数据集蒸馏旨在将大型数据集浓缩为少量的合成样本，这些样本可以直接用作训练新模型的替代数据。这项技术在可解释性、神经网络架构搜索、隐私保护以及持续学习等领域具有广泛的应用。尽管在监督学习领域取得了显著的成功，但这些方法尚未扩展到强化学习领域。在强化学习中，由于缺乏固定数据集，大多数蒸馏方法无法使用。

为了解决这一问题，我们正式提出了**行为蒸馏（behaviour distillation）**，这种设定旨在发现并提取训练专家策略所需的信息，将其压缩为一个包含状态-动作对（state-action pairs）的合成数据集，而无需直接访问专家数据。我们随后引入了\*\*基于进化策略生成数据集（HaDES, Hallucinating Datasets with Evolution Strategies）\*\*的方法，这是一种行为蒸馏方法。HaDES 可以发现仅包含四个状态-动作对的数据集，这些数据集通过监督学习能够在连续控制任务中训练出具有竞争力的代理（agent）。

我们证明，这些数据集可以在分布外进行泛化，适用于训练使用广泛架构和超参数的策略。此外，我们展示了该方法在下游任务中的应用，例如零样本（zero-shot）训练多任务代理。除了行为蒸馏之外，HaDES 在强化学习的神经进化领域相比现有方法有显著改进，并在一个标准的监督数据集蒸馏任务中达到了最新的性能（SoTA, State of the Art）。最后，我们还展示了对合成数据集的可视化，可以为人类提供可解释的任务见解。

**引言**

数据集蒸馏（Dataset distillation, Wang et al., 2018）是一项合成少量数据点的任务，这些数据点能够在下游任务中替代大规模真实数据集进行训练。数据集蒸馏不仅具有科学研究价值，还被应用于核心研究领域，如可解释性、架构搜索、隐私保护和持续学习（Lei & Tao, 2023; Sachdeva & McAuley, 2023）。尽管数据集蒸馏在视觉任务中取得了一系列成功，并且最近也扩展到了图学习（Jin et al., 2021）和推荐系统（Sachdeva et al., 2022），但蒸馏方法尚未应用于强化学习（RL）。这是因为这些方法通常假设先验专家（或真实数据集）的可用性。

为了解决这一研究空白，我们提出了一种新的设定，称为**行为蒸馏（Behaviour Distillation）**，旨在无需专家的情况下，发现并浓缩训练专家策略所需的信息到一个合成的状态-动作对数据集中。与数据集蒸馏仅仅将困难的监督学习任务替换为一个较容易的任务不同，行为蒸馏同时解决了以下两个挑战：

1. **探索问题**：发现具有高期望回报的轨迹。
2. **表示学习问题**：学习能生成这些轨迹的策略表示。  
   这两个问题是深度强化学习的核心。

因此，行为蒸馏旨在生成一个数据集，使得探索过程不再必要，本质上是“预先解决”了环境。因此，行为蒸馏数据集并不是对整个环境的总结，而只是对该环境中专家策略的总结。换句话说，它将数据收集（即探索）和对大量非平稳数据的顺序学习这两个联合问题，简化为对少量平稳、非顺序的合成数据进行监督学习的问题（如图1中的示例数据集所示）。

**图1说明**

图1展示了训练最优 Cartpole 策略（上方）和专家级 Hopper 策略（下方）所需的完整合成数据集。这些状态-动作对有助于解释学习到的策略。红框中包含 Cartpole 的观测特征和 Hopper 的动作标签（力矩）。

受到行为蒸馏挑战的启发，我们提出了 **HaDES（Hallucinating Datasets with Evolution Strategies）** 方法，这是一种基于元进化外循环的技术。具体而言，HaDES 使用双层优化结构来优化合成数据集：

* 外循环使用进化策略（ES）更新数据集，
* 内循环在当前数据集上进行监督学习（即“行为克隆”）。

ES 的适应度函数是监督学习步骤结束时策略的性能。实验表明，生成的数据集可以用于重新训练与原始数据集产生时的架构和超参数完全不同的策略，并在直接训练于原始环境的情况下实现竞争性的回报。更重要的是，仅使用不到一个完整回合数据的 1/101/10 或 1/1001/100 的数据量即可完成行为克隆。我们还展示了这些数据集在下游任务中的适用性，并开源了数据集，以期推动未来研究。

