

HEAS: 用于跨尺度建模和多目标搜索的分层进化代理模拟框架

张瑞宇

Department of Politics and Public Administration
The University of Hong Kong
ruiyuzh@connect.hku.hk

林焱

Department of Applied Social Sciences
The Hong Kong Polytechnic University
lin-apss.nie@polyu.edu.hk

新赵

Department of Applied Social Sciences
The Hong Kong Polytechnic University
xinnn.zhao@connect.polyu.hk

ABSTRACT

分层进化代理仿真 (HEAS) 是一个 Python 框架, 它统一了分层的基于代理建模与进化优化和锦标赛评估, 在一个可重复的工作流中。HEAS 将模型表示为轻量级进程 (“流”) 的层次结构, 这些进程以确定性的层次安排运行, 读取和写入共享上下文, 使跨尺度耦合明确且可审计。紧凑的 API 和 CLI——模拟、优化、评估——暴露了单目标和多目标进化, 通过参数展平/还原集成 PyTorch 策略, 并提供了具有用户定义评分和投票规则的一般锦标赛工具。该框架通过统一每步和每个纪元指标来标准化评估, 持久化种子、日志簿和名人堂档案, 并提供用于轨迹、帕累托前沿和比较结果的绘图助手, 减少了胶水代码并提高了研究间的可比性。HEAS 强调机制与编排的分离, 允许外源驱动程序、内生代理和聚合器进行组合和交换而无需重构, 同时同一个模型可用于正向仿真、优化或系统性比较。我们通过两个紧凑示例——一个生态系统和一个企业决策环境来说明使用方法。HEAS 为跨学科多层级研究提供了一个实用基础, 产生可靠且可重复的结果。

Keywords 基于代理的建模 · 进化优化 · 层次建模 · 跨尺度建模 · 多目标搜索

1 介绍

跨越自然科学和社会科学, 决策同时在多个层面展开: 个体在群体中适应; 群体在组织内协调; 组织对市场或机构作出反应; 生态系统反馈以塑造个体的适合度。虽然基于代理的建模 (ABM) 现在是研究此类自适应、相互作用实体 [1, 2, 3] 的标准方法, 但大多数开放工具使得分层组合模型并优化行为或参数变得笨拙, 在同一工作流中难以实现。本文介绍了 **HEAS** (层级进化代理仿真), 这是一个用于构建具有演化代理和可配置决策规则的多层仿真的 Python 框架, 具备集成的进化搜索、锦标赛评估和神经策略。

¹HEAS Python 软件包可以通过以下方式安装: `pip 安装 heas`。我们提供 HEAS 作为一个现成的、易于使用的软件包, 以便研究人员可以直接在他们的工作中使用它。有关文档和教程, 请访问: <https://pypi.org/project/heas/>。

传统 ABM 平台——例如, NetLogo[4]、Repast[5]、Mesa[6] 以及高性能引擎如 FLAME GPU[7, 7]——在定义代理状态和交互规则方面表现出色。然而, (一) 多层设计通常以定制代码的形式表达而不是作为模型对象的一级元素; (二) 进化优化 (单目标或多目标) 需要在仿真器和库 (如 DEAP[8] 或自定义遗传学代码) 之间进行大量的粘合工作; (三) 跨场景 (例如, 机构、生态) 和参与者 (例如, 策略、组织) 的比较锦标赛并不是一个内置概念; (四) 尽管在多代理系统 [9, 10, 11] 中越来越常见, 但将神经控制器集成到 ABM 中对用户来说并不是一项简单的任务。

