arXiv:2303.01584v2 [cs.CV] 2 Aug 2023

自监督学习的进化式增强策略优化

预印本

Noah Barrett[1, 2, †], Zahra Sadeghi[1, 2, †, *], 以及 Stan Matwin[1,2]

[1]加拿大哈利法克斯达尔豪斯大学计算机科学学院[2]哈利法克斯大数据分析研究所，加拿大[*]通讯作者：zahras@dal.ca

摘要

自监督学习（SSL）是一种机器学习算法，无需人工标注数据即可对深度神经网络（DNN）进行预训练。这种学习技术的核心思想是设置一个辅助阶段，即前置任务，通过数据增强自动生成标注数据，并利用这些数据对深度神经网络进行预训练。然而，文献中对各前置任务的效果尚未进行充分研究或比较。本文在受限设置下研究增强算子对自监督学习算法性能的贡献。我们提出一种进化搜索方法，用于优化前置任务中的数据增强流水线，并衡量增强算子对若干最先进的自监督学习算法的影响。我们将增强算子的不同组合编码为染色体，并通过进化优化机制寻找最优增强策略。此外，我们还提出了分析和解释优化后自监督学习算法性能的方法。结果表明，我们提出的方法能够找到准确率优于自监督学习算法的分类方案，这证实了增强策略的选择会影响自监督学习算法的整体性能。我们还比较了通过进化搜索机制找到的最优自监督学习方案，并展示了前置任务中的批量大小对两个视觉数据集的影响。

关键词 自监督学习（自监督学习），数据增强，进化算法，深度学习，深度神经网络，优化，可解释性，人工智能

1 引言

机器学习算法的强大能力建立在监督学习技术之上。然而，尽管这些算法已被证明在解决分类问题方面非常成功，但它们仍完全依赖大量人工标注的数据。因此，整个学习过程并非自主进行，且容易受到标注错误的影响。为应对这些局限，自监督学习（SSL）应运而生，使深度神经网络能够在不依赖标注数据的情况下进行训练。自监督学习的核心是一个辅助阶段，即前置任务，其重点是利用自动生成的标签数据对网络进行预训练。前置任务的一个关键环节是使用增强算子变换未标注图像，以生成已标注样本。尽管已有多种自监督学习范式，但尚无专门研究评估它们的有效性和实用性。此外，也没有专门的研究探讨增强算子对各类已提出的自监督学习算法性能的影响。在本文中，

[†]这些作者对本研究贡献相同

我们展示了增强算子的选择会影响下游问题所能达到的性能。通过一系列实验，我们力求衡量不同增强策略的影响。我们提出一种进化搜索算法，以优化能够产生最佳结果的变换设置。为此，我们聚焦于四种在多个数据集上表现出高性能的最先进的自监督学习算法。本文的原创贡献如下。首先，我们通过一种进化学习算法，研究四种最先进的自监督学习算法中增强算子的最佳组合。然后，我们比较这些自监督学习方法在优化前后的性能。我们针对两个不同的数据集，找出并评估每种自监督学习方法的最佳增强算子。最后，我们对优化后的染色体进行解释性分析，以了解增强算子对每种自监督学习算法和数据集的影响及重要性。本文结构如下：第 2 节概述自监督学习、自动增强和搜索优化算法。第 3 节介绍实现细节。接着，第 4 节描述我们的方法。第 5 节给出实验及结果。第 6 节介绍可解释性方法。最后，第 7 节总结全文。

2 相关工作

本节分三个小节回顾相关的前期研究。首先，我们介绍自监督学习算法的范式，并回顾该领域正在开展的研究。其次，我们探讨自动增强的最新进展。最后，我们概述智能搜索算法。

2.1 自监督学习

自监督学习凭借比传统无监督学习更高效地处理未标记数据的能力，在机器学习领域受到越来越多的关注。这是通过设计一种称为前置任务的新学习阶段实现的，该任务旨在解决一个辅助分类任务。辅助任务侧重于将数据分类到自标记类别中，以获得能够丰富表征数据底层结构的预训练模型。人们已提出多种设计前置任务的方法，可分为基于图像块、基于实例和基于视频的方法。基于图像块的方法旨在学习图像中各图像块之间的相对位置 Doersch et al. (2015)；基于实例的方法则利用整幅图像，并借助仿射变换 Doersch et al. (2015) 或着色 Isola and Efros (2016) 等增强任务。相比之下，基于视频的方法设计的任务能够利用视频片段中帧的顺序 Misra et al. (2016)。自监督学习也可分为生成式模型和对比式模型。生成建模方法旨在通过使用自编码器 Pathak et al. (2016) 和生成对抗网络 Donahue et al. (2016) 重建输入；对比方法则通过应用对比学习，尝试增大不同图像之间的差异，并缩小相似图像之间的差异 Oord et al. (2018)。自监督学习与对比式模型密切相关，而生成式方法通常被视为一种无监督学习方案。人们提出了多种不同的方法来解决前置任务，这些方法高度侧重于不同形式的数据增强，例如空间位置和图像块的相对位置 Noroozi and Favaro (2016)Dosovitskiy et al. (2014)、包括旋转在内的图像变换 Gidaris et al. (2018)、在保留局部统计量的同时改变全局统计量 Jenni et al. (2020)，以及数据增强、颜色抖动、绘制和着色 Isola and Efros (2016)。鉴于这些前置任务本质上各不相同，探究前置任务预训练过程的影响，以期发现下游任务中的性能差异，十分值得关注。本文重点比较四种最先进的自监督学习算法，即 BYOL gri (2020)、simsiam Chen and He (2021)、NNCLR Dwibedi et al. (2021) 和 SWAV **?**。所有这些自监督学习算法在网络架构和底层方法上都有基本相似之处。具体而言，它们的核心都采用了孪生架构。孪生架构由一个共享输入网络组成，该网络连接到两个彼此独立但共享权重的神经网络，其整体结构类似于自然界中的孪生体。传统上，孪生网络用于图像任务，其中输入为不同图像，图像之间的实际可比性则通过监督方式确定 Chen and He (2021)。