**背景与挑战**

近年来，由于其通用性和对非可微目标的适用性，以及在长时延奖励任务中的表现，进化策略（ES）在机器学习中的兴趣重新兴起（Lu et al., 2023; Salimans et al., 2017）。然而，目前的进化优化方法在可进化参数的数量上存在限制，因为大参数数量与大种群规模相结合会导致巨大的内存占用（见第5.1节）。这限制了 ES 在训练大规模神经网络中的应用。

为应对这一问题，我们将 HaDES 改进为一种用于神经进化的替代参数化和训练方案，即不重新采样内循环策略的初始权重。这种参数化具有独立于进化策略参数数量扩展的优势，从而减少了内存占用，并在多个环境中表现出与传统 ES 方法相当的性能。

**我们的主要贡献**

1. **正式定义行为蒸馏**：将数据集蒸馏的原理扩展到强化学习领域（见第4.1节）。
2. **提出 HaDES**：这是第一个用于行为蒸馏的方法（见第4节）。
3. **改进 ES**：通过对 HaDES 的小改动，提出一种新的 ES 参数化方式，降低了内存占用（见第5.1节）。
4. **实证分析**：证明 HaDES 能够在离散和连续控制环境中生成高效的合成数据集，并且可用于训练范围广泛的策略架构和超参数（见第5.2节）。
5. **下游任务应用**：将合成数据集用于快速训练多任务智能体，这些数据集来源于单一环境的个体数据（见第5.3节）。
6. **数据集蒸馏基准测试**：在通用数据集蒸馏基准上达到 SoTA（见第5.4节）。
7. **开源**：我们开源了代码和合成数据集，网址为 <https://github.com/FLAIROx/behaviour-distillation>。

### 2 相关工作

#### 2.1 数据集蒸馏

减少机器学习训练所需数据量的研究可以追溯到支持向量机简化模型（Reduced Support Vector Machines，Lee & Mangasarian, 2001; Lee & Huang, 2007; Wang et al., 2005）。在深度学习领域，Bachem 等人（2017）和 Coleman 等人（2019）将数据集选择（Coreset Selection）定义为从数据集中挑选少量具有代表性的样本，从而在不降低最终性能的情况下加速模型训练的问题。

Wang 等人放弃了对真实样本的限制，转而生成合成数据集，首次提出了数据集蒸馏的任务。自此以来，许多方法被提出用于蒸馏监督学习数据集（Lei & Tao, 2023）。大多数方法采用双层优化程序，可以分为以下四大类（Sachdeva & McAuley, 2023）：

1. **梯度匹配方法**（Gradient Matching Methods, Zhao et al., 2020）：旨在最小化模型在训练合成数据集和真实数据集时梯度更新之间的差异。
2. **轨迹匹配**（Trajectory Matching, Cazenavette et al., 2022）：最小化模型在两种数据集上训练后的检查点参数之间的距离。然而，这两种技术在没有专家策略、其检查点或至少专家轨迹数据集的情况下，无法应用于强化学习。
3. **分布对齐**（Zhao & Bilen 和 Wang 等）：直接将合成数据集的分布与真实数据集对齐。尽管效果显著，但由于强化学习中的数据分布是非平稳且与策略相关的，这种方法也无法应用。
4. **元模型匹配**（Meta-Model Matching, Wang et al., 2018; Nguyen et al., 2020; Loo et al., 2022）：在内循环中对合成数据集进行完全模型训练，同时在外循环更新数据集以最小化模型在真实数据上的损失。这些方法要么通过时间反向传播（Backpropagation Through Time, BPTT）计算昂贵的元梯度（Wang et al., 2018; Deng & Russakovsky, 2022），要么在内循环中使用神经切线核（Neural Tangent Kernel），以便对目标数据集上的分类器损失进行闭式计算（Nguyen et al., 2020）。

虽然这些方法可以通过选择适当的损失函数（例如 REINFORCE, Williams, 1992）应用于强化学习，但我们的工作在外循环中用进化方法代替元梯度，使外循环更新的成本与网络规模和内循环更新次数无关。这一改进尤为重要，因为我们在实践中可能需要在内循环中进行数百次策略更新，使用 BPTT 的成本过于高昂。