HEAS 通过四个设计选择来解决这些差距。首先, 它将模型表示为流的层次结构。流是一个具有步骤 () 方法的最小过程 (例如, 环境过程、代理策略或计算游戏结果的“裁判”)。流写入和读取共享上下文, 并在层中调度以形成有向无环执行图。这捕捉到了生物学和社会科学中的跨尺度反馈, 从多层次选择和群体结构进化 [12, 13] 到组织协调和投票 [14, 15]。其次, HEAS 将进化搜索集成为第一类 API: 用户声明参数模式、目标 (包括向量值目标), 并使用诸如 NSGA - II [16]、进化策略 [17] 或 CMA - ES 样式的算子 [18] 等已建立的算法获取名人堂档案和统计数据, 无需显式导入优化器库。第三, HEAS 提供了一个比赛/锦标赛模块用于跨场景 $\times$ 参与者的实验比较, 带有用户定义的评分和投票规则 (例如, 多数、argmax、Condorcet 样式的变体), 促进了单次运行 ABMs 之外的制度和组织选择研究。第四, HEAS 支持 PyTorch 策略: 用户可以插入神经控制器以供代理使用, 并且有辅助工具来展平/解除展平参数, 以便无梯度搜索能够轻松地在权重上操作, 就像在规则参数上一样。

我们展示了 HEAS 在两类问题上的应用。对于科学领域 (生态学、进化论、行为), 我们建模了分层种群过程 (例如, 环境驱动 $\rightarrow$ 群体动态 $\rightarrow$ 个体觅食), 并展示如何通过演化搜索发现多目标权衡下的稳健策略 (例如, 奖励与风险)。对于社会科学领域 (组织、政策、市场), 我们建模了层级决策流程 (信号 $\rightarrow$ 代理 $\rightarrow$ 聚合和投票), 并在不同制度场景下运行锦标赛以比较集体结果。在两个领域中, 相同的抽象概念 (流、层、锦标赛、可视化) 以及相同的优化 API 都适用。

概念上, HEAS 处于基于代理的模拟 [1, 2]、进化计算 [19, 8, 16] 和层级决策 [9, 10] 的交汇点。其贡献并非一个新的优化器或学习规则本身, 而是一个统一的软件抽象, 它 (i) 使层级组合明确且可重用; (ii) 将模拟与搜索和评估耦合在一起; (iii) 降低跨学科研究中的障碍, 在这些研究中分层机制和适应性策略是核心。通过发布 HEAS 作为具有简单 API、CLI 和绘图工具的开放 Python 软件包, 我们旨在实现可重复的多层次实验, 从而连接生物、生态和社会模型。我们认为这种组合将帮助研究人员提出并解答以前难以处理的问题: 哪些协调架构能够产生稳定的合作 [20, 13]? 哪些制度设置能使政策在不同环境中稳健? 哪些演化策略能够在层级或场景之间转移? HEAS 提供了一个紧凑且可扩展的基础, 用于此类研究。

2 相关工作

基于代理的建模 (ABM) 框架, 如 NetLogo[4]、Repast[5]、Mesa[6] 以及 GPU 加速引擎如 FLAME GPU[7], 已经使得跨学科领域的代理模拟变得广泛可及。这些工具强调了代理状态、交互规则和调度。然而, 多层或多层次的组合通常是临时编码的, 而不是作为第一类模型结构定义的。ODD 协议被引入以标准化 ABM 描述并提高再现性 [21], 但它并没有规定用于组成层次过程或将优化集成到模拟循环中的软件抽象。HEAS 旨在通过将层级和跨层流视为显式的建模对象, 并结合仿真与进化搜索和评估来补充这些框架。

在进化计算中, 广泛使用的方法包括遗传算法、演化策略和协方差矩阵适应 [18, 22]。多目标进化算法, 特别是 NSGA-II, 现在已经成为发现折衷方案和 Pareto 集合的标准方法 [16]。DEAP 为这类算法提供了一个灵活的 Python 工具包 [8], 然而将优化器连接到仿真模型通常需要大量的“粘合”代码来定义适应度函数、检查点以及统计数据。HEAS 在单一 API 后整合了这些功能: 用户声明参数模式和目标, 获取统计数据和名人堂存档, 并在分层仿真的现场评估候选方案。