BYOL、SimSiam、NNCLR 和 SwAV 都基于使用孪生架构进行对比学习这一总体理念，但实现细节和理论动机各不相同，因此四种算法之间存在显著差异。值得注意的是，这四种算法的作者都在不同程度上强调了算法对数据增强的不变性。本文探讨这些发现的局限，并指出，尽管所提出的算法整体上对增强策略表现出惊人的不变性，但要理解增强对对比学习的影响，仍有改进空间。BYOL 的核心思想是：给定一种称为目标表示的表示，可以训练出一种增强后的表示，即在线表示。BYOL 通过迭代方式，使用在线网络的缓慢移动平均值作为目标网络，从而改进所学到的表示 gri (2020)。BYOL 不需要负样本，并声称对批量大小和所用的数据增强类型不敏感。BYOL 无需大批量、负样本对，且对数据增强具有新颖的鲁棒性；正是这些特性使其在发布时成为一项突破性的算法。BYOL 对增强的鲁棒性是相较于最早期的对比式自监督学习方法之一 SimCLR **?**而言的；研究表明，仅使用随机裁剪时，BYOL 的性能下降幅度远小于 SimCLR。SimSiam 是一种追求简洁性的自监督学习方法，突显了孪生网络架构对于自监督学习方法的重要性。SimSiam 表明，共享权重配置是许多核心自监督学习算法的关键，而且可以使用简单的停止梯度方法来防止坍塌 Chen and He (2021)；坍塌是许多采用孪生网络的算法面临的常见问题。当模型将每幅图像都映射到同一个点时，就会发生坍塌：比较时相似度达到最大，但模型反而没有学到与当前问题相关的重要信息。SimSiam 提供了令人意外的证据，表明无需使用负样本对、大批量和动量编码器，也能学到有意义的表征。前两者在 BYOL 先前的研究中已被证明并非必需，但动量编码器曾被认为是避免坍塌解的关键组成部分。或许 SimSiam 最引人注目的发现是，只需一个简单的停止梯度操作，就足以成功训练孪生架构。与 BYOL 和 SimSiam 相比，SwAV 提供了一种新颖的学习机制。遵循对比学习的原始方法，BYOL 和 SimSiam 通过预测样本两个视图之间的“接近程度”来学习；这是孪生网络及其在表示学习领域的重要发现的核心概念。简而言之，SwAV 通过对图像的一个增强版本计算编码，然后从同一图像的其他增强版本预测该编码来学习。该方法在对数据进行聚类的同时，确保不同视图之间的聚类分配保持一致 **?**。此外，SwAV 引入了一种新颖的多重裁剪增强方式，通过使用少量大裁剪窗口和更多小裁剪窗口，实现更加节省内存的比较。这意味着，尽管采用了孪生架构，SwAV 实际上并不属于对比学习。然而，SwAV 的核心目标仍是学习与当前问题相关的强大潜在表示。这种替代方法带来了 BYOL 和 SimSiam 未能实现的一项新突破，即能够大幅减少成对特征比较的数量，从而降低算法的计算和内存需求。与上文讨论的其他自监督学习方法相比，NNCLR 在正样本对的选择方面引入了一个非常新颖的概念。它并非通过数据增强为图像生成正样本，而是表明可以将给定数据空间中目标图像的最近邻用作正样本。与 BYOL 和 SimSiam 一样，NNCLR 是一种对比学习方法，并且与 SimSiam 类似，采用了非常简单的网络架构。它的新颖之处在于从给定数据集中选择正样本，而不是通过数据增强生成正样本。NNCLR 通过在数据集学习到的潜在空间中计算给定样本的最近邻来采样正样本。与预先定义的变换相比，这种方法带来了更多语义类别变化，而预定义变换往往提供更多几何信息 **?**。与上文讨论的所有方法一样，NNCLR 也发现自己并不十分依赖数据增强。

讨论的这四种算法都采用了孪生架构；可以将 SimSiam 看作是使用孪生架构的这些学习方法中最简单的一种。BYOL 通过在孪生网络的目标组件中采用慢速移动平均方法，为学习方法增添了一定的复杂性。与 SimSiam 一样，NNCLR 也选择了简单的网络架构；然而，它并非仅依靠增强来生成

正样本，而是确定给定图像的最近邻，并据此选择正样本。SwAV 与其他三种方法的不同之处在于，它使用交换预测机制进行学习，而非典型的对比机制。这四种算法近期都在自监督计算机视觉领域取得了显著突破，因此我们选择它们作为本研究的重点。这四种算法都声称，根据实证结果，它们对数据增强具有不同程度的不变性。然而，BYOL、SwAV 和 SimSiam 的对比机制完全依赖数据增强。由于关于数据增强效果的讨论有限，其对不同自监督学习算法的影响仍是一个悬而未决的问题。在本研究中，我们假设，通过选择最佳的增强算子组合，可以提升自监督学习算法的性能。

2.2 自动增强

数据增强是一种正则化技术，主要用于通过增加训练样本的多样性，解决训练深度神经网络时的过拟合问题（Shorten 和 Khoshgoftaar，2019）。近期，这种方法被用于对比式自监督学习流程，在保留观测语义内容的同时生成额外的正样本（Von Kügelgen 等人，2021）。如上一节所述，在训练过程中，正样本会与负样本区分开来，从而获得可用于学习不同任务的丰富嵌入。尽管数据增强是自监督学习的核心基础，但人们很少关注其影响，也鲜有研究探讨增强任务在理论上的有效性。据我们所知，目前尚无系统研究评估在自监督学习中采用不同增强策略的效果。一些作者指出，变换算子对自监督特征学习的表现十分重要。Jenni 等人认为，变换函数的选择因数据而异，因此不同数据集可能适用不同的变换集合（Jenni 等人，2020）。此外，不同任务可能需要量身定制的增强技术。例如，Li 等人（2021）针对异常检测提出了一种新颖的代理任务。随着旨在实现机器学习流程自动化的自动机器学习日益发展，人们也开发了自动增强方法，以寻找训练深度神经网络的最优增强策略，从而提升其性能。AutoAugment 可视为设计自动数据增强学习流程的先驱方法之一（Cubuk 等人，2019）。该方法采用强化学习来探索增强策略的搜索空间。Lim 等人（2019）通过匹配训练数据与验证数据之间的密度，加速了自动增强流程。PBA（Ho 等人，2019）是另一种方法，它使用超参数搜索算法来学习增强策略的调度方案。