此外，一些工作已将数据集蒸馏扩展到图（Jin et al., 2021; 2022）和推荐系统（Sachdeva et al., 2022），但据我们所知，尚未有任何研究摆脱对预先存在的目标数据集的依赖。因此，我们的工作首次突破了数据中心化范式，并引入了第一个适用于强化学习蒸馏的通用方法。

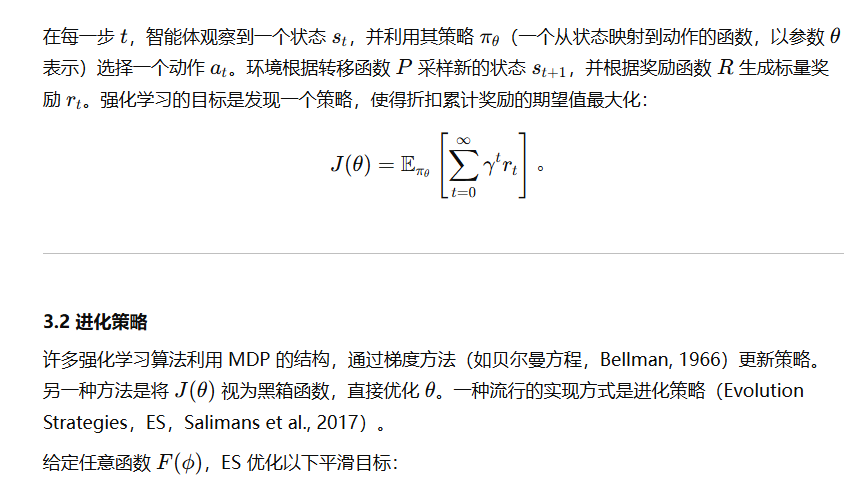
#### 2.2 神经进化与间接编码

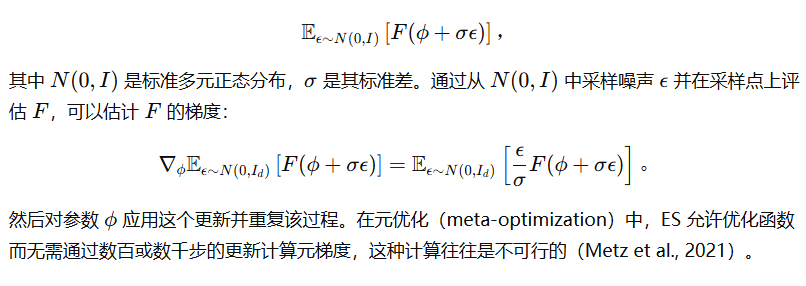
神经进化（Neuroevolution, Schwefel, 1977）在一些基准测试中表现出与强化学习相当的性能（Such et al., 2017; Salimans et al., 2017）。我们的一部分工作可以看作是神经进化的一种间接编码形式（Stanley et al., 2019）——一种用于进化神经网络权重的替代参数化方式。

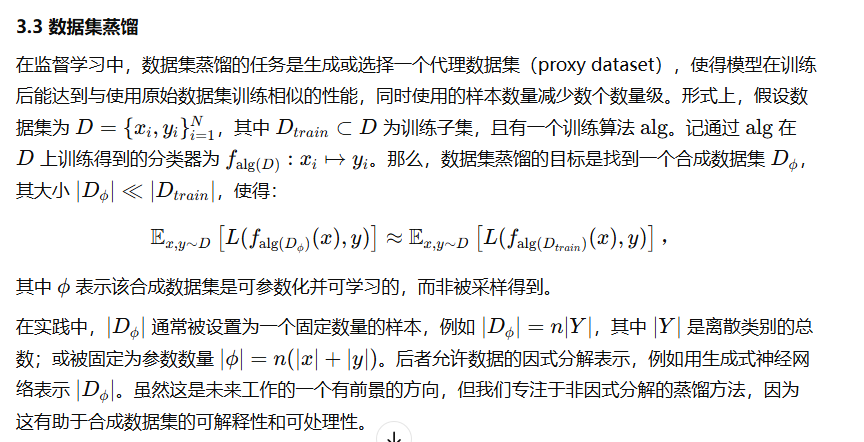
与直接进化神经网络参数不同，间接编码通过进化“基因型”的另一种表示（通常是压缩的），将其映射到网络参数上。例如，HyperNEAT（Stanley et al., 2009）是 HyperNetworks（Ha et al., 2016）的前身，它通过进化较小的神经网络来生成较大网络的权重。间接编码的优点在于，进化策略在参数数量增加时往往扩展性较差（Hansen, 2016）。我们的工作并非进化神经网络，而是进化一个小型数据集，然后用该数据集对较大的神经网络进行监督学习训练。

