 d’Avila Garcez，2016年），另一些方法则假设存在固定的符号结构，以此引导神经学习，同时不施加可微性约束（例如，Tsamoura 等人，2021年）。然而，一个关键缺口仍然存在：如何在不依赖预定义符号策略或可微性假设的情况下，同时学习不可微的符号策略和神经组件。

在本研究中，我们提出了一种框架，旨在弥补这一空白：使神经符号系统能够通过进化过程，同时学习符号策略并优化神经权重。我们的核心贡献在于证明，不可微的符号策略可以从零开始学习（从空策略起步），同时训练神经网络，且无需预定义知识或基于梯度的策略学习。基于 Valiant 可进化性框架（Valiant，2009年）以及将可指导策略与神经架构相结合的广泛研究（Michael 2023），我们将神经符号系统视为进化种群中的有机体；变异会引入新的符号规则并调整神经权重，而基于适应度的选择则引导系统逐步逼近隐藏的目标策略。

通过对 30 个随机生成的目标策略进行 150 次运行的大规模实验，本研究表明，从空的符号策略和随机初始化的神经权重开始的神经符号系统，能够成功近似隐藏的不可微目标策略。该

[1] CYENS 卓越中心，尼科西亚，塞浦路斯[2] 塞浦路斯开放大学，尼科西亚，塞浦路斯

通讯作者： Marios Thoma，Dimarchou Lellou Demetriadi 1，尼科西亚，1016，塞浦路斯。电子邮件：m.thoma@cyens.org.cy

进化方法始终能够实现接近 100% 正确表现的最终中位数正确表现，证明了其在不具备预定义知识或可微性假设的环境中同时开展神经学习与符号学习的可行性。

尽管该方法取得了较高的正确率，但与端到端神经基线相比，其计算成本也较高。“可解释性代价”这一说法已用于量化机器学习中的各种权衡，包括预测或奖励差距（Bertsimas 等，2019；Garcia 等，2024年）、可解释性约束下聚类目标的退化（Dasgupta 等，2020；Laber 和 Murtinho，2021年），以及监管约束下金融回报的差异（Dessain 等，2023）。在本文中，我们将这种代价具体定义为：与能够更高效地实现相近预测性能的端到端神经基线模型相比，为获得显式、易于人类理解的符号策略而需要投入的额外计算资源和更长训练时间，详见第4.6节。

本文其余部分安排如下。首先，我们将在第2 节和第3节介绍必要的背景信息并详细阐述所提框架。我们将在第4节呈现实证评估与研究发现，并在第5节对其进行分析。最后，我们将在第6节回顾相关文献，并在第7节总结全文并提出未来研究方向。

2 背景与预备知识

我们提出的框架借鉴了文献中的三个关键概念，以实现神经组件和符号组件的并行学习：(i) NeSy 框架 NeuroLog (Tsamoura 等人，2021年)、(ii) Valiant 可进化性框架（Valiant，2009年），以及 (iii) Machine Coaching（Michael 2019）。下面简要讨论这些概念，以及使它们能够结合使用的贡献（更多详情请参阅原始论文）。

2.1 Machine Coaching

原始的 NeuroLog 框架使用 Prolog 语言来表示符号策略（见第 2.3节），这些策略本身并不具备可变性或详述容忍性（McCarthy1998），因此给我们提出的框架中所采用的进化学习机制（见第 2.2节）带来了整合方面的挑战。为克服这一限制，我们采用 Machine Coaching（MC）的语义（

Michael2019），这是一种基于论证的学习框架，它将符号知识组织成由优先级规则集构成的策略，使其具有内在的可修改性和可扩展性；这些特性使其能够与可进化系统中的神经架构集成（Michael2023）。

具体而言，遵循 MC 的语义，论域中的二元概念使用命题逻辑表示，形式为原子。文字是原子的实例，可以为正（例如“a”）或负（例如“-a”）（MC 采用开放世界假设，即感知中缺少某个原子并不意味着该原子的否定）。进而，规则是一种逻辑结构，由一系列文字组成其

```
@KnowledgeBase
R1 :: -p1, -p2, -p3 implies h;R2 :: -p1, -p2,
 -p3, p4 implies -h;R3 :: -p5,
 -p6 implies -h;R4 :: -p5, -p6, p7 implies h;
R5 :: -p5, -p6, p7, p8 implies -h;
```

图1. 采用机器指导语义的符号策略结构示例。该策略由按优先级排列的规则构成（从上到下优先级递增），每条规则都包含一个体（文字序列）和一个头（单个文字）。此示例展示了我们的进化框架所学习并近似的不可微浅层命题策略类型。本文将通过此策略展示该框架的运行方式。

体，并由单个文字构成其头。如果给定由文字构成的上下文，规则体中的所有文字都被证实为真，那么规则头也可推断为真。例如，规则“a, b implies c”在上下文 {a，b} 下，可得出“c”成立。

一项策略将一组规则组织成按优先级排列的序列，后面的规则优先级更高。给定一个由文字构成的上下文，策略的决策由上下文触发的最高优先级规则决定。这一决策过程可能产生正确或错误预测，也可能弃权，即上下文未触发任何规则时不作判断。这种策略结构的示例见图1。

MC 允许将新规则添加到策略末尾；这些规则会自动获得高于现有规则的优先级，并可覆盖现有规则。这一特性使规则添加适合作为我们进化框架中的符号变异。

尽管策略中的规则可以形成链，其中一条规则的结论可作为后续规则的输入，但本研究将范围限定在浅层命题策略，即不含链式规则的策略，其中所有规则的头部均为特殊原子 head（或 -head）。

2.2 可进化性框架

Valiant 的可进化性框架（Valiant，2009年）被证明是可能近似正确（PAC）可学习性的一个受限情形（Valiant，1984年），它将进化形式化为一种旨在近似隐藏目标函数的学习机制。具体而言，它将函数视为种群中的个体，并通过变异和适者生存使其不断进化，逐步近似目标函数。将这一框架应用于 NeSy 上下文时，我们将神经符号系统视为***NeSy*** 生物体。

在原始框架中，种群中的每个有机体都是由上一世代的亲代有机体变异而来的后代。为了适应 NeSy 生物体的双重特性，我们为神经和符号组件分别实现了变异机制：(i)神经变异，即后代继承其亲代的神经网络权重，或从随机初始化的权重开始；(ii) 符号变异，即在 $parent$ 现有策略的基础上，通过添加新规则来扩展后代的符号策略，从而逐步演化有机体的符号推理能力。

演化分多个世代进行。每个世代开始时，$parent$ 有机体会繁殖，产生一个后代种群（$parent$ 本身不属于该世代）；由于发生变异，后代种群的规模会有所变化。我们使用基于隐藏目标函数创建的数据集评估种群中每个有机体的适应度。我们采用相同的机制，唯一的区别是，NeSy 生物体会先使用依据隐藏目标符号策略标记的训练数据集进行训练，然后使用验证数据集评估适应度。我们采用相对适应度指标，将每个后代在验证数据集上的性能与其 $parent$的性能进行比较。

对 $fittest$ 有机体的选择遵循原始框架的机制：后代根据其相对适应度与固定阈值参数 t的关系进行分组。当有机体 o_i 的相对适应度f_i分别落在（$-\infty$，$-t$）、$[-t,+t]$或（$+t$，$+\infty$）时，将其归类为有害、中性或有益。如果有有益后代可用，则使用某种适应度比例选择方式选出一个有益后代 o_i，其概率为 $f_i^k/\sum_j f_j^k$，其中指数 k 是固定的非线性参数。如果没有有益后代可用，则从中性后代 o_i中均匀随机地选出一个。最后，如果另外两组都为空[2]，则从有害组中选出 o_i 最高的后代 f_i；若有多个并列，则随机打破平局。选中的有机体会加入 $fittest$个体的谱系，并成为下一世代的创始个体。

此前，Markos 等人（2022年）展示了该进化框架在结合 Machine Coaching 语义学习符号策略方面的有效性。在此基础上，我们的工作通过引入下文所述的 NEUROLOG 架构，将这一方法扩展到神经符号环境，从而能够同时进化符号策略和神经网络权重。

2.3 NEUROLOG *NeSy* 神经符号框架

NEUROLOG 框架（Tsamoura 等人，2021年）通过将神经系统和符号系统视为纳入统一架构的独立“黑箱”，以组合方式将二者整合起来。这些黑箱作为独立的模块嵌入统一架构中。该架构由一个符号模块（包含一个符号策略）、一个神经模块（包含一个神经网络）以及一个预定义的转换器 函数组成，该函数使模块之间能够通信（图2）。该框架的组合特性和语义无关性，使我们能够轻松地将其调整为所提出的进化方法。

NEUROLOG 框架假设符号策略是给定的，但不对其可微性做任何假设。它通过溯因推理（Kakas 2017年）绕过可微性要求。溯因推理是一种反向推理形式，能够使用符号模块的符号策略训练神经模块的神经网络。策略 p 由一组原子 A（二元／布尔概念）上的逻辑表达式组成。该策略同时支持正向推理和反向推理：给定输入原子 $I \subseteq A$，正向推理会生成输出原子 $O = \text{deduce}(p, I)$，使得 $p \cup I \models O$。反之，给定输出原子 O，反向推理会生成一个或多个输入原子集合 $I \in \text{abduce}(p, O)$，使得 $p \cup I \models O$。任意合取的

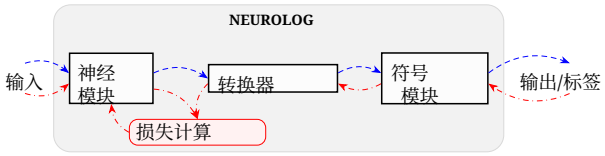


图 2. NEUROLOG NeSy 架构，由 Tsamoura 等人 (2021年)。

来自 abduce(p, O) 的原子构成给定结果（或标签）O的溯因证明。

原子 A 同时对应于 神经模块的输出神经元和 符号模块的二元概念（详见第 2.1节）。在训练过程中，神经模块 通过来自符号模块 的溯因反馈，借助语义损失计算（Xu 等（2018））进行学习。给定一个包含神经输入及其标签的训练实例，系统首先为该标签生成所有可能的溯因证明。这些证明被编译成句子决策图（SDD）（Darwiche（2011）），从而能够高效地进行加权模型计数（WMC）计算（Chavira 和 Darwiche（2008））。WMC 融合了 神经模块处理神经输入时产生的神经元激活值，并为溯因证明中的每个原子赋予权重。随后，将语义损失计算为 WMC 的负对数。

3 所提框架

综合上一节讨论的思想，我们所提出的框架由两个主要部分组成：经过大幅修改和扩展的 NEUROLOG 架构，以及一种进化算法。二者结合后，可以对神经模块和神经符号系统的符号模块进行并行训练。

3.1 扩展 NEUROLOG架构

我们进一步形式化 NEUROLOG 框架，为符号模块引入一种学习机制，使其能够在进化环境中进行训练。

原始框架采用演绎（正向推理）和溯因（反向推理）的概念，从而实现神经模块与符号模块之间的透明集成。我们将其扩展，纳入归纳（学习）的概念，即让两个模块都能够接受训练。神经模块原本就可以训练，但符号模块不行。为此，我们要求两个模块都提供三种方法：deduce()、abduce() 和 induce()，分别对应上述三种概念[3]。

系统中的正向推理通过串联两个模块的 演绎方法实现，并使用 转换器 促进它们之间的通信，如图 3 所示（黑色箭头）。稍后图4展示了一个具体示例，其中使用实际的 MNIST 输入和符号策略进行正向推理。

对于符号模块，induce() 方法提供了一种修改模块中符号策略的方式，从而获取新知识。尽管我们的框架不对所采用的符号策略语义作任何假设，但在本研究中，我们采用 MC语义，