2.3 搜索优化算法

搜索优化算法是机器学习技术取得巨大成功的核心组成部分之一，并且越来越多地被应用于参数调优、超参数调优和模型选择等基础问题中（Brazdil 等人（2022））。优化算法旨在通过最小化成本函数来调整机器学习算法的配置，从而实现准确的估计和预测。通过迭代搜索算法，可以获得最能描述可观测数据分布的最佳配置。机器学习任务中采用了多种搜索优化技术（Yang 和 Shami（2020））。其中最简单的一种称为穷举搜索，它根据期望的性能水平或可用计算资源，通过检查尽可能多的解来寻找最佳参数。这类方法中有两个广为人知的子类别：随机搜索随机探索搜索空间；网格搜索则系统地离散化搜索空间，并检查落在网格中的解。为实现更有效的搜索，人们还开发了其他更智能的方法，包括基于梯度的方法、强化学习和进化算法。基于梯度的方法是迭代搜索算法，尝试根据每一步的梯度值寻找最优解。这些算法通常会陷入局部最优点。另一方面，强化学习算法是一种基于以下内容遍历解空间的搜索范式：

表1：所使用的所有增强算子及其对应的强度范围。这组算子与 AutoAugment 相同（Cubuk 等人（2019））。

增强	强度范围
水平翻转	0.0, 1.0
垂直翻转	0.0, 1.0
X轴剪切	0.0, 0.3
Y轴剪切	0.0, 0.3
水平平移	0, 14
垂直平移	0, 14
旋转	-30, 30
颜色	0.1, 1.9
曝光反转	0.0, 1.0
对比度	0.1, 1.9
锐度	0.1, 1.9
亮度	0.1, 1.9

从预定义列表中采取行动。每个行动都会改变智能体的状态并产生奖励。目标是最大化回报或折扣奖励之和。进化优化是指仿生方法：通过迭代创建、进化并评估一个解群体。三种常见的进化算法变体是遗传算法（GA）、遗传编程（GP）和进化策略（ES）（Chiong 等（2012））。进化算法因其多种独特属性而受到研究者关注。它们能够以并行且独立的方式搜索空间，不依赖梯度计算，因此不会受到梯度爆炸或梯度消失的影响（Such 等（2017））。鉴于这些引人注目的特性，本文采用了一种遗传算法，具体内容将在下一节中介绍。我们遵循遗传算法的基本执行步骤。遗传算法的各个组成部分取决于具体问题，需要依据实验进行调整并适配任务（Bäck 等（2000），Eiben 等（1999））。每种遗传算法算子和组成部分都有多种定义方式。染色体可以采用二进制或数值表示进行编码。选择算法可以是轮盘赌、排名选择或锦标赛。交叉可以采用单点交叉、k点交叉或部分映射交叉。变异可以采用倒置变异或逆序变异（Katoch 等（2021））。常见做法是根据具体应用调整所有这些参数。在我们的实验中，我们采用参数调优策略来选择算子，并对交叉率和变异率实施参数控制。遗传算法有多个变体，例如混合遗传算法、并行遗传算法和混沌遗传算法（Katoch 等（2021））。本研究的目标并非考察和比较这些变体，而是利用经典遗传算法作为一种元启发式优化方法，来调优自监督学习算法的算子。我们在全监督、默认自监督学习以及遗传算法改进三种不同设置下，对四种自监督学习算法进行了比较。

3 实现细节

本研究旨在探索常见自监督学习算法前置任务中大量不同的数据增强配置。为此，我们采用了两个常用于图像分类的标准基准数据集，并使用 32 和 256 两种不同的批量大小，评估四种自监督学习算法的性能。实验在 CIFAR-10 和 SVHN 数据集上进行。以下小节将详细介绍各项实验的结果。

3.1 数据集

CIFAR-10。CIFAR-10 数据集 Krizhevsky（2009）是一个广受欢迎的彩色图像数据集，包含飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车这 10 个类别。该数据集包含 60000 张 32x32 图像，其中有 50000 张训练图像和 10000 张测试图像。测试集由每个类别中随机抽取的 1000 个样本组成

，其余 50000 个样本用于训练集。该数据集中的类别彼此互斥，创建者还特别注意确保卡车和汽车这两个相似类别之间没有重叠。

SVHN。街景门牌号（SVHN）数据集 Netzer et al. (2011) 是一个图像数据集，由从 Google 街景图像中拍摄的真实世界门牌号图像组成。该数据集被视为 MNIST Deng (2012) 的更具挑战性的版本，包含 600,000 个 32x32 的带标签数字，这些数字是从街景图像中裁剪得到的。SVHN 中的分类问题明显比 MNIST 更难，因为它包含真实世界中的数字图像。

3.2 网络架构与训练规范

在所有实验中，均使用一个小型且简单的卷积神经网络作为预训练和下游任务的骨干网络。如图 1 所示，该网络由 3 个卷积块组成。每个卷积块包含两个卷积层，卷积核大小为 3x3，后接 ReLU 激活函数、最大池化和批归一化操作。对于每种自监督学习算法，即 BYOL、NNCLR、SimSiam 和 SwaV，均根据相应算法的具体要求应用这一骨干网络。每种自监督学习算法进行网络训练时使用的超参数均遵循原论文设置。对于下游分类任务，所有实验均使用 Adam 优化器，学习率为 0.001，权重衰减为 0.0005。损失函数采用交叉熵。此外，在预训练网络之上使用一个带 ReLU 激活函数的三层线性模型。预文本任务和下游任务均训练 10 轮。首先使用四种自监督学习算法之一对骨干网络进行预训练，然后将线性头添加到网络中，以便在下游任务上进行微调。每项实验大致使用 4GB GPU 显存和 12 个 CPU。