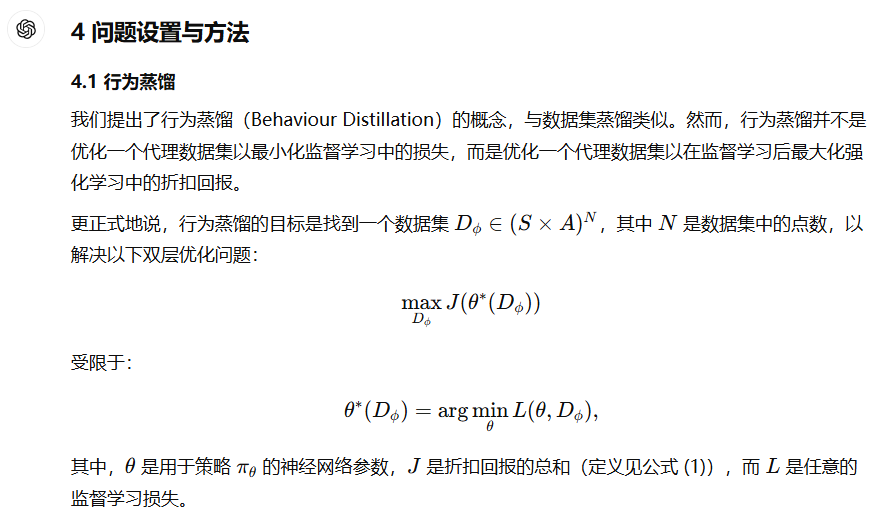
其他相关工作也进化了强化学习训练中的其他方面。例如，有些工作进化了强化学习策略目标（Lu et al., 2022; Co-Reyes et al., 2021; Houthooft et al., 2018; Jackson et al., 2024）和环境特征（Lu et al., 2023）。与我们工作最相关的是“合成环境”（Synthetic Environments, Ferreira et al., 2022），该方法进化神经网络以替代环境的状态动力学和奖励，从而加速训练。而我们的工作不是进化转换函数和奖励函数并用强化学习训练，而是进化用于行为克隆（Behavioural Cloning, BC）的监督数据。这显著提高了解释性并简化了内循环。