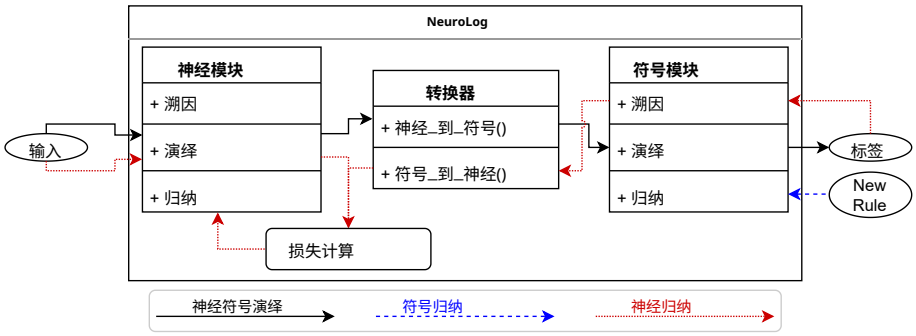


图 3. 扩展的 NEUROLOG 架构，支持神经学习与符号学习并行进行。黑色箭头表示正向推理（演绎）：NeuralModule 和 SymbolicModule 的演绎方法通过转换器 串联，以生成预测。红色箭头表示神经归纳：`NeuralModule` 利用 `SymbolicModule`的溯因方法所提供的溯因反馈，通过反向传播进行训练；该方法根据符号规则生成训练信号。蓝色箭头表示符号归纳：在进化突变过程中，通过`SymbolicModule`的归纳方法将新规则添加到其策略中。

如第 2.1 节所述。因此，符号模块的归纳 方法对应于向符号策略中添加新规则（图 3，蓝线）。我们采用推理引擎 *Prudens*(Markos和Michael，2022年)，该引擎实现了 MC语义，用于解释符号策略。*Prudens*能够通过此类策略进行演绎和溯因，因此为符号模块的演绎() 和溯因() 方法提供支持。符号模块。由于溯因过程计算开销较大，所提框架对溯因结果进行了大量缓存。

在`NeuralModule`中，归纳方法直接对应于在神经网络中使用反向传播；训练损失则是在利用来自 SymbolicModule 溯因方法的溯因反馈计算得出后（图 3，红线），才进行反向传播。与经典神经网络训练类似，系统训练会进行多个训练轮次，每个训练轮次都使用完整的训练数据集。

如前所述，扩展的 NEUROLOG 架构的实例被视为进化环境中的个体。尽管个体的`NeuralModule`主要通过来自`SymbolicModule`的溯因反馈进行训练（如第2.3节所述），但当个体仍在学习其符号策略时，这种溯因反馈并不完整（尤其是在进化过程的早期）。为探究额外的训练信号能否改善早期世代的收敛情况，我们在 NeuralModule 中引入了自监督学习机制，采用重构损失的形式。该辅助损失项提供的训练信号不受符号策略完备性的影响，在溯因反馈有限时可能有助于引导神经学习。不同损失配置的实现细节及消融研究结果见第4.2节和第4.5节。

3.2 进化过程

进化过程的目标是近似一个包含隐藏目标策略 p 的神经符号系统，从一个策略为空的神经符号系统开始，同时优化其神经权重。为此，使用目标策略 p 构建示例集 $\widehat{p}$ ，然后将其随机划分为训练 ($\widehat{p}_{train}$)、验证 ($\widehat{p}_{val}$) 和测试 ($\widehat{p}_{test}$) 子集。 $\widehat{p}_{train}$ 和 $\widehat{p}_{val}$ 用于进化过程，而 $\widehat{p}_{test}$ 则被保留，用于在进化过程结束后对其进行客观评估。

算法 1 概述了进化过程，其中使用我们改进版 NEUROLOG 架构的实例作为 NeSy 生物体。在 $gen = 0$ 时，数组 ℓ（用于存储 $fittest$ 个体的谱系）以一个单一个体作为初始成员，该个体的符号策略为空，神经权重则随机初始化。此后，每一代都从上一代的$fittest$ 个体开始繁殖，生成一个由突变后代组成的种群（pop_{gen}）。

具体而言，对于符号变异，所有后代都会继承其 $parent$的符号策略，并按以下方式进行修改：(i) S_0（克隆 变异）：后代继承其 $parent$符号策略的精确副本；(ii) S_+（加法变异）：基于从 2^A 中抽取的随机上下文 x（因为每个原子都可以为正或为负），通过向策略中添加新规则“x 推出 head”或“x推出 -head”，创建两个后代；(iii) $S_\downarrow$（简化变异）：针对每个 j创建一个后代，方法是向其策略添加规则“body$_{-j}$ 推出 head”；其中，“body 推出 head”是最近添加到其 $parent$的策略中的规则，而 body$_{-j}$ 是从 body 中减去其第 j 个文字。类似地，我们将神经权重 w 是否从亲代遗传给后代视为神经变异，具体包括：(i) N_{pw}：后代从亲代继承 w；(ii) N_{rw}：后代以随机初始化的 w开始。创建足够多的后代，以涵盖符号变异与神经变异的所有可能组合（这两类变异相互独立，因此可以进行所有组合；例如， $N_{\text{pw}}S_0$ 后代是符号克隆且继承了神经权重，而 $N_{\text{rw}}S_+$后代则是加法符号变异且神经权重随机初始化）。

随后， pop_{gen} 中的所有个体都使用$\widehat{p}_{train}$ 训练若干个训练轮次。然后，计算它们的相对适应度，用来衡量每个后代相对于其亲本的进展；具体而言，通过逐项比较后代与亲本在$\widehat{p}_{val}$上的演绎结果变化来计算，并根据变化方向给予奖励或惩罚，所用评分矩阵见表2 （附录 A）。随后，根据参数 t 和 k，使用 pop_{gen}中的 $fittest$ 机制，选出一个个体

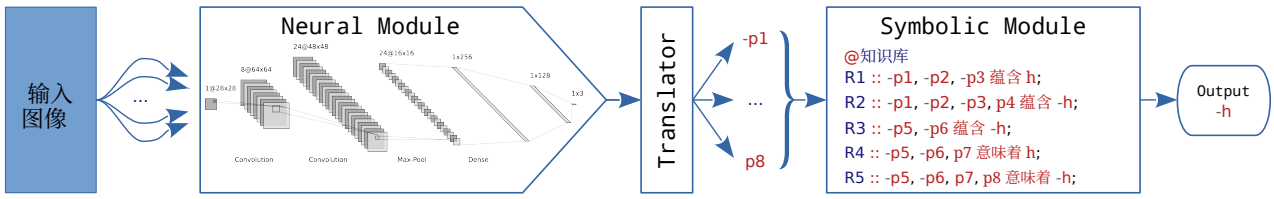


图4. 扩展的神经符号架构中使用具体目标策略进行正向推理的示例。输入的 MNIST 图像序列由 神经模块 (卷积神经网络)处理，并通过转换器生成符号原子预测；符号模块 使用这些预测及其策略（与图 1所示的策略相同）生成最终输出。该策略是实验中使用的随机生成目标策略之一。

算法1：进化过程算法。输入：数据集 $\widehat{p}_{train}$、$\widehat{p}_{val}$ 和 $\widehat{p}_{test}$。参数 t、k 和 $maxgen$。1 在 $gen = 0$处，初始化数组 ℓ 以保存适应度最高的生物体的谱系，其中初始包含一个具有空策略且神经网络权重随机初始化的有机体；2 对于 $gen = 1$至 $maxgen$，执行3 $parent = \ell[gen-1]$; /* 上一代中适应度最高的亲本有机体 */ 4 将 $parent$的所有后代（由神经突变（N_{pw}、N_{rw}）和符号突变（S_0、S_+、$S_\downarrow$）的所有组合产生）填入数组 pop_{gen}；5 使用 $\widehat{p}_{train}$训练 pop_{gen} 中的所有个体；6 使用 $\widehat{p}_{val}$，计算 pop_{gen} 中每个个体 o_i 相对于 $parent$的相对适应度 f_i；7 根据 t，将 pop_{gen} 分为 $beneficial$、$neutral$ 和 $detrimental$ 组；8 如果 $beneficial$ 不为空，则 9 使用指数 k进行适应度比例选择，从 $beneficial$中选择 $fittest$；10 否则，如果 $neutral$ 不为空，则 11 从 $neutral$ 中随机均匀地选择 $fittest$；12 否则 13 选择 $detrimental$中 f_i最高的有机体作为 $fittest$；14 $\ell[gen] = fittest$; 15 清空 pop_{gen}; 16 如果 $f_{fittest} \geq 99\%$则 17 跳出 /* 提前停止 */ 18 使用 $\widehat{p}_{test}$ 测试 ℓ 中的所有个体；

如第 2.2 节所述，并将其加入谱系 ℓ，使其成为下一代的创始个体。进化过程将持续进行，直到满足以下条件之一：(i) 超过指定的生成结果数量（$maxgen$）；或 (ii) 触发提前停止准则（例如，达到在 $\widehat{p}_{val}$ 上正确表现的阈值）。进化过程结束后，将使用 $\widehat{p}_{test}$ 对 ℓ 中的个体进行评估。 ℓ 中的最终个体代表神经符号系统中神经组件与符号组件并行训练的结果，这也是我们所提框架的目标。

3.3 技术贡献

将进化算法用作教练，意味着在连续世代中反复对多个 NeSy“ 生物体 ” 进行变异（指导）和训练，直到出现一个能够成功近似隐藏符号策略的 NeSy 生物体。初步实现表明，这一过程面临显著的计算挑战，原因是存在两个主要瓶颈：实验期间 NeSy 生物体训练的顺序性，以及训练期间损失计算的计算成本。未经优化时，即使是训练单个世代所需的个体种群，速度也远低于实际可行水平，这使得本研究中所述的实验难以大规模开展。以下小节将详细介绍我们为解决这些计算瓶颈而开发的技术贡献。这些优化共同实现了第 4.2 节所述的大规模实验，其有效性将在第 4.4 节中通过实证验证。

3.3.1 单个 *GPU* 上的神经网络并行训练训练神经网络时，一个常见挑战是单个 GPU 的显存容量有限，这往往需要使用显存更大的 GPU，或采用多 GPU 训练策略。相比之下，本研究介绍的 NeSy 进化实验涉及训练大量规模相对较小的神经网络，每个网络都嵌入一个 NeSy 生物体中。由于个体数量众多，顺序训练这些神经网络会带来显著的计算挑战。为此，我们开发了一种并行训练方法，使多个神经网络能够在单个 GPU 上并行训练。并行训练通过 PyTorch 的多进程模块实现。一项重要的工程考量是确保所有进程间对象都可序列化，这是 Python 运行环境的要求，从而使这些对象能够安全地传递给其他进程。

3.3.2 语义损失计算期间的操作缓存NeSy 生物体的训练，尤其是其神经模块，基于语义损失的计算（Xu 等（2018）），这一过程的计算量可能很大。计算量大的原因在于，根据符号组件的溯因反馈推导损失时涉及复杂的步骤。如第 2.3 节所述，对于每个训练实例，符号模块 会生成溯因证明，将其编译为句子决策图（SDD）（Darwiche（2011）），并通过深度优先搜索（DFS）遍历 SDD 来计算加权模型计数（WMC）（Chavira 和 Darwiche（2008）），同时

整合神经激活值。反复生成溯因证明，更重要的是为每批训练数据构建并遍历 SDD，都会带来显著的计算开销。

为缓解这一计算瓶颈，我们的框架实现了一种操作缓存策略。其核心思想是利用训练批次内及批次间计算的重复性。在一个训练批次中，针对给定标签首次遇到一组特定的溯因证明（因而对应特定的 SDD 结构）时，会执行加权模型计数所需的初始深度优先搜索操作。随后，将操作序列及此次遍历的结构缓存为计算图。同一批次中，之后所有需要针对相同 SDD 结构计算加权模型计数的实例（但通常具有不同的神经激活值），都会重复使用这个预编译的计算图。

这种缓存机制大幅缩短了训练时间。其一大优势是显著减少了 CPU 与 GPU 之间的交互。传统上，符号推理步骤（溯因、SDD 构建、深度优先搜索）在 CPU 上执行，因此数据需要在 CPU 与 GPU 之间来回传输，而神经计算和损失反向传播则在 GPU 上进行。通过缓存加权模型计数计算的计算图（该图可由神经网络输出参数化），语义损失计算的核心部分便能直接在 GPU 上反复执行。这减少了 CPU 与 GPU 之间耗费高昂的数据传输和同步点，从而让 GPU 资源的使用更加流畅、高效。