还有一系列工作处于学习和演化的交叉领域。神经演化方法 (例如 NEAT) 优化神经网络的拓扑结构和权重 [23], 并且可扩展的演化策略已被探索作为基于梯度的强化学习的替代方案 [24]。在多智能体强化学习 (MARL) 中, 诸如 RLlib[11] 和 PettingZoo[25] 等框架提供了标准化环境、向量化滚动以及分布式训练。这

些库专注于在扁平任务结构下的学习控制；它们不针对分层外源过程或演化搜索与层次智能体模拟的紧密集成。HEAS 是互补的：它通过 PyTorch 支持神经控制器，同时保留了用于分层组合和演化解优化的统一抽象。

与 HEAS 相关的第二个线程是制度选择、投票和锦标赛风格比较。社会选择理论中的基本结果，包括阿罗不可能定理和多数规则 [14, 15] 的公理化特征，继续为集体决策的计算研究提供信息。在实践中，ABM 论文通常嵌入特定问题的评分、一对一比较和临时聚合规则，这可能会妨碍研究之间的可比性。HEAS 引入了一个通用的锦标赛模块，通过参与者形式化场景，支持用户定义的评分和投票规则，并将评估与优化和可视化结合在一起，从而降低跨组织、政策或生态策略进行系统比较实验的摩擦。

总结来说，先前的工作提供了强大的 ABM 模拟器、成熟的进化算法和有能力的 MARL 堆栈，但在软件方面仍存在一个缺口，即没有将层级组合作为首要概念，并且原生地结合了模拟、进化搜索（包括多目标）和锦标赛评估。HEAS 通过设计紧凑的 Python API 来填补这一缺口，该 API 旨在服务于科学和社会科学研究案例。

3 HEAS 的架构

HEAS 将系统建模为一个分层图流，在离散时间中推进共享的模拟环境。流是最小处理单元，具有步骤 () 方法和可选度量挂钩。每个流从代表当前节拍下所有进程可见状态的关键值上下文中读取并写入数据。通过排列流层，HEAS 在每一节拍内定义了一个有向无环执行顺序：上游层生成信号和状态，下游层可以消费这些内容。这种明确的分层将多尺度耦合（例如环境 → 群体 → 个体）转化为首要模型结构而非临时控制流。图 1 概述了 HEAS 中使用的抽象流-层架构。