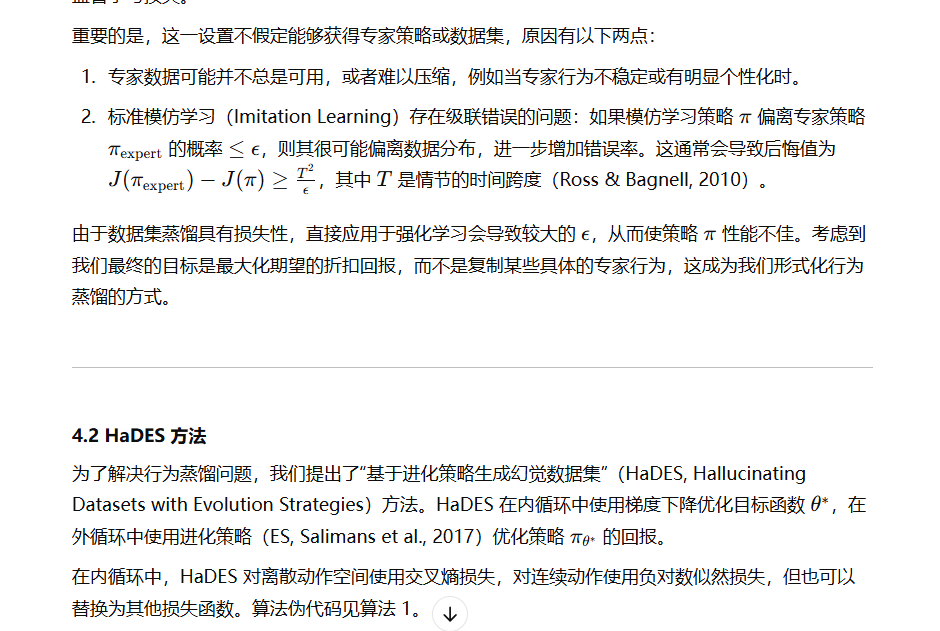


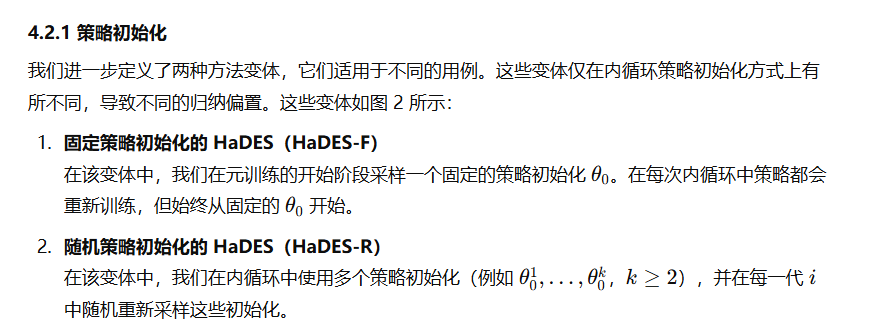


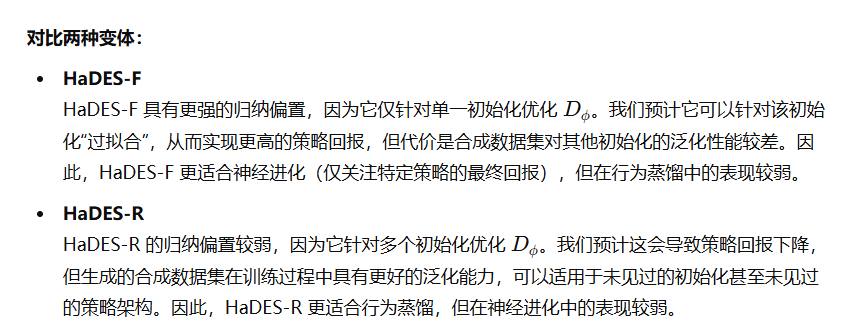












### 图 2: 左：标准神经进化。中：HaDES-F。右：HaDES-R。HaDES-F 使用单一固定的策略初始化，HaDES-R 每代随机采样 k≥2k \geq 2 个策略初始化。

### 5 实验

接下来，我们展示 HaDES 在广泛设置下的成功应用。具体而言，我们研究以下四个应用场景：

1. 我们验证 HaDES 作为神经进化中间编码在强化学习任务中的有效性（第 5.1 节），并在附录 C.1 中分析蒸馏预算。
2. 我们展示了合成数据集能够在广泛的架构和超参数中成功泛化（第 5.2 节）。
3. 我们展示了合成数据集可以用于训练多任务模型（第 5.3 节）。
4. 我们证明，尽管我们的重点是强化学习，HaDES 在监督学习中的数据集蒸馏任务中也具有竞争力（第 5.4 节）。

实验细节（包括运行时间比较）见附录 B.2。

#### 5.1 HaDES 蒸馏强化学习数据集的性能评估

我们首先测试 HaDES 在训练强化学习竞争性策略方面的有效性，实验涉及 Brax 套件中的 8 个连续控制环境。图 3a 显示了以下三种方法的性能：固定策略初始化的 HaDES（HaDES-F）、两种随机初始化的 HaDES（HaDES-R）以及通过进化策略（ES）直接进行的神经进化。

实验结果表明，HaDES-F 在所有环境中都实现了最高的回报，而 HaDES-R 在 6/8 个环境中也与基线相匹配或超过了基线。在 Humanoid-Standup 环境中，我们的方法发现了碰撞物理中的一个漏洞，使得其能够将自身弹射到空中，从而获得极高的回报2。