4 实验与结果

本节介绍了我们对所提框架开展实证研究的关键方面。更多细节见附录 A；用于复现所有实验的代码已公开（参见补充材料）。

4.1 目标策略与数据集

实验中使用的目标策略是随机生成的：首先将 A 固定为一组包含 8 个二元概念的集合$\{a1,\ldots,a8\}$，用于上下文和规则体；随后随机生成一组 P目标策略，其使用来自 A 的原子（图1展示了一个目标策略示例）。我们采用了 Markos 等人（2022年）开发的随机策略生成器。需要特别指出的是，由于目标策略具有离散性质，并且其结构由逻辑蕴含定义，因此所使用的目标策略是不可微的，这使得基于梯度的优化方法无法直接应用。
直接应用基于梯度的优化方法。

数据集构建如下：为每个目标策略构建一个示例集 $\widehat{p}$， $p \in P$每个由 8 个原子组成的数据实例（或上下文）均通过从 2^A 中均匀随机采样构建（由于每个原子可以为正、负或未观测状态，但未观测原子不会包含在数据实例中），并根据 p对每个数据实例进行标签标注，然后筛除其中 p弃权的数据实例。随后，通过使用从 MNIST 数据集（Lecun等人

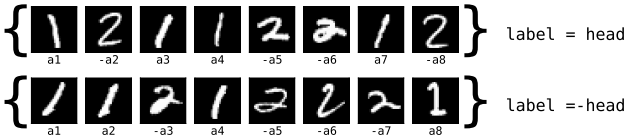


图5。 来自由图1中的符号策略标注的一个 $\widehat{p}_{train}$ 数据集的两个训练实例。每个实例由一串包含 8 张 MNIST 数字图像的序列组成，这些图像表示原子：数值为 1 ( ) 的数字表示正原子，数值为2 ( ) 表示负原子。标签（头或-头）由符号策略的规则应用于原子序列后确定。

1998）中随机选取的手写数字图像来表示各个原子，将每个训练实例转换为图像表示，具体约定如下：数值为 1（ ）的 MNIST 数字用于表示正原子，数值为 2（ ）的数字用于表示负原子。例如，上下文 {a1, ..., -a8} 将表示为 {, . . . , }。图 5 展示了两个完整上下文训练数据实例示例，这些实例使用图1中的策略进行标签标注。 $\widehat{p}_{train}$ 和 $\widehat{p}_{val}$ 使用了来自 MNIST 训练子集的图像，而 $\widehat{p}_{test}$ 使用了来自 MNIST测试子集的图像。

4.2 实验设置

实验中使用了一组 P由随机生成的30个目标策略组成的集合。对于每个策略，使用不同的随机生成数据集重复实验5次（共$30 \times 5 = 150$ 次实验）。数据集的规模如下： $\widehat{p}_{train}$ 20000 个数据实例、 $\widehat{p}_{val}$ 2000和$\widehat{p}_{test}$ 2000。每一世代的种群通过组合所有可能的符号（S_0、 S_+、 $S_\downarrow$）和神经（N_{pw}、 N_{rw}）变异生成。实验表明，每一世代引入多次 S_+ 变异可以显著缩短进化过程，因此每一世代使用了 5 S_+次变异。

在神经模块中使用了卷积神经网络（卷积神经网络）架构，由2个卷积层和其后的3个全连接层组成（图4和13）。由于数据实例由8张MNIST图像组成的序列构成，因此每张图像都依次输入卷积神经网络，并将每张图像对应的输出向量拼接成一个单一输出向量（第4.6节所述的端到端基线也使用了相同的基础编码器，并额外添加了用于直接二元分类的输出层）。后代的神经权重使用Xavier初始化随机初始化（Glorot和Bengio, 2010年）。使用了Adam优化器（Kingma和Ba, 2017年），学习率为0。001。

语义损失的计算遵循第 2.3 节所述的方法，并采用我们在第 3.3.2 节详细介绍的优化技术。加权模型计数（WMC）使用 PySDD 库（wannesm, 2024年）执行。

如第3.1节所述，为了探究在溯因反馈有限时，额外的训练信号是否能够改善收敛，我们加入了重构损失分量。对于该重构损失，我们使用镜像反卷积解码器，以神经网络输出向量为输入来重构输入的 MNIST 图像，并使用均方误差（MSE）进行计算。为了限制解码器仅重构

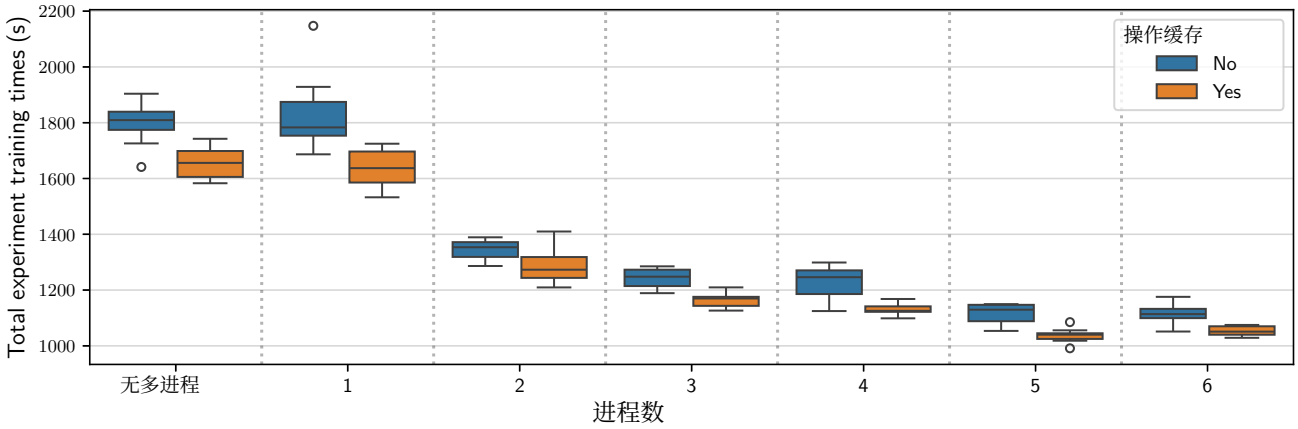


图6. 箱线图展示了操作缓存和并行处理对多神经符号个体训练实验总耗时的影响。每个箱体代表20次实验（10种符号策略，每种运行两次），每次实验均涉及训练100个 NeSy 生物体（在非进化设置下）。横轴表示在单个 GPU 上使用的并行进程数（每个进程对应一个 CPU），纵轴表示实验总训练时间（秒）。蓝色箱体表示在语义损失计算中未使用操作缓存的训练结果，橙色箱体表示启用操作缓存的训练结果。结果表明，在所有并行化配置下，操作缓存都能持续缩短训练时间。“无多进程”条件表示使用单个 CPU 进行顺序训练，且不使用多进程框架；而1个进程表示在多进程环境中使用单个进程。这些配置之间的相似性证实，多进程环境本身带来的开销极小。此外，增加并行进程数可显著缩短实验总耗时，但效率提升在并行进程数约为5时趋于平缓。

MNIST 数字的一个版本（数值 $\{1, 2\}$），在将神经网络输出向量传递给解码器之前，对其应用了 Gumbel-Softmax操作（Jang等人，2017; Maddison等人，2017）。为了评估组合这些损失分量的不同方法，我们进行了消融实验，比较了三种语义损失与重构损失比：1:0（仅语义损失）、1:1（等权重）和 1:10000（高权重重构损失），详情见第4.5节。

对于进化过程，使用上限 $maxgen$ =500，并将提前停止准则设为：当 $\widehat{p}_{val}$ 上的正确表现达到 $\geq 99\%$ 时停止。使用了阈值参数 $t = 0$ 和非线性参数$k = 2$。个体使用 $\widehat{p}_{train}$训练 5 轮，批量大小为 2000。如图 7所示，选择此批量大小是因为它在每轮训练时间和累计训练效率之间实现了最佳的整体平衡：与较小的批次大小相比，它显著提升了性能，同时避免了较大批次大小所带来的边际收益递减。

4.3 主要结果

图 8 展示了一次典型实验运行的示例（更多示例见附录 B）。上方子图展示了进化过程中选出的 $\widehat{p}_{val}$ 和 $\widehat{p}_{test}$ 条件下谱系中 $fittest$ 个体的性能。以此为例，个体的正确表现（蓝线）逐渐提高，而弃权表现（绿线）相应下降。错误性能（红线）在进化初期略有上升，但最终回落至接近零。

图 9展示了所有实验在各自测试子集上的汇总结果 $\widehat{p}_{test}$。顶部子图展示了训练进程中的性能指标：正确表现（蓝色）稳步上升，从接近零增长到训练结束时接近中位数 ~99%，而弃权表现（绿色）则呈现相反趋势，从最初占主导地位逐渐下降至接近零，体现了符号策略的逐步形成。错误性能（红色）始终相对较低，中位数低于 15%，并随着正确表现的提升最终进一步下降。中间和底部子图展示了语义损失和重构损失的相应下降。重构损失的降低反映了神经学习，而语义损失的下降则表明两个组成部分实现了成功的联合优化。

4.4 性能优化验证

为实证验证第3.3节所述技术贡献的有效性，我们开展了一系列实验，测量这些贡献对计算性能的影响。验证实验在各种并行处理和操作缓存配置下，对 100 个 NeSy 生物体进行了训练，共进行 20 次独立运行（10 种符号策略，每种运行两次）。我们记录了每种配置的总训练时间，结果见图6。

语义损失计算过程中的操作缓存，在所有并行化配置下均显著提升了速度。比较图6中的未缓存结果（蓝色箱线图）与缓存结果（橙色箱线图）可见，缓存 WMC 计算图能够稳定且显著地缩短训练时间。这表明，重复进行 WMC 计算构成了一个主要的计算瓶颈，而缓存能够有效解决这一问题。

在单个 GPU 上并行训练多个神经网络可进一步提升性能。结果表明，将并行度从顺序执行提高到

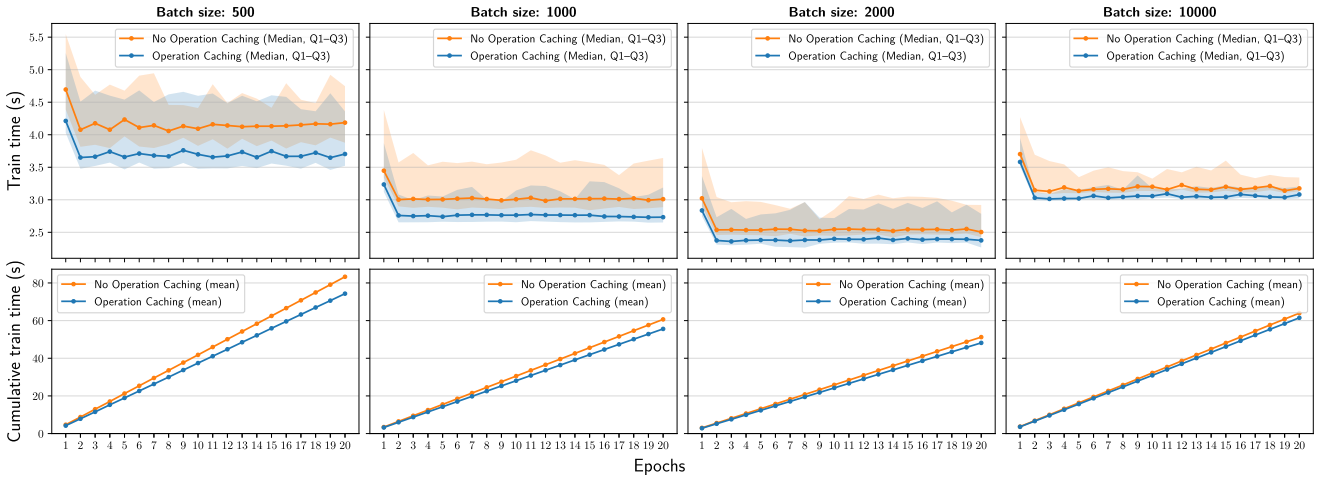


图7. 在语义损失计算过程中，比较不同批量大小（500、1000、2000、10000）下启用和未启用操作缓存时的训练时间。对于每种批量大小，上方子图显示每轮训练时间（实线和阴影区域分别表示中位数和第一四分位数至第三四分位数范围），下方子图显示各训练轮次的累计训练时间（均值）。批量大小2000总体平衡最佳：每轮训练时间最低，同时累计训练时间也较为高效，因此是主要实验的最优选择。