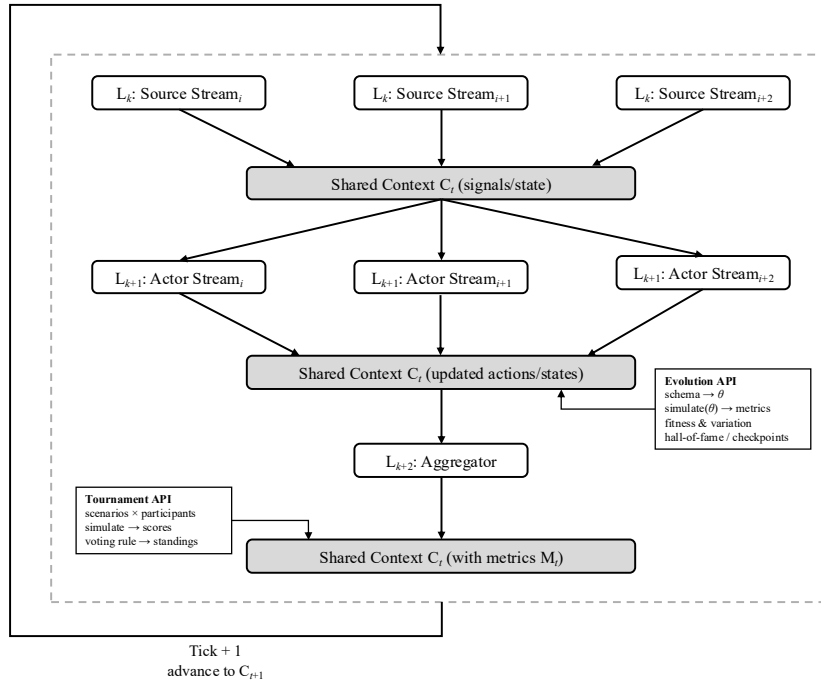


图 1: 摘要流层架构在 HEAS 中。模型由最小的步骤 () 流组成，这些流被调度到确定性层次结构中，读取和写入共享键值上下文 C_t 。在每个时间点内，上游层次更新 C_t ，中间层次从 C_t 读取并写入动作/状态，下游层次聚合指标 M_t ；然后调度器推进到 C_{t+1} 。虚线连接表示用于演化搜索和比赛评估的外部挂钩。

形式上，用 C_t 表示时间 t 的上下文。层 $L^{(k)}$ 包含流 $\{s_i^{(k)}\}$ ，每个流实现一个部分变换 $C' = s_i^{(k)}(C)$ ，读取选定的键并写入新的或更新的键。在一个 tick 期间，HEAS 按确定顺序应用 $L^{(1)}$ 中的所有流，将它们的写操作合并到上下文中（带有用户控制冲突策略），然后继续进行 $L^{(2)}$ ，依此类推。在最后一层触发后，调度器增加 tick 并记录由流暴露的每步指标。由于读取和写入是通过共享上下文进行的，因此跨层反馈变得简单：一个

裁判流可以在下一个 tick 发布收益或最后行动供代理流使用；一个市场流可以聚合微观级别的交易并公开宏观价格以供下游控制器使用。

这种以情境为中心的观点支撑了三种难以结合的能力。首先，层次结构 组成：任何可以表示为情境转换的过程都可以放置在适当的层级中，允许外生驱动因素、内生代理和聚合器共存而不需要特殊情况。其次，评估：HEAS 按步按事件统一指标，使分析能够跨越微观痕迹、中观摘要和宏观结果。第三，搜索：进化 API 将模型运行视为在声明的参数模式上的纯评估函数 $f(\theta)$ ；HEAS 提供种群初始化、变异、适应度塑造、统计信息和名人堂档案，包括多目标适应度。因为仿真本身对优化器视而不见，相同的层次结构模型可以用于正向仿真、锦标赛比较或参数优化而不需重构。

神经控制器自然集成。策略是由 PyTorch 模块支持的普通数据流；HEAS 包括辅助工具来展平和还原模型参数，以便无梯度演进能够像作用于规则参数一样作用于权重向量。这使得在同一个分层系统中混合符号规则和学习控制器成为可能。最后，HEAS 将比较研究编码为锦标赛：场景生成器产生一组上下文或超参数；参与者是模型构造者或策略族；评分函数生成每集的结果；投票规则选择每个场景或总体的获胜者。因为锦标赛在同一上下文和度量接口上运行，它们可以与演化（例如，优化比赛表现的策略）以及层级结构（例如，比较作为不同中间层实现的机构）组合。

3.1 包结构

HEAS 被组织以使上述概念具体化和可重用：

- `heas.api` 提供一站式编排：模拟, 优化, 评估。
- `heas.config` 定义了实验、算法和评估的类型配置以及参数模式。
- `heas.hierarchy` 实现了流、层、图以及执行分层有向无环图并管理共享上下文和指标钩子的复合模型。
- `heas.agent` 提供基础混合项和单个回合滚动的运行器；它与所选代理逻辑无关。
- `heas.evolution` 包含进化算法、种群工具、日志记录和多目标支持，并提供一个稳定的接口（最终用户不直接导入优化器库）。
- `heas.game` 提供了竞技场、场景、锦标赛、评分和投票规则，以便在不同参与者和环境中进行比较实验。
- `heas.torch_集成` 包括神经策略基础和参数展平/还原展平工具，支持权重空间演化和设备管理。
- `heas.schemas` 包含基因原语和基因型到参数映射工具，以连接搜索空间和模型构造器。
- `heas.vis` 提供了步长跟踪、帕累托前沿、进化日志、锦标赛结果和架构图的绘图辅助功能。
- `heas.cli` 暴露了命令（运行, 运行图, 调整, 竞技场, 锦标赛, 即），使得非专业人员可以从 shell 执行分层模拟、优化运行和可视化。
- `heas`. 示例 主持展示层次结构、锦标赛和优化模式的最小可运行模型。
- `heas.utils` 提供可重复的随机数生成器、度量辅助工具和输入输出功能。