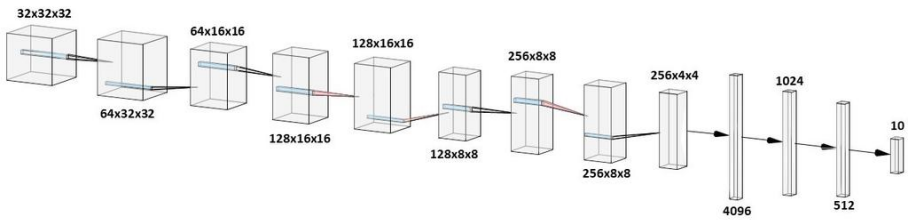


图 1：下游任务所用网络的架构。

本研究的实现依赖若干不同的库。对于深度学习组件，例如模型架构、数据集加载和监督训练，使用 PyTorch（Paszke 等，2019）。基于 PyTorch，Python 库 Lightly（Susmelj 等，2020）实现了先进的自监督学习算法；本研究使用该库实现了四种自监督学习算法。为实现本研究中的进化机制，使用了 Python 库 DEAP（Fortin 等，2012）。

4 方法

我们提出了两种通过遗传算法进行增强算子优化的进化机制，以寻找表现最佳的染色体及其对应的模型。第一种方法是单优化方法，其中四种自监督学习算法各自的增强策略分别进行优化；第二种方法是多优化方法，不仅优化增强策略，还通过寻找表现最佳的算法来优化下游任务性能。我们采用遗传算法，为四种不同的前沿自监督学习算法优化增强策略。从遗传算法的角度来看，最需要关注的关键环节是适应度函数的定义。所提出的适应度函数充当代理函数，旨在衡量针对给定自监督学习算法改变增强策略所产生的影响。下面将详细阐述适应度函数的形式化定义。为了将增强策略表示为

染色体，我们采用数值表示。我们对两种不同形式的进化式增强优化进行了实验，即单一优化（SO）和多重优化（MO）模式。在 SO 模式下，我们选择一种特定的自监督学习算法，仅优化每个基因中呈现的增强算子；而在 MO 模式下，染色体中的一个基因指定自监督学习算法，并通过优化过程找出最佳算法。因此，SO 与 MO 的主要区别在于染色体结构，该结构将在第 4.1 节中详细阐述。

4.1 染色体定义与增强策略

受 AutoAugment 及此前增强策略学习研究的启发，所提出的数值编码假设存在一组 k 数据增强，即 $A = \{a_1, a_2, \ldots, a_k\}$。对于每种增强 a_i，都可以定义一个强度值i 。Cubuk 等人（2019）为每种增强算子的强度定义了一个取值范围。本研究沿用了这些范围。表1展示了实验中使用的所有不同算子及其可能的 i 取值。设 i_k 为分配给增强算子 a_k 的强度值， K 为单次优化（SO）中增强算子的最大数量，则我们将染色体中的增强策略表示为 $C_i = \{(a_1, i_1), (a_2, i_2) \ldots, (a_l, I_l)\}$，其中 $l \leq K$。本文研究的所有算法均采用 SimCLR ? 使用的增强策略。研究发现，单一变换不足以学习到良好的表征；组合多种数据增强会使任务更困难，但能显著改善学得的表征 ?。本研究将 l设为 3。理论上，增加增强算子的数量有可能进一步提升效果，但这会增加实验运行的计算成本。选择 3 个增强算子，旨在平衡计算负担、问题的简洁性以及学得表征的质量。在多次优化（MO）中，我们将染色体表示为 $C_i = (alg_j, \{((a_1, i_1), (a_2, i_2) \ldots, (a_l, i_l)\})$，其中 alg_j 是$\{SwAV, NNCLR, Simsiam, Byol\}$中的一种自监督学习算法。

4.2 适应度函数

所提出的适应度函数旨在根据下游任务评估自监督学习算法的增强策略。更具体地说，下游监督任务中的测试准确率将作为增强优化的适应度。需要注意的是，本文关注的是前置任务的增强策略，而监督下游任务使用固定的数据增强集合。适应度函数中最关键的部分是固定随机性。通过固定深度学习流程中的每个随机组件，可以以可靠的相对方式比较两种增强策略之间的差异。为了补偿固定深度学习流程随机组件所导致的偏差，我们使用 N 个不同的随机种子评估所提出的方法，用于神经网络初始化和数据打乱。每次实验所使用的种子值如表3所示。

4.3 选择

存在大量不同的选择方法，例如轮盘赌、锦标赛和排序 O'Neill（2009）。我们的方法采用一种简单且常见的轮盘赌方法。在这种方法中，每个染色体被选中的概率与其适应度成正比。

4.4 交叉

选择交叉操作时，必须谨慎处理，以防子代染色体中出现重复的数据增强。为解决这一问题，我们采用了 Reeves（2003）提出的部分映射交叉（PMX）。该方法可避免生成含有重复基因的染色体。在两个或多个染色体之间进行交叉时，防止基因重复的机制仅依据增强算子，而不考虑强度。两个染色体交叉时，共有基因会被完整复制，以保持与

父代基因相同的强度。在我们的多次优化方法中进行交叉时，PMX 应用于染色体中的增强策略部分，而自监督学习基因则以一定概率与另一条染色体上的对应基因互换。

4.5 变异

变异时，仅对增强算子的强度进行变异。为此，我们创建了一个自定义变异函数，称为 $MutGaussianChoice$。该变异算子以概率方式在特定算子的可接受强度范围内变异基因的强度值。强度会逐步增加或减少，增减幅度为强度值范围（如表1所示）除以增量值；增量值根据公式1计算。对于多次优化（MO）， $MutGaussianChoice$ 应用于染色体中的增强策略部分；对于自监督学习基因，则将其随机变异为另一种自监督学习算法。

$$increment = \frac{max(i_{range}) - min(i_{range})}{10} \tag{1}$$

4.5.1 自适应变异率

应用变异和交叉时，是否对染色体进行相应操作由变异率和交叉率以概率方式决定。受 Srinivas 和 Patnaik（1994）的启发，我们的方法采用自适应交叉率和变异率（即 p_c 和 p_m），如公式2和公式3所示:

$$\begin{aligned} p_c = (f_{max} - f')/(f_{max} - \bar{f}), f' \geq \bar{f} \\ p_c = 1, f' < \bar{f} \end{aligned} \tag{2}$$

$$\begin{aligned} p_m = (f_{max} - f)/(f_{max} - \bar{f}), f' \geq \bar{f} \\ p_m = 0.5, f < \bar{f} \end{aligned} \tag{3}$$