在 MinAtar 环境中，我们发现 HaDES-F 在 Breakout 中超越了 ES 基线，并在 SpaceInvaders 和 Freeway 中与其持平。我们推测，MinAtar 环境的符号化（而非连续性）特性使得该方法更难以应用。对于这两种设置，我们还绘制了在进行 5×1075 \times 10^7 步训练后 PPO 策略的性能。尽管 ES 和 RL 之间仍存在性能差距，HaDES 显著缩小了这一差距。

对于 HaDES 的策略网络，我们使用宽度为 512 且种群规模为 2048 的网络，但由于内存限制，我们不得不在 MinAtar 环境中将 ES 基线的网络宽度减半。事实上，当生成新种群或估计每一代的元梯度时，ES 需要将整个种群分配在单个设备上。尽管分布式方法可以用于外循环计算，但这会增加工程实现的复杂性，而我们的方法能够减轻这一负担。因此，HaDES 大幅提高了内存效率，允许使用更大的种群和网络规模，并且仅需最小的代码修改。为了全面性，我们也使用宽度为 256 的 HaDES 进行实验。

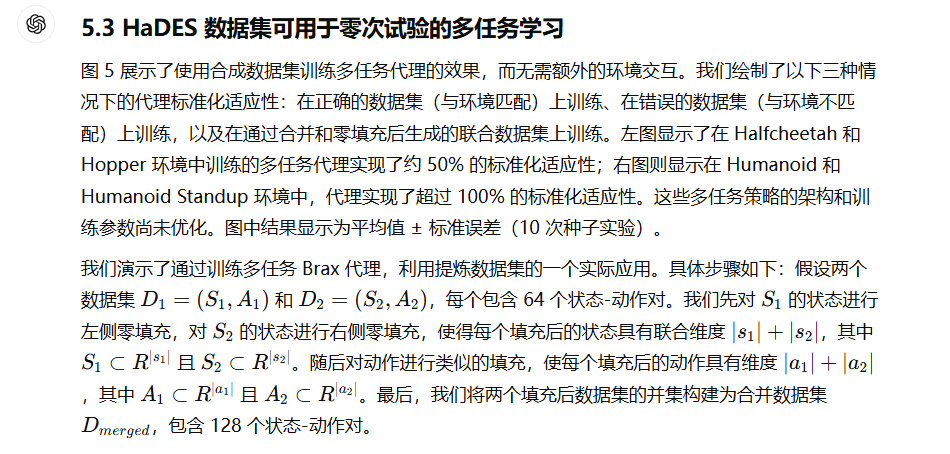
### 5.2 HaDES 数据集在不同架构和超参数中的泛化能力

接下来，我们将注意力转向数据集本身，使用它们以零次试验的方式训练新策略，即无需额外的环境交互。我们选择了两个针对 Hopper 生成的合成数据集——一个由 HaDES-R 生成，另一个由 HaDES-F 生成——并使用它们从零开始训练新策略。

对于每个数据集以及 7 种不同的策略网络规模，我们训练了 50 个策略，每个策略使用不同的学习率和训练轮次。具体而言，学习率跨度为 3 个数量级，训练轮次在 100 到 500 之间均匀分布。对于每个数据集和网络宽度，我们舍弃表现最差的 25 个策略，并绘制剩余 25 个策略的回报分布（如图 4 所示）。

尽管这些数据集是为宽度为 512、固定学习率和固定训练轮次的策略而设计的，它们能够很自然地泛化到不同设置和架构的训练中。特别是，我们观察到 HaDES-R 数据集对策略架构和训练参数变化的鲁棒性优于 HaDES-F 数据集，因为后者引入了更强的归纳偏置。

我们推测，合成数据集的泛化性能可以通过随机化更多的因素（如 HaDES-R 中的策略初始化，以及架构和训练参数）进一步提高，从而减少我们实现中存在的一些归纳偏置。如何在泛化能力、数据集大小和策略性能之间进行最佳权衡，仍是未来研究中一个有趣的问题。



我们在两个环境对上进行了实验：（Halfcheetah, Hopper）和（Humanoid, Humanoid Standup）。蓝色表示在 DiD\_iDi​ 上训练并在环境 iii 上评估的基线性能。红色表示在 DiD\_iDi​ 上训练并在环境 −i-i−i 上评估的性能，粗略地反映了一个环境的数据对另一个环境策略学习的帮助程度。绿色表示在 DmergedD\_{merged}Dmerged​ 上训练的策略的性能。我们观察到，在第一个环境对中，多任务代理实现了单任务性能的约 50%，但在第二个环境对中没有性能损失。