此布局在建模（流和层）、评估（指标和锦标赛）以及搜索（进化和模式）之间划分了关注点，同时保持了一个薄的公共 API 和 CLI，涵盖了常见的工作流程：模拟层次结构、优化参数或权重、比较不同场景下的参与者，并可视化动态和结果。

4 科学应用演示：生态案例

我们提供了一个紧凑的生态示例来说明 HEAS 如何支持跨尺度建模、评估和搜索。该意图是教学性的而非机械性的：模型是一个具有季节性强迫、斑块网络、捕食者-猎物动态和特征驱动扩散的理想化元群落。重点在

于分层流和共享上下文如何让我们在一个连贯的工作流程中组合驱动力、种群、运动、聚合器、多目标进化以及场景锦标赛。

架构。第一层提供外源驱动因素：一个气候流生成具有罕见冲击的季节性信号；一个景观流将此信号映射到斑块质量并暴露一个带有移动成本的稀疏图。第二层包含内生过程：猎物增长与风险敏感的觅食和类型 II 捕食损失；捕食者的反应包括转化和死亡率；一个移动流根据扩散特征重新分配猎物向更吸引人的斑块。第三层汇总每步和每个情节指标（平均生物量、变异系数、捕食者密度、累积局部灭绝）。由于各流读写命名空间键，因此跨层次耦合明确且可审计。

Component	Role	Key params
L1: Climate	seasonal driver + shocks	amp, period, shock_prob
L1: Landscape	patch quality + graph	n_patches, fragmentation, move_cost
L2: Prey	density growth + risk foraging	r, K, risk, β_F , γ_V
L2: Predator	consumption response	conv, mort
L2: Movement	dispersal on graph	dispersal $\in [0, 1]$
L3: Aggregator	per-step & episode metrics	ext_thresh

表 1: HEAS 生态层次和流。短语；参数作为符号或范围。

目标与搜索。我们定义了一个两目标的事件损失：最小化负面猎物生物量和最小化变异性，并带有较小的灭绝惩罚。HEAS 宣布了两个基因模式（风险敏感性，扩散）并应用 NSGA-II；模拟器被视为纯粹的功能评估。然后在同一层次结构中进行比较，在变化气候幅度和破碎度的比赛中使用每步平均生物量作为得分，并在每个事件中采用 argmax 规则。

Component	Purpose	Params
Evolution (NSGA-II)	multi-objective search	pop=24, ngen=6–8, weights= (-1, -1)
Schema	trait genes	risk: [0, 1], disp: [0, 1]
Scenarios	environment grid	amp $\in \{0.4, 0.8\}$, frag $\in \{0.2, 0.5\}$
Participants	policies/traits	baseline= (0.55, 0.35), evolved=champion
Scoring	biomass metric	metric= 捕食者. 捕食者. 平均值_x
Voting	episode winner rule	rule=argmax (per episode)

表 2: 搜索和锦标赛设计在生态演示中的应用。

发现（演示）。Pareto 搜索恢复了倾向于专家的战略：高风险敏感性加上接近零的扩散最大化了训练制度下的生物量。然而，当在一小网格的分布外场景中进行评估时，基线策略赢得了我们玩具锦标赛中的所有回合。这个差距是一个已知模式：局部适应可以在环境中权衡稳健性，适度的扩散可以提供空间保险。在我们的玩具设置中，移动成本和碎片化使平衡偏离了训练目标下的扩散，而分布外的变化暴露了脆弱性。