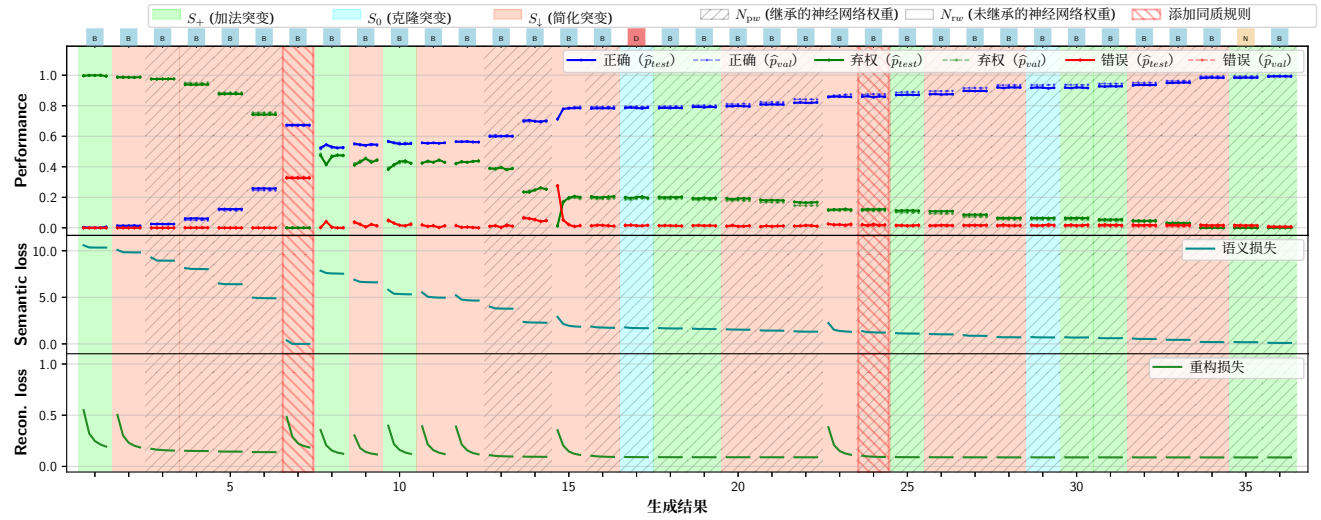


图8. 单次实验运行的结果：各列代表不同世代，展示 $fittest$ 有机体每个世代的性能和训练历史。第1个子图显示 $\hat{p}_{val}$ 和 $\hat{p}_{test}$上的正确、弃权和错误性能。第2和第3个子图分别显示语义损失和重构损失。符号突变和神经变异通过背景颜色和阴影图案加以区分，同质规则添加也以阴影图案标记。顶部横轴上的字母表示该世代的 $fittest$ 有机体是从哪个适应度组（有益组、中性组或有害组）中选出的。由于初始有机体采用空策略，因此未进行训练，故省略世代0。

约 5 个并行进程，可显著缩短实验总时长。超过这一数量后，效率提升趋于平缓，这可能是由于 GPU资源饱和以及管理并发进程所产生的开销。顺序训练（“无多进程”）与单进程执行（1 个进程）的性能相当，这证实多进程框架本身引入的开销极小。

这些验证结果表明，本文所呈现的大规模进化实验要切实可行，必须同时采用这两项优化。对于第4.2节报告的主要实验，我们采用了每块 GPU 配置 4 个并行进程的方案，并根据观察到的性能特征，在计算效率与资源利用率之间取得平衡。

4.5 消融研究：损失比率比较

如第 3.1节所述，引入重构损失，是为了在进化早期代溯因反馈有限时提供额外的训练信号。为评估这一辅助信号是否能改善收敛，消融实验比较了三种不同的语义损失与重构损失比：1:0（仅语义损失）、1:1（等权重）和 1:10000（高权重重构损失）。结果见图10，该图按代数汇总结果（与图9不同，图 9 将结果插值到统一尺度）。

在单项实验中观察到的相同总体趋势（例如，图 8）适用于所有损失比配置：在进化过程中，正确表现逐渐提升，同时

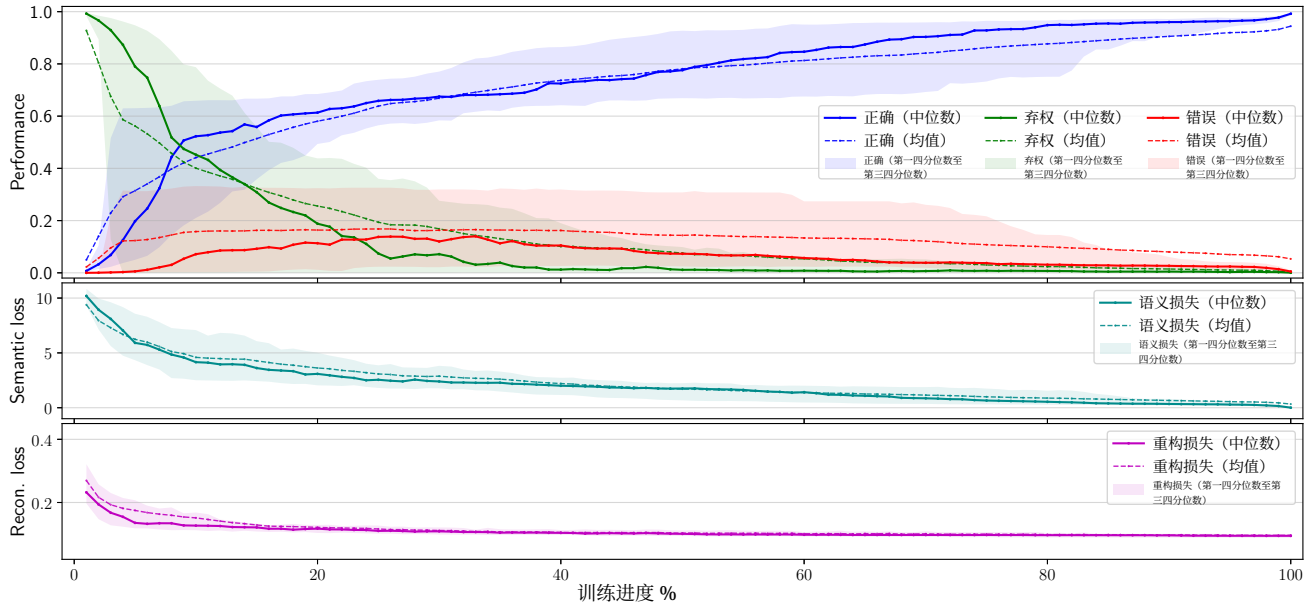


图9. 所有实验在各自 $\widehat{p}_{test}$ 子集上的汇总结果。由于实验结束时的代数各不相同，我们将结果插值到统一的1–100尺度，以便对所有实验运行进行有意义的汇总。阴影区域表示各项指标的四分位距（第一四分位数至第三四分位数）。

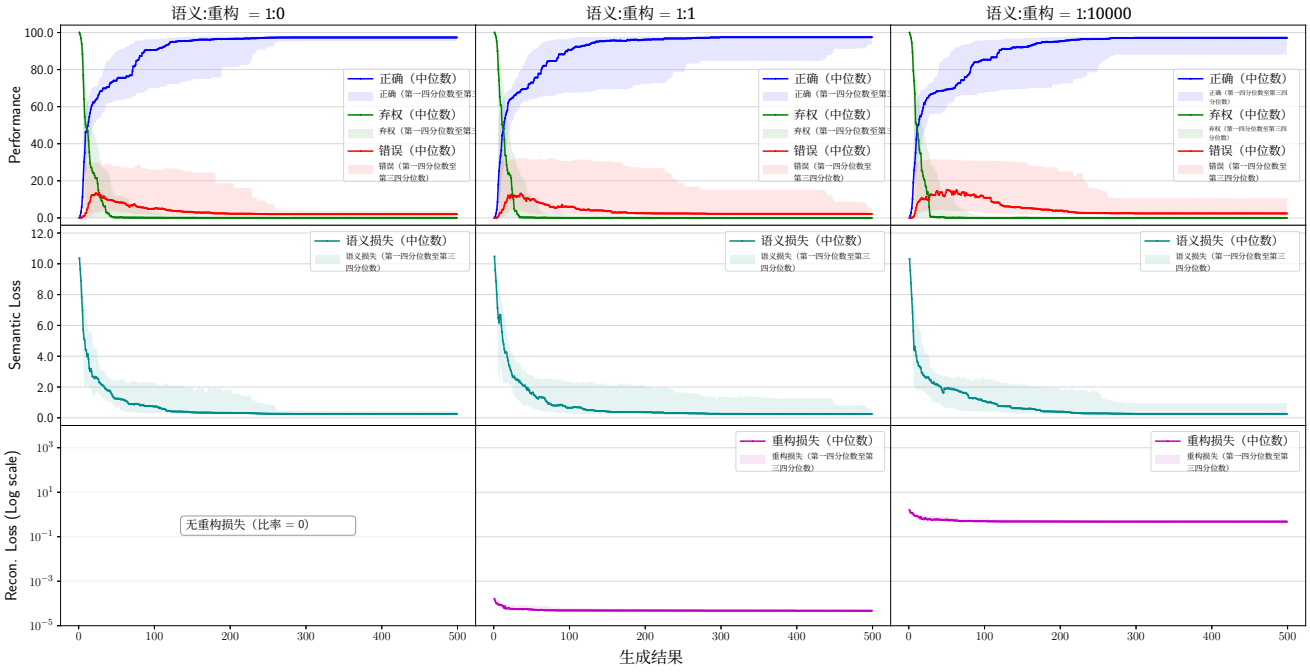


图10. 比较三种语义损失与重构损失比（1:0、1:1和1:10000）在各自 $\widehat{p}_{test}$ 子集上的汇总结果。结果按代数汇总：对于每一代，将所有达到该代的实验数据合并。由于实验在不同代数终止，后续各代汇总的实验数量较少。顶部子图显示性能指标（正确、弃权和错误预测），中部和底部子图分别显示语义损失和重构损失。阴影区域表示各项指标的四分位距（第一四分位数至第三四分位数）。

弃权性能逐渐下降。由于进化算法具有贪婪特性，每一世代都会选出单个有机体作为 $fittest$ ，因此在进化过程的早期，错误性能会有所上升；随后，随着正确表现的提升，错误性能会降至接近零。

为评估不同损失比是否会影响进化效率，我们比较了每次实验运行在所有世代中的累积正确表现（每次实验运行得到一个每次实验的汇总值），按损失比率分组。该累计指标反映了整体演化效率：即使最终性能相近，越快达到高性能的配置所累积的分数也越高。采用Dunn非参数检验进行统计分析后发现，三种损失比率配置之间的差异均无统计学意义（所有两两比较 $p > 0.05$），详见表 1。这表明，尽管重构损失分量的引入有其理论依据，但它并未带来可测量的

表1. 语义损失与重构损失比（1:0、1:1、1:10000）之间累积正确表现比较的 Dunn 检验 p 值。未发现显著差异（所有值 > 0.05）。

比值	0	1	10000
0	1.00	0.78	0.87
1	0.78	1.00	0.91
10000	0.87	0.91	1.00

不完整的溯因反馈，却无法为收敛带来可测量的收益。关于这一发现的影响将在第 5节讨论。

4.6 基线

由于本研究所用策略具有特殊性质，因此无法直接与文献中的其他神经符号框架进行比较。诸如 MetaAbd（Dai 和 Muggleton，2021年）、NSIL（Cunnington 等人，2022年）和 NeuralFastLAS（Charalambous 等人，2023）等框架采用了不同的策略结构、学习范式和知识表示假设，因此与我们用于学习不可微浅层命题策略的进化方法不兼容。我们转而将本方法与端到端神经基线进行比较；具体而言，采用了与神经符号个体中 神经模块 相同的卷积神经网络架构，但直接使用标注数据进行训练，不包含任何符号组件。