¯其中， f_{max} 和 f 分别表示种群中的最大适应度值和平均适应度值，而f' 表示待交叉或变异的两个染色体中适应度较高者的适应度。

5 实验

在这项工作中，我们旨在了解四种常见自监督学习算法在前置任务中使用的增强算子的影响。通过采用进化机制来演化增强策略，算法在一次运行中会训练数百个模型。为了提供比较基线，我们根据原论文中的增强算子，分别以监督学习和自监督学习的方式训练模型。这样，我们便能了解演化得到的数据增强流水线与纯监督学习方法，以及采用原始数据增强流水线的自监督学习方法相比表现如何。鉴于我们在受限设置下训练网络（即使用小型网络和最少的训练轮数），我们还开展了另一项实验，研究训练更多轮数时，表现最佳的演化增强流水线在前置任务和下游任务中的效果。在这项实验中，我们采用通过所提出的进化搜索机制找到的最优增强算子。

监督设置。训练监督基线模型时，我们使用与自监督学习实验中下游任务完全相同的超参数，在不进行预训练的情况下开展监督分类任务；也就是说，我们从头开始训练一个随机初始化的网络架构。该基线使我们能够比较通过自监督学习得到的表征与随机初始化的表征。在所有监督基线中，我们都使用下游任务中所述的相同配置。



(a) 批量大小 32, CIFAR-10

(b) 批量大小 256, CIFAR-10

图 2：在前置任务和下游任务中增加训练轮数的模型训练。横轴表示不同的自监督学习算法，色调表示前置任务所用的训练轮数（50、100 或 1000），纵轴表示数据增强流水线初始准确率的提升幅度。

自监督基线。为了解所学习到的增强流程与原始自监督学习算法相比如何，我们使用原始论文中采用的数据增强来训练模型。由于与原始论文相比，我们的具体训练配置在网络架构和训练轮数方面受到显著限制，这些基线有助于我们了解自监督学习算法在受限设置下的表现。训练这些基线时，使用与采用演化数据增强训练的算法完全相同的超参数，例如批量大小和训练轮数。这样，我们就能了解在我们的设置中改变数据增强，相对于原始增强配置会如何影响自监督学习算法。

进化式自监督学习。所提出的遗传算法采用下游测试准确率作为适应度函数的指标。在基于进化的搜索中，自监督学习配置的超参数设置与自监督学习基线完全相同，但前置任务中的数据增强除外。我们采取了预防措施，确保自监督学习流水线中的所有随机组件均由固定随机种子控制，从而保证流水线中唯一发生变化的部分是数据增强本身。默认的自监督学习配置以及使用进化所得增强策略的进化优化增强配置，与监督学习基线使用完全相同的下游配置。因此，通过基本的自监督学习和监督学习基线，我们能够比较采用进化所得增强策略进行自监督预训练后，模型表现相较于监督学习和自监督学习基线的提升情况。

批量大小的影响 我们研究了批量大小对前置任务的影响，因此在监督学习、自监督和进化自监督模式下，分别对批量大小为 32 和 256 的情况进行了实验。

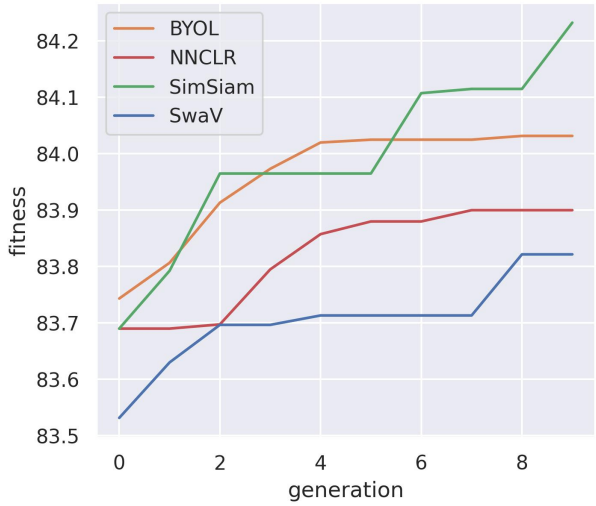
训练轮数数量的影响鉴于我们的实验是在受限条件下进行的，进化算法采用了较少的训练轮数。为了更好地评估自监督学习算法的性能，我们开展了一项实验：使用通过进化搜索机制找到的最佳增强策略，在前置任务中将自监督学习模型分别训练50、100和1000个训练轮数，并在下游任务中训练50个训练轮数。

5.1 结果

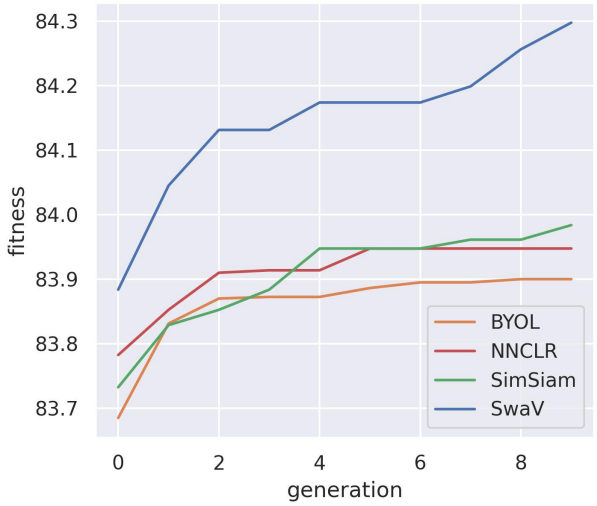
如前文所述，为了了解改变上述四种自监督学习算法前置任务中的增强算子所产生的影响，我们执行了进化算法。对于BYOL、SimSiam、SwaV和NNCLR这四种算法中的每一种，均使用了SVHN和CIFAR-10两个数据集，以及32和256的前置任务批次大小。为减轻随机性带来的偏差，使用了 N 个随机种子来控制算法中深度学习组件的所有随机性，其中 N 每项实验的数值见表3。这确保了训练结果的差异并非由训练流程中的随机性所致，例如模型初始化或数据打乱。对于每种自监督学习算法、批量大小和数据集，我们均采用了种群规模为15的进化优化增强算法。运行10代，种群规模为15，并针对表3所示数量的随机种子，总计有**15000**种不同的增强配置相互竞争。请注意，由于遗传算法的性质，个体会被保留至下一代，因此并非全部15000种配置都是唯一的。