范围和用途。这些结果并非针对任何特定生态系统；它们展示了 HEAS 如何表达分层机制，统一指标，并将搜索与比较测试结合在一起。为了科学地探索鲁棒性，人们会（一）在场景分布上进行训练（场景批处理），（二）采用风险意识聚合（最坏情况或 CVaR），（三）重新平衡灭绝惩罚，以及（四）对特征添加先验或正则化项。HEAS 支持将这些作为第一类配置而无需更改模型代码：只需更换模式、目标或场景生成器。

5 社会科学中的应用：企业决策制定

企业制定战略选择——投资、合规、合作、标准设定——这些选择是在由公共政策、行业监管和市场信号塑造的制度领域中做出的。HEAS 为这类问题提供了一种自然分解：外生驱动因素（政策、需求冲击）存在于第一层；企业和联盟层面的过程运作于第二层；第三层则汇总结果用于诊断、选择和演化搜索。

我们建模了受监管行业中的两家重点企业 A 和 B。每个时期，它们都会从诸如投资与延期、合作与差异化或遵守规则与搭便车等成对的行动中进行选择。收益取决于价格和成本（受到需求和技术的影响）、合规罚款

Item	Value
Baseline policy	Torch MLP (ref, seeded)
Evolved policy	Torch MLP (NSGA-II champion)
Episodes / scenario	4
Steps / episode	140
Score (per step)	mean prey biomass
Grid-avg biomass (ref)	51.584
Grid-avg CV (ref)	0.167
Grid-avg biomass (champ)	52.641
Grid-avg CV (champ)	0.146
Δ biomass (champ – ref)	+1.057
Δ CV (champ – ref)	–0.022
NSGA-II (pop, ngen)	(20, 5)

表 3: 玩具生态演示的总结。

(政策制度) 以及跨企业的外部性 (溢出效应、标准兼容性)。政府可以是合作型 (以激励为导向) 或指令型 (以惩罚为导向); 行业监管可以宽松或严格; 市场面临具有可配置幅度和持续性的需求冲击。在 HEAS 中, 这些是在第一层中的场景参数。第二层流实现了企业政策 (带有战略记忆)、联盟形成 (带有一个讨价还价规则和旁支付) 以及收益核算。第三层收集每一步骤和每个阶段的指标 (利润和分散度、合规性、联盟稳定性和社会福利), 并将它们传递给优化器和锦标赛。

表 4 列举了流和关键旋钮。策略是紧凑的神经控制器 (PyTorch MLPs), 读取命名空间信号——价格、需求、税收、反垄断、补贴和时间索引, 并生成连续动作: 投资强度和合作倾向。我们用多目标遗传算法进化权重并记录名人堂档案。至关重要的是, 支持优化的同一层次结构也支持锦标赛评估。HEAS 运行一系列场景并将播种参考策略与进化的冠军策略进行比较, 使用每集分数和投票规则。

Layer	Stream	Role	Inputs (key params)	Outputs
L1	GovernmentPolicy	Subsidies & penalties	regime $\in$ {coop, dir}; tax; audit_intensity	subsidy; penalty_rate
L1	IndustryRegime	Standards & thresholds	regulation $\in$ {lenient, strict}; std_compat	compliance_thr; audit_prob
L1	MarketSignal	Demand & shocks	base_demand; shock_amp; attr	demand_t; price_signal
L2	FirmPolicy_A	Action + memory	costs; signals; last actions; compliance_thr	action_A; state_A
L2	FirmPolicy_B	Action + memory	(same as A)	action_B; state_B
L2	AllianceMediator	Join/leave & splits	bargain_rule; side_payment	alliance_state; transfers
L2	PayoffAccounting	Profits, fines, welfare	price_fn; cost_fn; penalties	profit_A; profit_B; welfare_t
L3	Aggregator	Metrics & logging	welfare_weights; risk_penalty	$\bar{\pi}$; $\text{var}(\pi)$; compliance; stability