端到端神经基线在与 NeSy 实验相同的 150 个数据集上进行了评估（30 种目标策略 $\times$ 5，每种策略各进行随机数据集生成，如第 4.2 节所述），并采用相同的训练／验证／测试集划分。基线使用 Adam优化器进行训练，学习率为 0.001，批次大小与 NeSy 个体相同。基线网络接收由 8 张 MNIST 数字图像组成的序列作为输入，并通过监督学习直接预测二元标签，不进行任何中间符号推理。虽然基础卷积神经网络编码器架构与神经模块 所用架构相同（依次处理 8 张图像，生成 n 维输出向量），但端到端基线增加了用于直接二元分类的层：将拼接后的编码器输出展平（生成一个 8n 维向量），再传入两个全连接层（64 个神经元并采用 ReLU 激活，然后是 2 个输出神经元），最后经过 Softmax，从而能够直接预测 正面 或 -正面，而无需符号推理。图 11 展示了端到端神经基线在 100 个训练轮次中的训练进程。尽管中位数性能达到

$> 98\%$ 的准确率在所有集合上，结果在不同网络之间呈现出显著差异，偶发的收敛失败导致图中可见的四分位距较大。

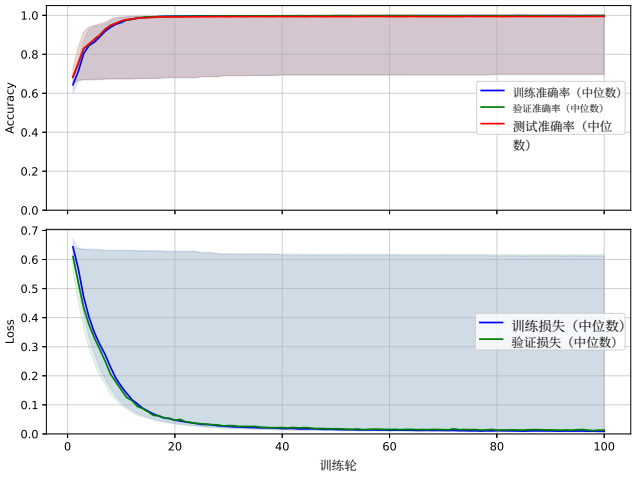


图11。 端到端神经基线在150个数据集上经过100个训练轮次的训练进程（每个目标策略进行30次× 5 随机数据集生成）。该基线采用与神经符号个体的神经模块 相同的卷积神经网络架构，但直接使用标注数据进行训练，不包含符号组件。网络使用交叉熵损失和Adam优化器进行训练。性能指标显示了 $\widehat{p}_{train}$（蓝色）、$\widehat{p}_{val}$（绿色）和 $\widehat{p}_{test}$（红色）上的准确率，中位数准确率为 $> 98\%$。下方子图显示训练损失和验证损失。阴影区域表示每项指标的四分位距（第一四分位数至第三四分位数）。范围的巨大差异表明，尽管大多数网络都能学得很好（中位数准确率接近100%），但有些网络偶尔无法有效收敛，从而导致准确率较低且损失较高。

5 讨论

第 4.3 节所呈现的结果表明，所提出的进化框架能够有效地使神经符号系统在训练神经网络的同时，学习不可微的符号策略。从空的符号策略和随机初始化的神经网络权重开始，进化过程持续发现能够近似隐藏目标的策略；到训练结束时，正确性能中位数接近 99%（图 9）。在各项实验中观察到的典型模式——正确性能稳步提升，而弃权逐渐减少（图8）——表明进化过程逐步构建出能够覆盖输入空间更大部分的符号策略。两种损失同时下降，证实了学习过程的有效性。重构损失的下降表明通过自监督实现了神经学习，而语义损失的下降则反映出联合优化取得了成功，因为该指标同时取决于神经网络输出和符号策略的溯因反馈。

这些发现验证了本研究的核心假设：不可微的符号策略可以通过进化搜索从头学习，无需预先定义的符号知识，也无需基于梯度的策略优化。大多数实验都收敛至近乎完美的正确表现，这一事实证明了该方法用于并发神经符号学习的可行性。

尽管大多数实验都遵循了这种成功的收敛模式，但在某些情况下，进化过程进入了我们所称的 停滞状态，即个体在训练数据和验证数据上的表现部分正确，但在后续世代中未能进一步提升的局部最优。图 12展示了这样一个例子：进化过程在第3代陷入停滞状态。这种停滞状态一直持续

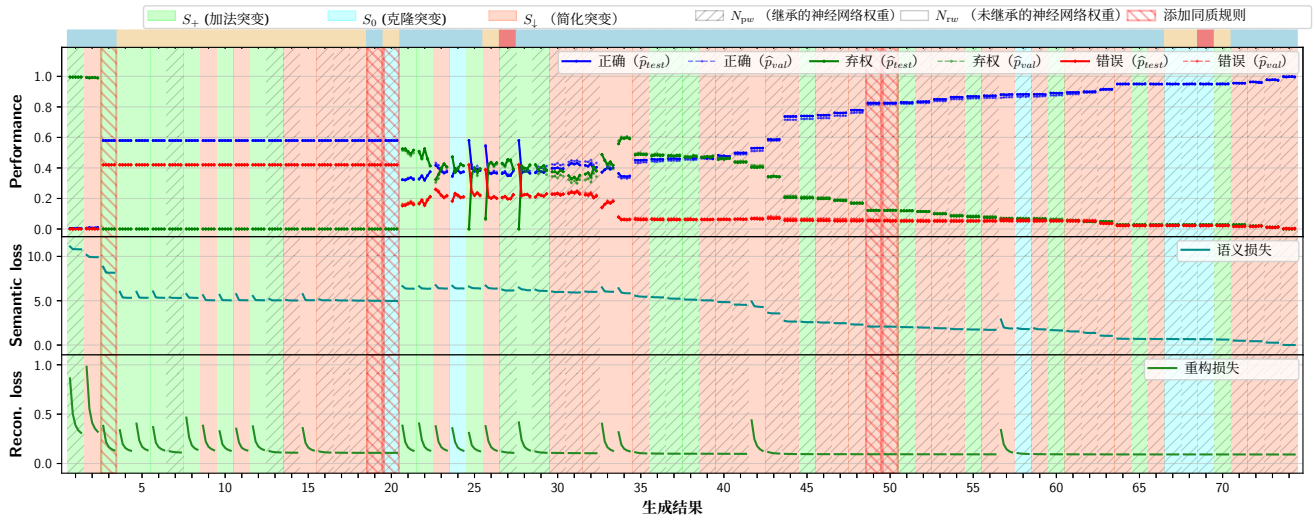


图12。 展示进化过程部分阶段陷入停滞状态的实验运行。

直到第21代才被打破，使进化过程得以依次选出一条个体谱系，并在第74代达到 $> 99\%$ 的正确表现。

停滞状态源于向个体的策略中添加齐次规则——即其体中包含的文字全部具有相同符号的规则——，添加方式为 S_+ 或 $S_\downarrow$突变。同质规则的示例包括“a1，a2 蕴含 head”（全部为正原子）或“-a1，-a2 蕴含 head”（全部为负原子）。当此类规则添加到策略中时，神经模块有时会收敛到一个平凡解，将所有输入图像分类为同一类别（全部分类为数字 1 或全部分类为数字 2），而不考虑实际输入内容。这种统一分类会导致齐次规则持续触发，使个体对所有数据实例产生相同的预测（head 或 -head）。因此，当个体的神经模块统一预测某个原子时，它能够通过“正确”预测其中带有相同原子标签的 $\widehat{p}_{train}$ 和 $\widehat{p}_{val}$ 的比例，从而获得具有误导性的较高相对适应度（head 或 -head）。这种高适应度会导致此类个体被优先选择，使进化过程陷入局部最优（附录B中给出了更详细的技术解释）。

停滞状态现象，以及在进化早期代中符号策略不完整时训练神经网络所面临的挑战，促使我们引入重构损失作为一种自监督学习机制。如第3.1节所述，当个体仅具备最少的符号策略时，可用于计算语义损失的溯因反馈受到限制，可能会妨碍有效的神经学习。该框架引入了带有 Gumbel- Softmax 正则化的重构损失分量，以便在这些充满挑战的阶段提供额外的训练信号。然而，第4.5 节报告的消融研究结果表明，仅语义损失就足以实现成功收敛，这说明来自符号模块的溯因反馈即使在策略最初不完整的情况下，也能为神经网络提供充足的训练信号。这简化了所提框架：重构损失分量虽然最初基于理论动机被纳入，但对于实现强劲性能而言并非必要。该框架仅依靠语义损失即可有效运行，从而降低架构复杂性，并消除与解码器网络和重构损失计算相关的计算开销。

归根结底，只要世代足够多，大多数陷入停滞的实验最终都会摆脱停滞状态。这一点在图9中显而易见：训练结束时，中位数正确表现接近100%，而相应的均值性能略逊一筹（但仍呈上升趋势），这表明训练结束时仍处于停滞状态的实验比例极小。图9阴影区域所示的四分位距（第一四分位数至第三四分位数）反映了实验间的差异，这种差异源于进化过程（随机规则生成与选择）和神经网络训练（权重初始化与批次采样）各自的随机性。这种差异体现了进化搜索中固有的探索与利用权衡：有些谱系通过偶然的突变更快地发现有效的符号策略，而另一些则需要经历更多世代才能摆脱搜索空间中的次优区域。

鉴于第4.6节所述的端到端神经基线表现出色，一个自然而然的问题是：既然端到端神经基线以大幅较低的计算成本取得了相当甚至更优的性能，为何还要采用进化式神经符号方法？该基线使用单个网络训练100个训练轮次，中位数准确率超过98%；而进化框架需要在多个世代中训练多个个体，因此计算开销相差显著。这种差异体现了引言中提到的可解释性的代价，即除了神经组件之外，还要获得人类可读的符号策略，所需的额外计算资源和更长训练时间。

这种权衡的价值在于，进化过程中产生了可解释的符号策略。与纯神经网络的黑箱特性不同，学习得到的符号策略能够提供

明确且可检查的逻辑规则，用以解释系统的推理过程。领域专家无需具备机器学习专业知识，就能审查这些策略，并在必要时进行修改，同时验证其是否与领域知识一致。无论端到端神经方法的预测性能如何，符号组件都能提供其根本不具备的透明度。

在可解释性和信任至关重要的领域，这种权衡是合理的。在安全关键型应用（医学诊断、自主系统）、法律和监管场景（信贷决策、刑事司法）以及需要可解释模型的科学领域，检查和验证推理过程的能力可能比计算效率更重要。当决策必须向利益相关方说明、接受合规审计或由领域专家验证时，符号策略所提供的透明度是端到端神经基线模型无法比拟的。额外的计算成本换来了可验证的符号策略，使问责和领域专家验证成为可能，而这在此类场景中可能至关重要。

6 相关工作

早期的神经符号集成方法侧重于使符号推理可微，从而能够进行基于梯度的学习。Serafini 和 d'Avila Garcez (2016年) 提出了逻辑张量网络（LTNs），将逻辑构造映射为可微张量运算；而 Evans 和 Grefenstette (2018年) 提出了 ∂ILP，将符号规则嵌入为可微组件，并通过以可微计算实现的前向链式演绎，使其能够进行反向传播。这些奠基性工作表明，符号推理可以通过可微性与深度学习集成，但两者都要求符号组件支持梯度传播。

近期研究推动了神经感知与符号推理的联合学习。Mao 等人(2019) 提出了神经符号概念学习器（NS-CL），展示了解耦感知与符号推理如何通过联合学习视觉概念、词义和语义解析，实现透明的组合式视觉理解。 Dai 和 Muggleton(2021 年) 提出了 MetaAbd，该方法在期望最大化框架下联合学习原始数据中的亚符号感知和符号推理，成功归纳出可复用的逻辑程序，并在预测准确率和数据效率方面优于端到端神经基线。