我们发现，应用我们提出的进化方法能够持续提升自监督学习算法的性能。图3展示了四种算法在各代中所有随机种子的最佳适应度平均值。具体而言，对于每个随机种子的实验，我们提取每一代的最佳适应度，然后计算所有随机种子的平均值。在 CIFAR-10 和 SVHN 上，我们都观察到优化总体呈单调趋势，这证实了我们提出的进化方法能够提升自监督学习算法的性能。总体而言，NNCLR 的净提升幅度最小，而 BYOL 的提升幅度最大。此外，我们比较了在两个数据集上，使用两种不同批量大小进行自监督学习前置任务时，进化过程最终一代中不同随机初始化所得最佳解的分类准确率分布。图4表明，批量大小为 256 时，下游任务分类准确率的提升幅度更大。此外，我们观察到，SVHN 数据集上四种算法的平均准确率大致相当。相比之下，在 CIFAR-10 上，两种批量大小下四种算法的平均准确率均存在明显差异。批量大小为 32 时，SimSiam 在 CIFAR-10 上、BYOL 在 SVHN 上的性能最高；批量大小为 256 时，四种算法的结果相近。

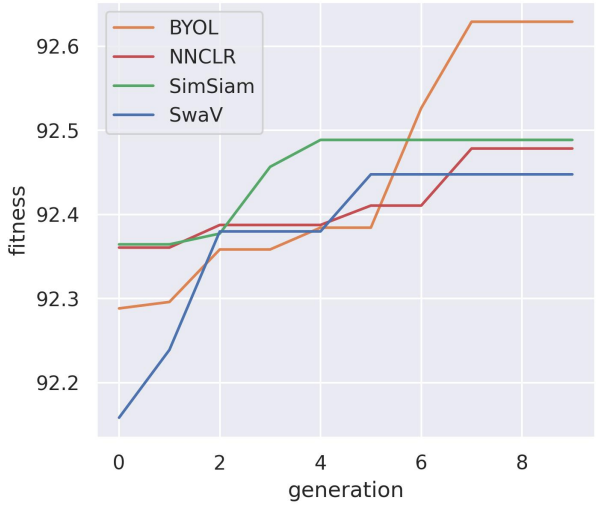
此外，表4、表5、表6和表7比较了四种算法各自取得的最优性能详情。结果表明，在两个数据集中，使用较大批量大小时通常能获得更高的准确率。此外，SO和MO找到的解均优于监督学习和自监督学习所得的结果。在所有情况下，包括两个数据集、两种批量大小以及四种自监督学习算法，SO算法找到的解都优于MO。这是因为单一优化过程专注于某种特定自监督学习方法的解，其搜索空间也更有限。更值得注意的是，在所有情况下，当我们采用SO找到的增强算子所对应的最佳配置，并在前置任务中训练50轮、在下游任务中训练100轮时，结果均优于监督训练。表中的SSL（优化后）即表示这些结果。为确保我们提出的策略始终优于基线，我们针对表8所列的自监督基线，以0.95和0.99的显著性水平进行统计学 t 检验，将其与我们的单一优化策略进行比较。结果显而易见，在所有情况下，自监督基线的表现差异在统计上均具有显著性，这证实了我们提出的增强策略优化方法能够显著提高自监督学习算法分类结果的准确率。如前所述，在所有实验中，我们对前置任务和下游任务均训练10轮。然而，为了更好地了解在我们的实验设置下，所提出的演化增强流程在更多训练轮数时的表现，我们又开展了一项实验，测量训练轮数对结果的影响。在该实验中，我们为每种自监督学习算法的相应前置任务采用其最佳增强流程，并分别训练50、100和1000轮。随后，在下游任务中对所得预训练模型进行50轮微调。通过这项实验，我们比较了前置阶段和下游阶段均训练10轮的模型，与前置任务训练50、100和1000轮、下游任务训练50轮的模型。如图2所示，我们发现性能提升幅度不断增大