这表明由 HaDES 演化的合成数据集能够显著加速强化学习基础模型的未来研究。事实上，基于这些数据集训练新模型仅需几秒钟，从而以原始计算成本的一小部分进行架构和多任务表征学习的实验。此外，这种方法可以将跨任务参数共享和表征学习的研究与强化学习内在的探索问题分离开来，单独研究其特性。

**5.4 HaDES 可应用于监督数据集提炼**

虽然我们的重点在于强化学习中的行为提炼，但通过用目标数据集的交叉熵损失替代环境回报，我们的方法也可以直接应用于标准数据集提炼场景。我们在 MNIST 和 FashionMNIST 上以每类 1 张图片的条件进行了数据集提炼实验，结果见表 1。我们使用训练集上的交叉熵损失作为 HaDES 的适应性指标，并报告在最终合成数据集上训练的分类器在测试集上的平均准确率。我们运行了 3 次不同的种子实验，并对每个最终数据集训练了 20 个分类器。与 RFAD（目前在这些数据集提炼设置中表现最佳的方法）对比，我们报告了 60 个最终分类器的均值和标准差。

实验表明，HaDES 在 MNIST 上具有竞争力，在 FashionMNIST 上以 10 张图片达到了最先进的提炼效果。然而，我们未能将方法扩展到 CIFAR-10 数据集，因为其 RGB 图像需要处理更多参数，而非灰度图像。

在调优我们的方法时，我们发现最重要的两个参数是外层学习率和数据集初始化。我们发现以类别均值为数据集的初始化效果最佳，这与 Zhao 和 Bilen 的发现一致，他们也认为暖启动数据集优于从零开始初始化。

**5.5 HaDES 可解释性**

HaDES 数据集的最终优势在于，合成的示例有助于提高可解释性。回到图 1，我们可以看到，生成的数据集具有直观的特性。例如，针对 Cartpole 的双状态数据集，展示了当杆子向左倾斜时，策略应该向左走，而在其他情况下应该向右走。

尽管进行可解释的强化学习研究超出了本文的范围，但在批判性分析中，Atrey 等人强调了采取假设驱动的方法来解释深度强化学习策略的重要性，即先提出可能的解释，然后通过消融实验和仔细实验进行严格验证。考虑到这一点，我们认为，我们的合成数据集是进行这种假设验证的有效起点。例如，可以通过对数据集进行变换并观察其对训练策略的影响来进行测试。

**6 讨论与结论**

在本文中，我们介绍了一种新的策略神经进化参数化方法，通过演化一个小的合成数据集并通过行为克隆训练策略。我们展示了该方法在连续控制和离散任务中生成具有竞争力回报的策略。我们的方法可用于行为提炼，将环境中最优行为的所有相关信息总结为一个小的合成数据集。这种“行为软盘”可以快速训练不同架构参数化的新策略。接着，我们展示了通过训练多任务模型来利用提炼数据集的有效性。最后，我们表明，尽管我们的方法主要关注于强化学习，但它同样适用于监督学习中的常规数据集提炼，并且在某些设置中实现了最先进的结果。

本研究的主要局限性在于计算方面，因为进化方法需要一个大规模的种群才能有效。此外，虽然我们的替代参数化方法使我们能够演化比标准神经进化方法更大的神经网络，但参数数量仍然会随着数据点数量的增加而线性增长，尤其是在像素级环境中，数据维度通常非常高。因此，解决这个问题，比如采用因式分解的提炼方法，是未来研究的一个有前景的方向。我们工作中的另一个缺点是超参数的数量，因为我们需要调整外层循环中的 ES 参数和内层循环中的监督学习参数。然而，经验表明，ES 可以适应内层循环的参数，例如当学习率较低时通过增加数据集的规模来调整。理解这些参数之间的相互作用将有助于更快速和更好的调整。一个相关的方法是与数据集一起演化内层循环的参数。最后，提炼数据集的潜在应用值得进一步探讨，例如在持续学习或终身学习中，或通过正则化数据集来进一步促进可解释性的提高。