基于这一范式，Cunnington 等人(2022年, 2024年) 开发了 NSIL，它交替进行潜在概念的神经学习和答案集编程假设学习，后来又将其扩展到处理未标注原始数据。 Charalambous 等人 (2023) 提出了 NeuralFastLAS，通过语义损失，在训练神经网络的同时学习规则后验概率，从而以更短的训练时间实现高准确率。 Daniele 等人 (2023) 提出了深度符号学习（DSL）。该方法不再依赖给定的符号知识，而是在可微分流程中联合学习感知和符号组合函数，并通过受强化学习启发的策略函数做出离散的符号选择。

自动谓词发明方面也取得了互补进展，Sha 等人 (2023) 提出了 NeSy-π ，用于从视觉场景中自动发明谓词；而Barbiero 等人(2023) 提出的深度概念推理器无需显式的概念监督，即可根据概念嵌入生成并评估模糊逻辑规则，展示了直接从嵌入中学得的语义一致的推理能力。

集成方法在各类方法学中日趋多样化。 D´ıaz-Rodr´ıguez 等人 (2022年) 将深度学习与领域知识图谱相结合，通过可解释人工智能指导的训练，使特征归因与知识保持一致。 Pryor 等人 (2023) 提出了 NeuPSL，将神经谓词扩展到概率软逻辑中，以支持端到端梯度训练。 Zhong 等人 (2023) 通过 ChatABL 将大语言模型集成到溯因学习中，把视觉特征转换为自然语言逻辑事实，以便进行基于大语言模型的推理。

近期的综述为该领域提供了系统性的视角。 Vermeulen 等人（2023）对 NeSy 任务进行了分类，并指出概率逻辑编程方法虽然计算成本较高，但能取得更优的性能；而Marra （2024年）提出了神经符号配方，这是一种统一的两阶段流程，可促进不同方法之间的系统比较与设计。

尽管这些方法推动了神经符号集成的发展，但大多数方法都假设符号结构、规则模板或背景知识是预先定义的。我们的工作通过进化学习解决了这一局限：从空策略开始，通过变异和选择逐步构建符号知识，无需依赖可微性或预定义结构。这种并行开展神经学习与符号学习的进化方法，为神经符号系统提供了互补的发展方向，尤其适用于难以从领域专家处获取符号知识的领域。

7 结论与未来工作

在本研究中，我们提出了一种新框架，可在神经符号系统中同时训练神经组件和符号组件。大量实验验证了该方法的有效性，表明神经符号系统可以从空的符号策略和随机初始化的神经权重开始，并逐步进化，以逼近隐藏的目标策略。可以说，我们工作的最重要贡献在于证明了在神经符号系统中，可以在训练神经网络的同时学习不可微策略。神经符号系统中可进化策略这一能力的展示，或许是推动相关研究的第一步，尤其是在难以从领域专家处获取符号知识的领域。

尽管我们针对重构损失开展的消融研究表明，它并非必不可少，但未来的工作可以探究框架其他组成部分对性能指标的各自贡献。此外，我们还将探索能否用其他搜索方法取代进化，例如采用不那么贪婪的进化过程版本，以便在种群中保留不同形式的多样性（例如质量多样性优化技术，参见Chatzilygeroudis 等人，2021年；Mouret 和

Clune，2015；Vassiliades 等人，2018年）。另一条研究方向是尝试更大的目标策略（即增加二元概念的数量），并使用所提框架处理来自真实世界场景的数据。

资金支持

本项目获得了欧盟地平线2020研究与创新计划的资助（资助协议第739578号），并得到塞浦路斯共和国欧洲项目、协调与发展总局的补充资助。

致谢

我们衷心感谢 Kyriacos Mosphilis 和 Vasileios Markos 在项目期间进行的富有见地的讨论。

补充材料

本文所述全部实验的复现所需源代码、实验配置和数据生成脚本均已公开发布，地址为 https://github.com/CYENS/evolvable-nesy。

注释

1. “策略”一词用于指代基于逻辑的知识，文献中对此有多种称呼，例如“理论”（Tsamoura 等人，2021年）、“程序”（Gaunt 等人，2017年）、“知识库”（Michael2019）等。2. 原始的可进化性框架通过克隆突变（第3.2节）确保中性组非空，但在我们的情形下，这一点并不能得到保证，因为神经模块引入了随机性，且会与符号模块发生交互。3. 值得注意的是，溯因方法在神经模块中并不对应任何有意义的功能，但为了保持对称而将其纳入其中。

参考文献

Barbiero P、Ciravegna G、Giannini F、Zarlenga ME、Magister LC、Tonda A、Lio P、Precioso F、Jamnik M 和 Marra G （ 2023 ）《可解释神经符号概念推理》。载于：第*40*届国际机器学习大会论文集。PMLR，第1801–1825页。Bertsimas D、Delarue A、Jaillet P 和 Martin S（2019）《可解释性的代价》。doi：10.48550/arXiv.1907.03419。Besold TR、d’Avila Garcez A、Bader S、Bowman H、Domingos P、Hitzler P、K¨uhnberger KU、Lamb LC、Lima PMV、De Penning L、Pinkas G、Poon H 和 Zaverucha G（2021）第1章：神经符号学习与推理：综述与解读。载于：Hitzler P 和 Sarker MK（编）人工智能与应用前沿。IOS Press。ISBN 978-1-64368- 244-0 978-1-64368-245-7，第1–51页。doi：10.3233/FAIA210348。Charalambous T、Aspis Y 和 Russo A（2023）Neural-FastLAS：基于原始数据的快速逻辑学习。doi：10.48550/arXiv.2310.05145。Chatzilygeroudis K、Cully A、Vassiliades V 和 Mouret JB（2021）质量-多样性优化：随机优化的一个新分支。载于：Pardalos PM、Rasskazova V 和 Vrahatis MN（编）黑箱优化、机器学习与无免费午餐定理，第170卷。Cham：Springer International 出版信息。ISBN 978-3-030-66514-2 978-3-030-66515-9，第109–135页。 - doi:10.1007/978-3-030-66515-9 4。Chavira M 和 Darwiche A（2008年）基于加权模型计数的概率推理。Artificial Intelligence 172(6)：772–799。doi:10.1016/j.artint.2007.11.002。Cunnington D、Law M、Lobo J 和 Russo A（2022年）从原始数据归纳学习复杂知识。载于：AAAI秋季研讨会，第3332卷。美国弗吉尼亚州阿灵顿：CEUR研讨会论文集，第1–21页。Cunnington D、Law M、Lobo J 和 Russo A（2024年）从原始数据进行神经符号学习，学习答案集程序。doi:10.48550/arXiv.2205.12735。Dai WZ 和 Muggleton SH（2021年）从原始数据进行溯因知识归纳。doi:10.48550/arXiv.2010.03514。Dai WZ、Xu Q、Yu Y 和 Zhou ZH（2019）通过溯因学习连接机器学习与逻辑推理。载于：Wallach H、Larochelle H、Beygelzimer A、dAlch´e-Buc F、Fox E 和 Garnett R（编）《神经信息处理系统进展》，第32卷。加拿大温哥华：Curran Associates, Inc.，第2811–2822页。Daniele A、Campari T、Malhotra S 和 Serafini L（2023）深度符号学习：从感知中发现符号与规则。doi:10.48550/arXiv.2208.11561。Darwiche A（2011年）SDD：命题知识库的一种新型规范表示。载于：第二十二届国际人工智能联合会议，第819–826页。Dasgupta S、Frost N、Moshkovitz M 和 Rashtchian C（2020）可解释的k均值聚类：理论与实践。载于：XXAI研讨会。奥地利维也纳，第1–8页。d’Avila Garcez A 和 Lamb LC（2023）神经符号人工智能：第三次浪潮。Artificial Intelligence Review 56(11)：12387–12406。doi:10.1007/s10462-023-10448-w。Dessain J、Bentaleb N 和 Vinas F（2023）人工智能中的可解释性代价：以信用评分模型为例。载于：Longo L（编）《可解释人工智能》。Cham：Springer Nature Switzerland。ISBN 978-3-031-44064-9，第498–516页。doi:10.1007/978-3-031-44064-9 26。D´ıaz-Rodr´ıguez N、Lamas A、Sanchez J、Franchi G、Donadello I、Tabik S、Filliat D、Cruz P、Montes R 和 Herrera F（2022年）可解释神经符号学习（X-NeSyL）方法：将深度学习表示与专家知识图谱融合——MonuMAI文化遗产用例。Information Fusion 79：58–83。doi:10.1016/j.inffus.2021.09.022。Evans R 和 Grefenstette E（2018年）从噪声数据中学习解释性规则。Journal of Artificial Intelligence Research 61：1–64。doi:10.1613/jair.5714。Garcia GGP、Steimle LN、Marrero WJ 和 Sussman JB（2024年）高血压治疗规划中的可解释策略与可解释性的代价。Manufacturing & Service Operations Management 26(1)：80–94。doi:10.1287/msom.2021.0373。Gaunt AL、Brockschmidt M、Kushman N 和 Tarlow D（2017年）基于神经网络库的可微程序。载于：第34届国际机器学习会议论文集。PMLR，第1213–1222页。Glorot X 和 Bengio Y（2010）理解深度前馈神经网络训练的难点。载于：第十三届国际人工智能与统计会议论文集。《JMLR》研讨会与会议论文集，第249–256页。