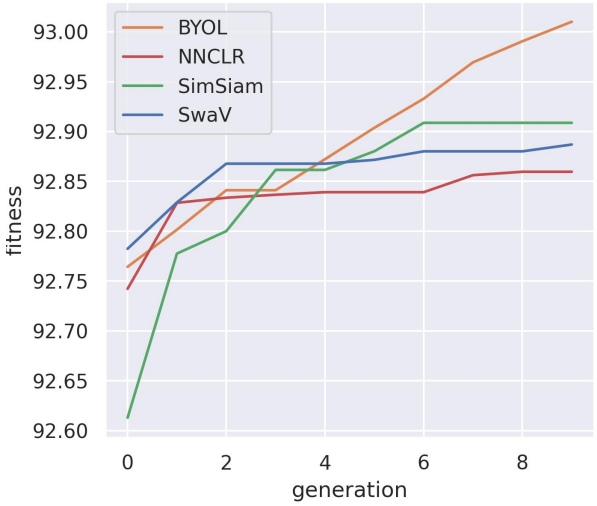
(a) CIFAR-10，批量大小



(b) CIFAR-10，批量大小

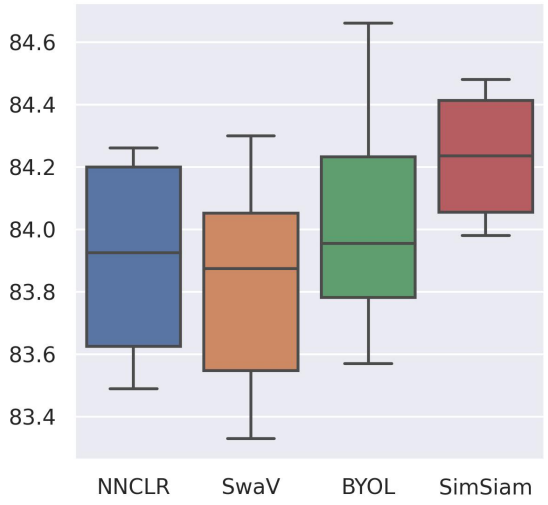


(c) SVHN，批量大小
32

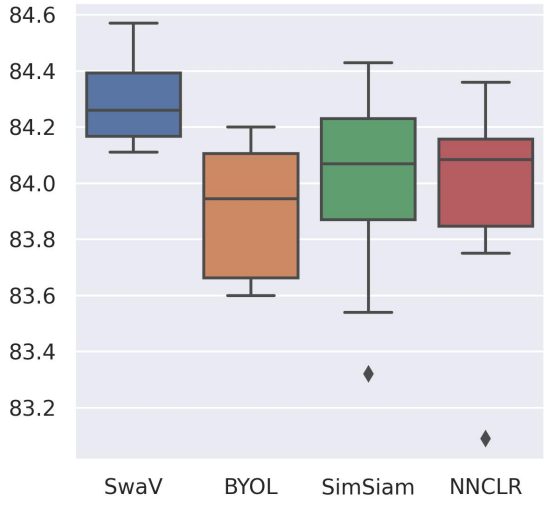


(d) SVHN，批量大小
256

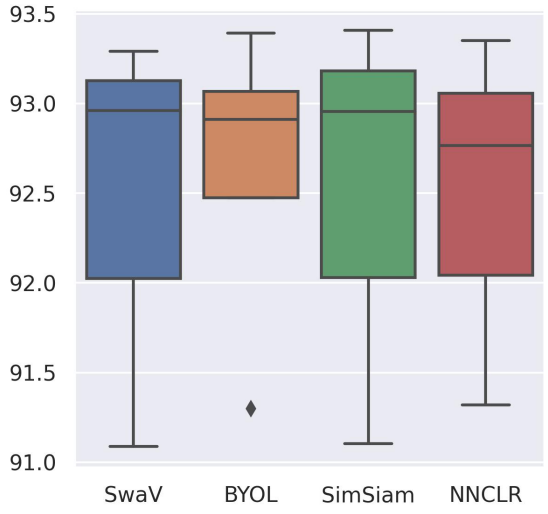
图3：对于表3中列出的所有随机种子，监督分类任务在每一代中通过 SO 策略找到的最佳下游测试准确率的平均值



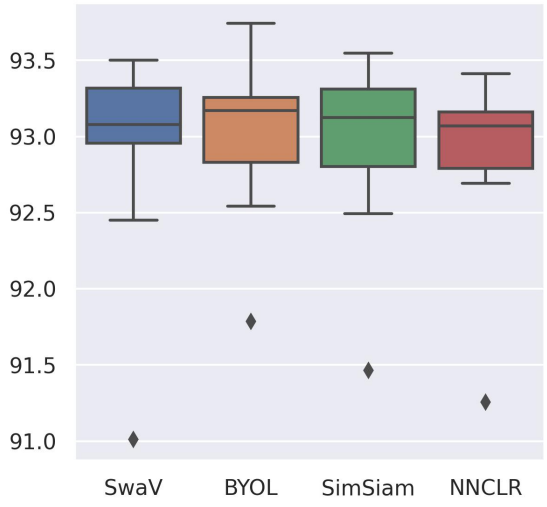
(a) CIFAR-10，批量大小



(b) CIFAR-10，批量大小



(c) SVHN，批量大小
32



(d) SVHN，批量大小
256

图4：使用SO策略时，不同随机种子下下游任务测试准确率的比较。所有使用的种子值均列于表3中

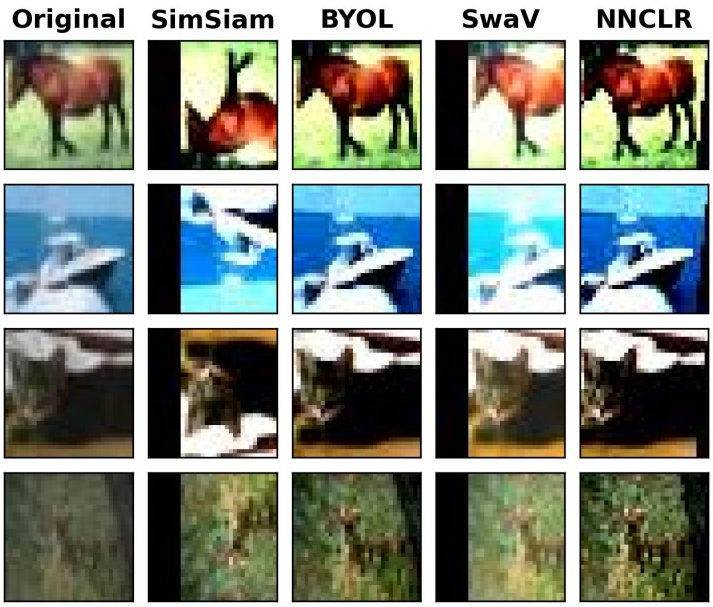


图5：CIFAR-10 上找到的最佳增强策略的可视化。这些策略见表2。每一行依次展示了增强算子应用于示例图像的效果。

表2：找到的最佳增强策略。aug_i 表示增强应用的顺序。op_i 表示相应的强度值

数据集	自监督学习算法	aug_1	op_1	aug_2	op_2	aug_3	op_3
svhn	SwaV	颜色	1.48	水平平移	3.00	X轴剪切	0.13
svhn	BYOL	锐度	0.95	对比度	1.28	曝光反转	0.32
svhn	SimSiam	对比度	1.15	水平平移	5.00	Y轴剪切	0.12
svhn	NNCLR	颜色	0.90	Y轴剪切	0.05	曝光反转	0.32
CIFAR-10	NNCLR	锐度	0.88	对比度	1.18	X轴剪切	0.10
CIFAR-10	SwaV	水平平移	8.00	亮度	0.69	颜色	0.50
CIFAR-10	BYOL	对比度	0.97	锐度	0.34	旋转	-1.00
CIFAR-10	SimSiam	水平平移	8.00	垂直翻转	0.64	对比度	1.24