表 4: HEAS 分层设置用于企业决策的玩具演示。

表 5 汇总了 32 种情形下的利润 (制度 $\oplus$ 行业 $\oplus$ 税收 $\oplus$ 反垄断 $\oplus$ 补贴)。进化冠军在各方面均优于种子参考。收益较大, 并且在各制度下异常均匀 (大约平均增加 660, 无论是在合作还是指令性制度下), 而各个行业的差异主要在于绝对基线的不同。每种情形下的标准差对于两个参与者来说都相对温和, 表明了稳定的控制而非脆弱的机会主义。在每个制度内部, 较高的反垄断政策略微降低了能源行业冠军的平均值, 这与对类似联盟行为的轻微约束一致, 而小规模补贴则为两个行业的利润提供了微弱提升。由于 HEAS 将学习和测试

分开，相同的设置可以立即探究反事实情况：交换评分规则以利润减去排放处罚，实施方差上限或公平性约束，或者在压力场景下评估最坏情况下的表现。

Group	Ref mean μ_{ref}	Champ mean μ_{champ}	$\Delta = \mu_{\text{champ}} - \mu_{\text{ref}}$
面板 A: 总体			
Overall	375.31 $\pm$ 173.20	1,036.26 $\pm$ 194.00	660.95
面板 B: 按政府体制分类			
Cooperative	375.33	1,036.15	660.81
Directive	375.29	1,036.36	661.08
面板 C: 按行业部门划分			
Energy	223.06	880.76	657.71
Tech	527.56	1,191.74	664.18

表 5: 玩具演示在 32 场景下的利润总结 (制度 $\times$ 部门 $\times$ 税 $\times$ 属性 $\times$ 补贴)。

此案例展示了 HEAS 在社会科学问题上的三种设计优势。首先，层次结构是一种建模原语：政策、行业、市场、企业、中介者和聚合器是通过共享上下文进行通信的独立流。交换政策时间表、添加消费者剩余计算器或插入卡特尔合规审计员都不需要重构企业的逻辑。其次，评估是一致的。精确的每步密钥和情节指标驱动仪表盘、优化目标和比赛评分，这使得结果在不同运行之间具有可比性，并减少了胶水代码。第三，学习是优先考虑但可选的。研究人员可以进化神经策略、调整规则参数或保持策略固定并专注于制度变化。任何组合都通过声明一个模式、一个基于情节指标的目标以及场景生成器来表达。

6 创新与结论

创新。HEAS 提供了一个统一的软件抽象，将层次结构组成、评估和进化搜索作为首要建模问题。具体而言，它引入了：(i) 流和层作为显式构建模块，编译成共享上下文上的分层 DAG；(ii) 以上下文为中心的合同，使跨尺度耦合可审计并简化过程重用；(iii) 集成进化（单目标和多目标）通过稳定的 API 公开，使得用户声明模式和目标而不是编写粘合代码；(iv) 比赛模块形式化了场景 $\times$ 参与者，并允许用户定义评分和投票规则；以及 (v) 通过 PyTorch 集成神经策略，参数扁平化/非扁平化使得无梯度搜索可以轻松地在权重上操作，就像对规则参数一样。一个薄公共表面—模拟，优化，评估和 CLI 反映了这些功能，降低了非专家运行分层模拟、优化和比较实验的障碍。

这实现了什么。通过将层级与搜索和锦标赛相结合，HEAS 自然地提出了跨越层次和设计的问题而无需重构：微观规则如何传播到中观结果和宏观稳定性？哪些制度配置在外部环境中是稳健的？进化策略何时会在各层或场景之间传递？由于所有流都读取和写入相同的上下文并暴露统一指标，同一个模型支持 (a) 前向模拟和诊断；(b) 参数或神经权重的进化调整；以及 (c) 跨场景和参与者的锦标赛比较。这种“一个模型，多种视角”的模式减少了冗余工程并提高了实验间的可比性。