Hitzler P 和 Sarker MK（编）（2021年）《神经符号人工智能：技术现状》，《人工智能与应用前沿》，第342卷。IOS Press。ISBN 978-1-64368-244-0 978-1-64368-245-7。doi:10.3233/FAIA342。Jang E、Gu S 和 Poole B（2017年）使用Gumbel-Softmax的分类重参数化。doi:10.48550/arXiv.1611.01144。Kakas AC（2017年）溯因。载于：Sammut C 和 Webb GI（编）《机器学习与数据挖掘百科全书》第二版。美国马萨诸塞州波士顿：Springer US，第1–8页。Kingma DP 和 Ba J（2017年）Adam：一种随机优化方法。doi:10.48550/arXiv.1412.6980。Laber ES 和 Murtinho L（2021年）关于某些聚类问题中可解释性的代价。载于：第38届国际机器学习会议论文集。PMLR，第5915–5925页。Lecun Y、Bottou L、Bengio Y 和 Haffner P（1998）应用于文档识别的基于梯度的学习。《IEEE会刊》86(11)：2278–2324。doi:10.1109/5.726791。LeNail A（2019）NN-SVG：可用于出版的神经网络架构示意图。《开放源代码软件期刊》4(33)：747。doi:10.21105/joss.00747。Liang Y、Li S、Yan C、Li M 和 Jiang C（2021年）解释黑箱模型：深度神经网络局部解释方法综述。《神经计算》419：168–182。doi:10.1016/j.neucom.2020.08.011。Maddison CJ、Mnih A 和 Teh YW（2017年）Concrete分布：离散随机变量的连续松弛。doi:10.48550/arXiv.1611.00712。Mao J、Gan C、Kohli P、Tenenbaum JB 和 Wu J（2019）神经符号概念学习器：从自然监督中解释场景、词语和句子。doi:10.48550/arXiv.1904.12584。Markos V 和 Michael L（2022年）Prudens：一种用于认知助手的基于论证的语言。载于：Governatori G 和 Turhan AY（编）《规则与推理》，《计算机科学讲义》。Cham：Springer国际出版公司。ISBN 978-3-031-21541-4，第296–304页。doi:10.1007/978-3- 031-21541-4 19。Markos V、Thoma M 和 Michael L（2022年）使用代理教练进行Machine Coaching。载于：Kuhlmann I、Mumford J 和 Sarkadi S（编）《第一届论证与机器学习研讨会论文集》，CEUR研讨会论文集，第3208卷。威尔士卡迪夫：CEUR，第45–64页。Marra G（2024年）从统计关系人工智能到神经符号人工智能。《AAAI人工智能会议论文集》38(20)：22678–22678。doi:10.1609/aaai.v38i20.30294。McCarthy J（1998）详述容忍性。载于：《Common Sense 98》，第98卷。英国伦敦，第2页。Michael L（2019）Machine Coaching。载于：IJCAI 2019可解释人工智能研讨会。中国澳门，第80–86页。Michael L（2023）第11章。自学式与可辅导神经架构。载于：Hitzler P、Sarker MK 和 Eberhart A（编）《神经符号人工智能纲要》，《人工智能与应用前沿》，第369卷。IOS Press。ISBN 978-1-64368-406-2 978-1-64368-407-9，第235–248页。doi:10.3233/FAIA230143。Mouret JB 和 Clune J（2015）通过精英映射照亮搜索空间。doi:10.48550/arXiv.1504.04909。Pryor C, Dickens C, Augustine E, Albalak A, Wang W and Getoor L (2023) NeuPSL：神经概率软逻辑。doi:10.48550/arXiv.2205.14268。Serafini L and d'Avila Garcez A (2016) 逻辑张量网络：基于数据和知识的深度学习与逻辑推理。doi：10.48550/arXiv.1606.04422。Sha J, Shindo H, Kersting K and Dhami DS (2023) 神经符号谓词发明：从视觉场景中学习关系概念。载于：*NeSy 2023，第17届神经符号学习与推理国际研讨会*，第3432卷。意大利锡耶纳蓬蒂尼亚诺修道院：CEUR研讨会论文集，第1–15页。Trinh TH, Wu Y, Le QV, He H and Luong T (2024) 无需人工演示解决奥林匹克几何问题。《自然》《自然》625(7995)：476–482。doi：10.1038/s41586-023-06747-5。Tsamoura E, Hospedales T and Michael L (2021) 神经符号集成：组合视角。*AAAI人工智能会议论文集*35(6)：5051–5060。Valiant LG (1984) 可学习性理论。《*ACM*通讯》27(11)：1134–1142。doi：10.1145/1968.1972。Valiant LG (2009) 可进化性。《*ACM*学报》56(1)：1–21。doi：10.1145/1462153.1462156。Vassiliades V, Chatzilygeroudis K and Mouret JB (2018) 使用质心Voronoi剖分扩展多维表型精英档案算法。*IEEE*进化计算汇刊22(4)：623–630。doi：10.1109/TEVC.2017.2735550。Vermeulen A, Manhaeve R and Marra G (2023) 神经符号系统的实验综述。载于：Bellodi E, Lisi FA and Zese R (编)归纳逻辑编程，《计算机科学讲义》。Cham：Springer Nature Switzerland。ISBN 978-3-031-49299-0，第124–138页。doi：10.1007/978-3-031-49299-0 9。wannesm (2024) PySDD。Xu J, Zhang Z, Friedman T, Liang Y and Van den Broeck G (2018) 用于结合符号知识的深度学习的语义损失函数。载于：第*35*届国际机器学习会议论文集。瑞典斯德哥尔摩：PMLR，第5502–5511页。Zhong T, Wei Y, Yang L, Wu Z, Liu Z, Wei X, Li W, Yao J, Ma C, Li X, Zhu D, Jiang X, Han J, Shen D, Liu T and Zhang T (2023) ChatABL：通过与ChatGPT进行自然语言交互实现溯因学习。doi：10.48550/arXiv.2304.11107。

A 实证设置（详情）

进化过程中的相对适应度是根据表中所示的评分矩阵计算得出的2。

实验中使用的卷积神经网络架构如图所示13。NeSy 生物体的 神经模块 和端到端基线均使用了这一基础编码器架构。由于每个数据实例由八张 MNIST 图像组成，因此依次使用同一个神经网络八次，以获得所有输入图像的预测。对于 NeSy 生物体，编码器为每张图像输出 n 个神经元（8 个原子），总共产生 8n 个输出；这些输出经过拼接后用作符号原子激活。对于端到端基线，这 8n 个拼接后的输出则会被展平，并传入另外两个全连接层：一个包含 64 个神经元并采用 ReLU 激活的隐藏层，后接一个包含 2 个神经元的输出层（分别表示 正面和-正面），并通过 Softmax 进行二元分类。

每项实验均在 Rocky Linux 8.5 环境中运行，使用 Python v3.11.6，配备 4 个 CPU 核心和 48GB 内存，以及一块具有 24GB 显存的 Nvidia RTX A5000 GPU。

表 2。 用于计算相对适应度的评分矩阵

		后代		
		Correct	*Abstain*	*Wrong*
Parent	*Correct*	0	−1	−1
	Abstain	1	0	−1
	Wrong	1	1	0

B 结果（详细信息）

B.1 汇总结果

所有实验在最后一代的详细汇总结果如表3所示。

表 3。 $\widehat{p}_{test}$ 上所有实验结束时的汇总结果。

	中位数	Mean	SD
正确	0.992	0.944	0.109
弃权	0.000	0.002	0.003
错误	0.005	0.053	0.110

B.2 停滞状态

图 17 至 20 展示了部分进化过程中出现停滞状态的各个实验运行的结果；在所有这些情况下，停滞状态最终都被打破。

在某些进化运行过程中，个体进入了停滞状态（即局部最优：它们在训练和验证数据上取得了部分正确表现，但在随后的世代中无法进一步改进）。在符号 S_{+} 或 $S_{\downarrow}$ 变异向策略中添加齐次规则后，个体便会陷入停滞。齐次规则是指其体包含的文字符号具有相同符号的规则，例如“a1、a2、a3 推出 head”（全部为正原子），或“-a1、-a2、-a3 推出 head”（全部为负原子）。

添加此类规则会引发齐次符号结构与序列神经网络架构之间的不利相互作用。具体而言，神经模块会收敛到一种平凡解，将所有输入图像都归为同一类别（全部归为数字 1 或全部归为数字 2），而不考虑输入内容的实际情况。这种一致的分类会使符号策略中的齐次规则对所有数据实例都持续生效，导致符号模块始终给出相同的预测（head 或 -head）。因此，如果神经模块统一输出的原子（head 或 -head）对应的 $\widehat{p}_{train}$ 和 $\widehat{p}_{val}$ 标签占比较高，该个体就会通过“正确”预测这一比例，获得虚高的相对适应度。

这种高适应度使这类个体在进化过程中更容易被优先选择。停滞状态之所以持续存在，是因为后续突变很少能产生适应度超过由遵循齐次规则的个体所达到的局部最优的后代。

上述模式是由神经模块所采用的神经网络架构与不完整的假设策略之间的不利交互导致的。由于假设策略中加入了齐次规则，训练期间这些策略向神经网络提供的信号是片面的。由于神经网络会依次处理每个原子的每张图像（见图4），齐次规则提供的信号有时会导致神经网络无论输入是什么，都始终预测同一个值。

然而，在任何给定时刻，这种模式只出现在 ∼15% 的实验中，而且随着进化过程的推进，这一比例会逐渐降低，这表明在大多数实验中，这种模式最终会被打破。图9清楚地显示了这一点：训练结束时，中位数正确表现接近100%，而相应的平均性能略低一些（但仍呈上升趋势），这表明有一定数量的实验结束时仍处于停滞状态。根据经验观察，只要世代数足够多，大多数陷入停滞的实验最终都会摆脱停滞状态。图12 展示了这样一次实验运行：它在进化过程早期陷入停滞，但最终摆脱停滞，并达到接近100% 的正确表现。