层次化表示将机制与编排分开：外生驱动因素、内生代理和聚合器是独立的流，按层调度。评估是一致的：每一步骤和每个时段的指标为仪表盘、目标和投票提供动力，无需专门适配器。搜索是声明式的：模式将基因型映射到构造函数关键字参数，名人堂档案和日志本会自动生成，并且原生处理多目标适应度。对于循环学习研究，PyTorch 策略作为普通流插入；对于基于规则的研究，进化可以针对符号旋钮进行优化。在这两种情况下，场景生成器和投票规则使得比较静态分析和稳健性分析可重复。

HEAS 强调明确配置 (`heas.config`)，确定的时隙内顺序，以及上下文写入冲突的可控解决。种子、运行元数据、日志簿和名人堂工件被持久化以供重新运行和审计；绘图助手 (`heas.vis`) 提供了对跟踪记录、帕累托前沿和锦标赛结果的标准视图。示例画廊展示了生态学和组织领域的最小可执行模式，旨在作为领域特定模型的模板。架构有意设计得很小：大多数用户仅与 API/CLI 进行交互，而高级用户可以扩展流、模式或投票规则。

基于代理的建模、进化计算和多代理学习已经平行成熟，但研究人员在将层级结构、搜索以及比较评估结合到单一工作流时仍然会遇到可以避免的摩擦。HEAS 通过提升分层组合、统一度量、集成演化和锦标赛比较至首要软件原语来解决这一问题。结果是一个紧凑且可扩展的框架，有助于科学家和社会科学家提出多级问题并以可重复实验的方式回答这些问题。通过发布具有简单 API、CLI 和示例库的 HEAS，我们旨在将工作重心从管道转移到研究：指定机制、阐明目标以及在不同场景下质询鲁棒性。在此过程中，HEAS 使跨尺度推理成为常规，并扩大了可以使用基于代理的模拟进行可信研究的问题范围。

参考文献

- [1] Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.
- [2] Steven F Railsback and Volker Grimm. *Agent-based and individual-based modeling: a practical introduction*. Princeton university press, 2019.
- [3] Nigel Gilbert. *Agent-based models*. Sage Publications, 2019.
- [4] Seth Tisue, Uri Wilensky, et al. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA, 2004.
- [5] Nicholson Collier and Michael North. Parallel agent-based simulation with repast for high performance computing. *Simulation*, 89(10):1215–1235, 2013.
- [6] Jackie Kazil, David Masad, and Andrew Crooks. Utilizing python for agent-based modeling: The mesa framework. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 308–317. Springer, 2020.
- [7] Paul Richmond, Robert Chisholm, Peter Heywood, Mozghan Kabiri Chimeh, and Matthew Leach. Flame gpu 2: A framework for flexible and performant agent based simulation on gpus. *Software: Practice and Experience*, 53(8):1659–1680, 2023.
- [8] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner Gardner, Marc Parizeau, and Christian Gagné. Deap: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, 13(1):2171–2175, 2012.
- [9] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.
- [10] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(4):341–379, 2003.
- [11] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International conference on machine learning*, pages 3053–3062. PMLR, 2018.
- [12] Samir Okasha. *Evolution and the levels of selection*. Clarendon Press, 2006.
- [13] Martin A. Nowak. Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563, 2006.
- [14] Ian MD Little. Social choice and individual values. *Journal of Political Economy*, 60(5):422–432, 1952.
- [15] Kenneth O May. A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica: Journal of the Econometric Society*, pages 680–684, 1952.
- [16] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

- [17] Ingo Rechenberg. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie: Workshop, Hannover, 29. Sept.–1. Okt. 1977*, pages 83–114. Springer, 1978.
- [18] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.
- [19] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.
- [20] Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.
- [21] Volker Grimm, Uta Berger, Donald L DeAngelis, J Gary Polhill, Jarl Giske, and Steven F Railsback. The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768, 2010.
- [22] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies—a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.
- [23] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
- [24] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.
- [25] Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.