B.3 单次实验图

图 14 至 16 展示了典型实验运行的结果，而图 17 至 20 展示了在其进化过程的部分阶段呈现停滞状态的单次实验运行结果。

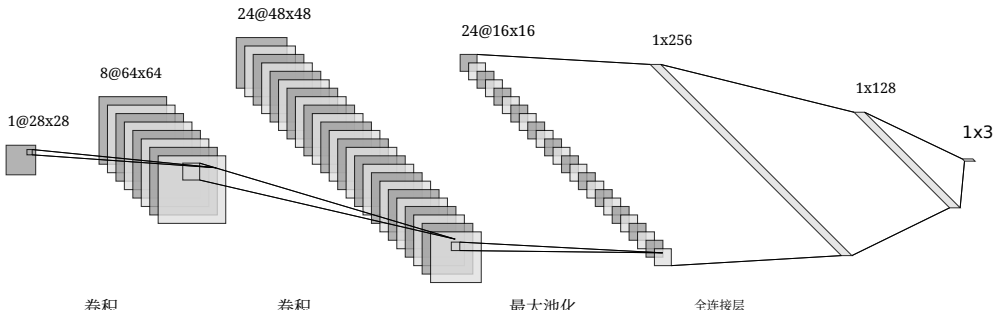


图 13. 实验中使用的卷积神经网络架构。该图使用 NN-SVG 创建（LeNail2019）。

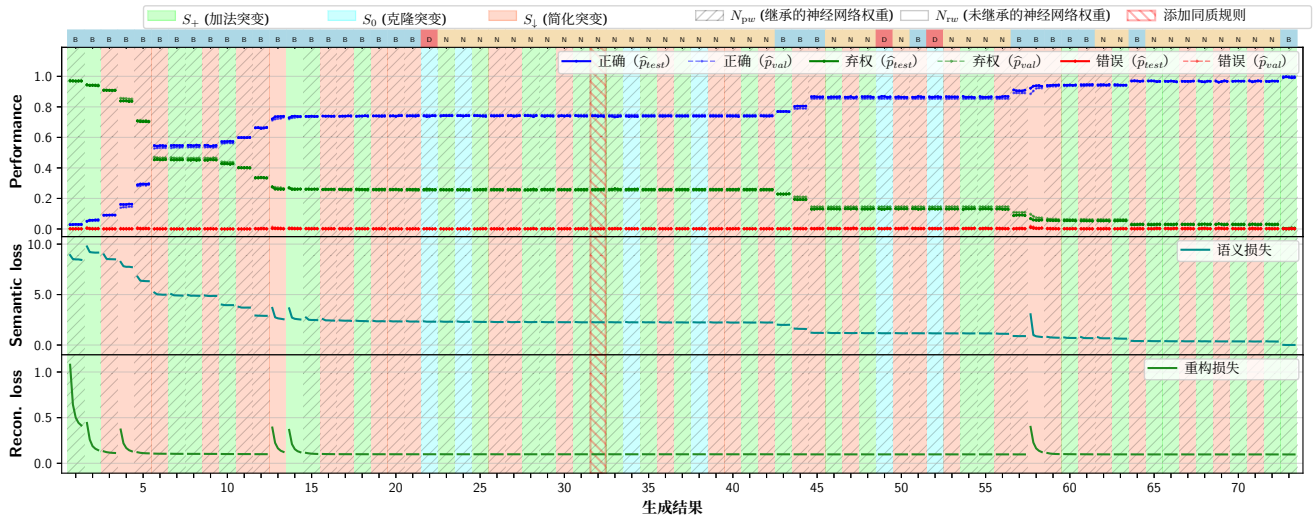


图 14. 典型实验运行。

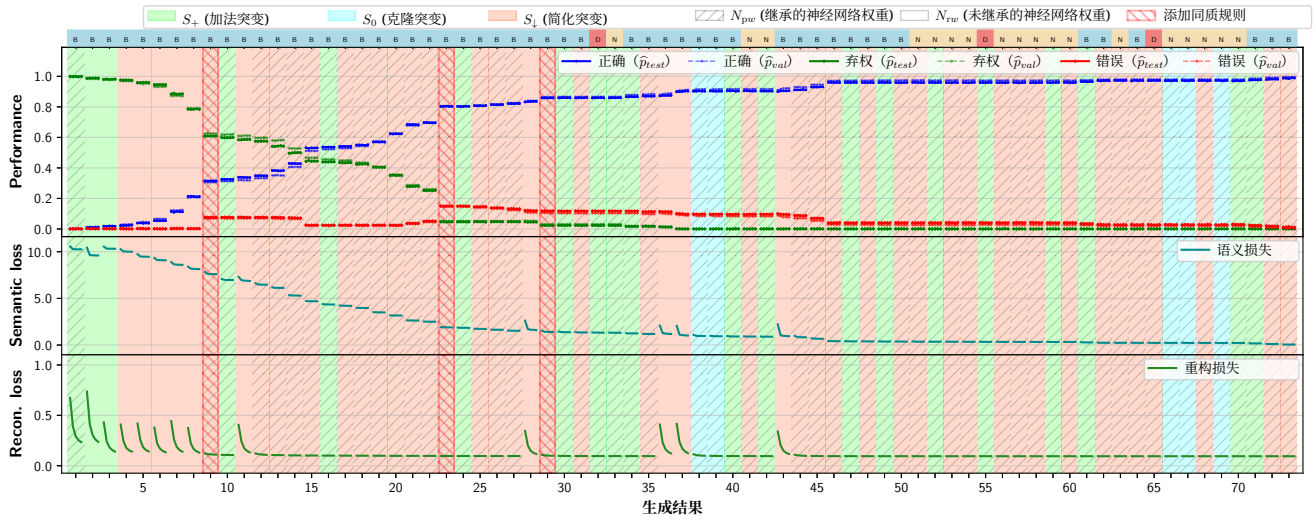


图 15. 典型实验运行。

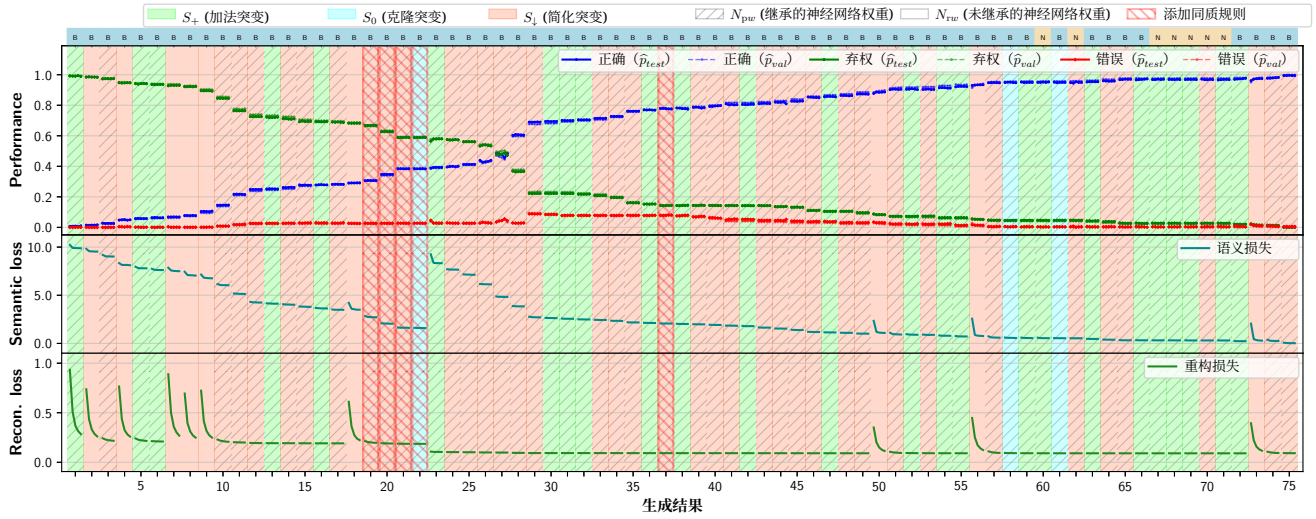


图16. 典型实验运行。

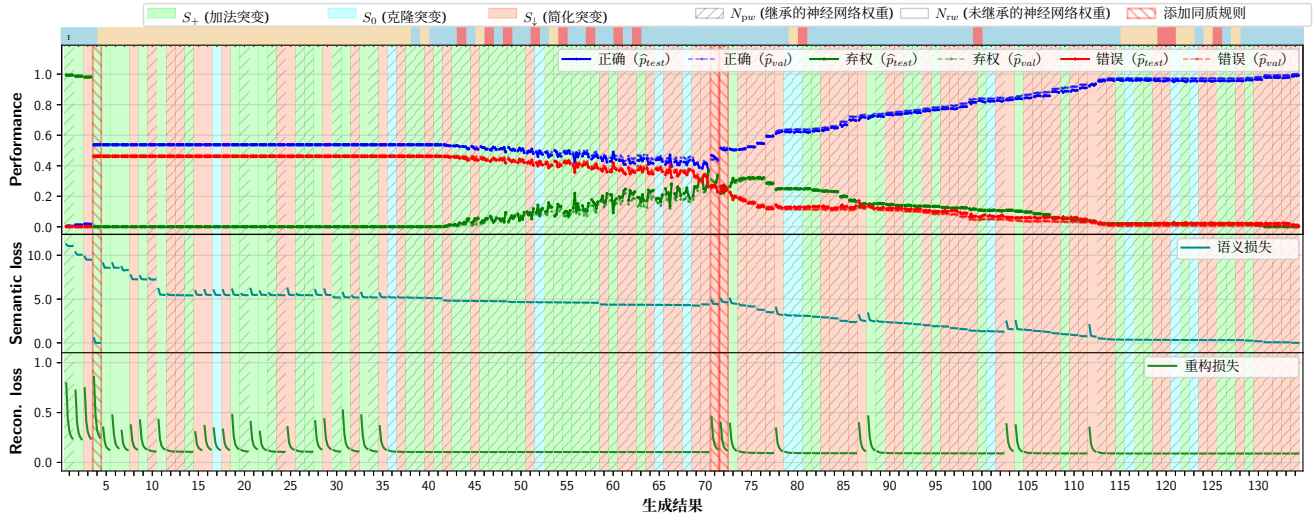


图17. 展示进化过程部分阶段陷入停滞状态的实验运行。

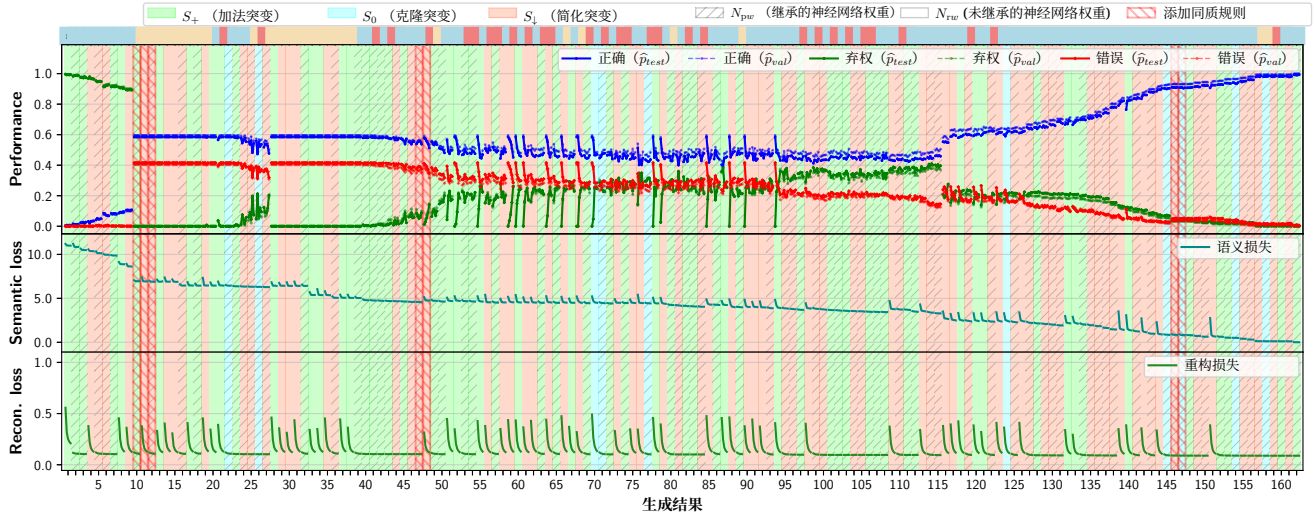


图18. 展示进化过程部分阶段陷入停滞状态的实验运行。

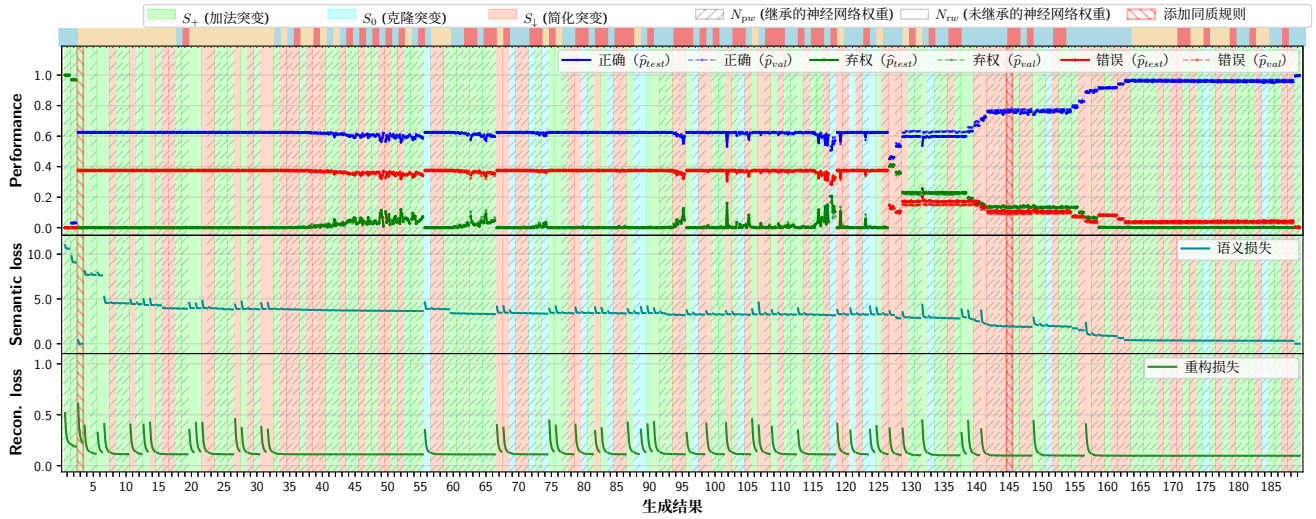


图19. 实验运行展示了进化过程中的一段时间处于停滞状态。

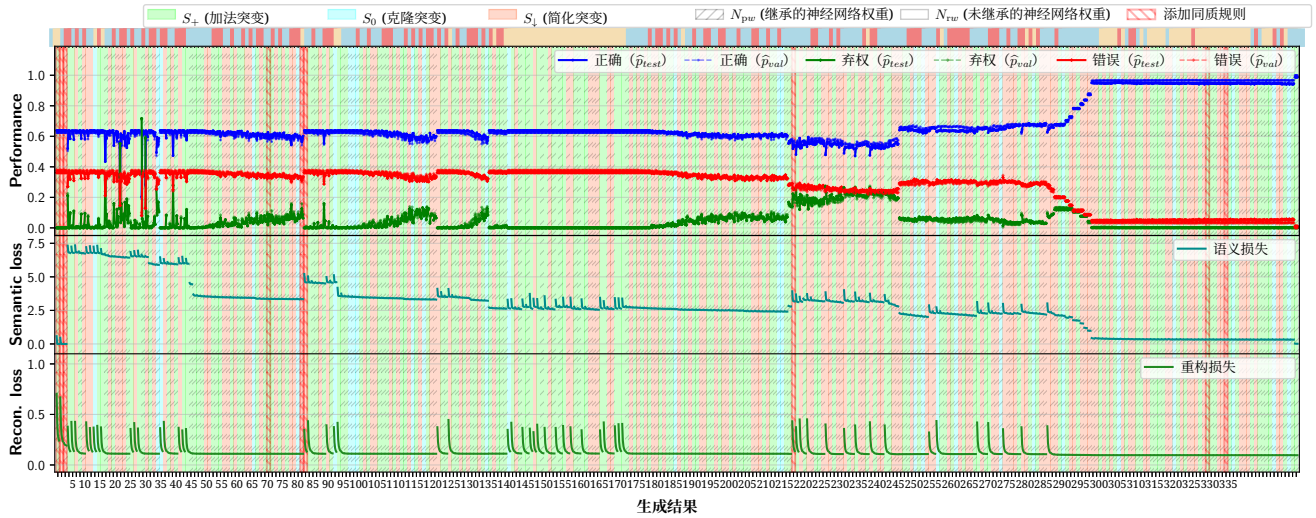


图20. 实验运行展示了进化过程中的一段时间处于停滞状态。