在下游任务中，前置任务使用 50 和 100 个训练轮数时，准确率有所提升；但使用 1000 个训练轮数时，这种提升会减弱。这一发现表明，前置任务训练时间过长会导致过拟合，并使分类准确率下降。此外，我们在表2中总结了找到的最佳策略，并在图5和图6中分别通过几个示例展示了这些策略对 CIFAR-10 和 SVHN 数据集的影响。每项实验所用随机种子的数量详见表3。

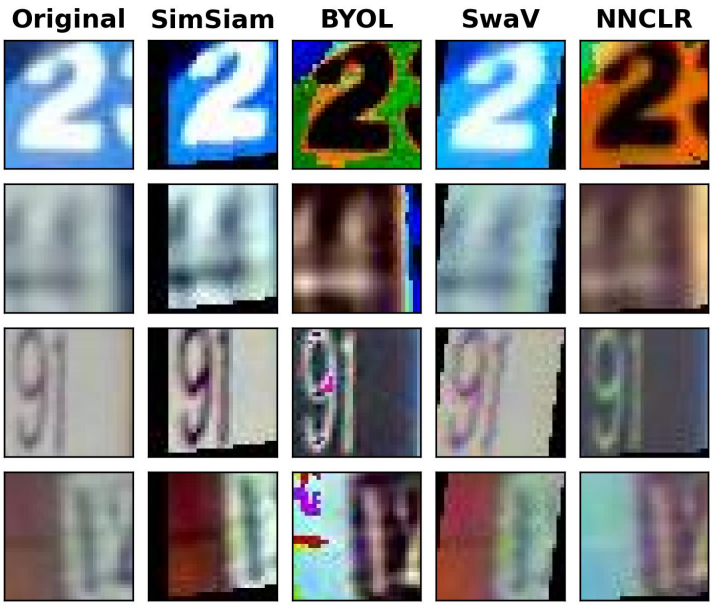


图6：SVHN最佳增强算子的可视化结果。相关策略见表2。每一行展示了按顺序将增强算子应用于示例图像后的效果。

表3：每个实验的种子值，包括预文本批量大小和数据集。Algo表示使用SO策略进行优化的自监督学习算法。

exp	algo	seed
bsize=256，data=SVHN	SwaV	[0, 1, 2, 3, 4, 5, 6, 7, 8]
bsize=256，data=SVHN	BYOL	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
bsize=256，data=SVHN	SimSiam	[0, 1, 2, 5, 6, 7, 8]
批量大小=256，数据=SVHN	NNCLR	[0, 1, 2, 3, 4, 5, 6, 7, 8]
批量大小=32，数据=SVHN	SwaV	[0, 1, 2]
批量大小=32，数据=SVHN	BYOL	[0, 1, 2, 3]
批量大小=32，数据=SVHN	SimSiam	[0, 1, 2]
批量大小=32，数据=SVHN	NNCLR	[0, 1, 2]
批量大小=32，数据=CIFAR-10	NNCLR	[0, 1, 2, 3]
批量大小=32，数据=CIFAR-10	SwaV	[0, 1, 2, 3, 4, 5]
批量大小=32，数据=CIFAR-10	BYOL	[0, 1, 2, 3, 4, 8]
批量大小=32，数据=CIFAR-10	SimSiam	[0, 1, 2, 3]
批量大小=256，数据=CIFAR-10	SwaV	[0, 1, 2, 3, 4, 5, 6, 7]
批量大小=256，数据=CIFAR-10	BYOL	[0, 1, 2, 3, 4, 5, 6, 7]
批量大小=256，数据=CIFAR-10	SimSiam	[0, 1, 2, 3, 4, 5, 6, 7]
批量大小=256，数据=CIFAR-10	NNCLR	[0, 1, 2, 3, 4, 5, 6, 7]

表 4：在前置任务中使用批量大小为 32 时，CIFAR-10 下游任务上找到的最佳准确率。SO：单次优化。MO：多次优化。SSL（优化后）：使用找到的最佳配置训练更长训练轮次时的最佳结果。粗斜体值表示 SO、MO 和两个基线中找到的最佳结果。粗体值表示对找到的最佳 SO 方法进行更长训练轮次后得到的改进结果。

algo	NNCLR	SimSiam	SwaV	BYOL
SSL（SO）	***84.26***	***84.48***	***84.30***	***84.27***
SSL（MO）	83.94	84.18	84.18	83.92
SSL（默认）	82.81	83.59	81.59	83.83
监督学习	83.24	83.24	83.24	83.24
自监督学习（已优化）	**85.58**	**85.72**	**84.74**	**86.03**

6 可解释性

本节提出两种算法，用于分析性能最佳的染色体中编码的最佳模型，并解释增强算子的作用。在下一节中，我们将介绍增强敏感度和增强重要性分析，以了解对每个数据集和自监督学习算法影响最大的算子。

6.1 增强敏感度

为了衡量每种算法对每个增强算子的敏感度，我们设计了一项消融研究，衡量移除各算子后性能的差异。具体而言，为了衡量 a_i 算子的敏感度，我们将其从所有包含该算子的染色体中移除，并用增强集合中的所有可能算子替换它，从而获得一组新的染色体。然后，计算这组染色体的平均性能与原始染色体性能之间的差值：OS(a_i) =性能(a_i) - average(性能(a_k))，对所有 k 均如此。伪代码见算法 1。

算法 1计算敏感性

$op \leftarrow$目标算子
$C \leftarrow$所有染色体的集合
$Cop \leftarrow C$ 中包含算子 op的染色体子集
$Sensitivity \leftarrow 0$
对于 c_i inCop **执行**
  $AvgSimAcc \leftarrow 0$
  $TotalSimilarChromos \leftarrow 0$
  对于 c_j inC **执行**
    $NumEqualOps \leftarrow 0$
    对于 op_i 在 c_i.操作 **执行**
      $NumEqualOps \leftarrow NumEqualOps + 1$
    结束循环
    如果 op 不在 c_j.操作中且 $NumEqualOps \equiv c_i$.长度 -1 **则**
      $AvgSimAcc \leftarrow AvgSimAcc + c_j$.准确率
      $TotalSimilarChromos \leftarrow TotalSimilarChromos + 1$
    结束条件
    $AvgSimAcc \leftarrow AvgSimAcc \div TotalSimilarChromos$
    $Sensitivity \leftarrow Sensitivity + |c_i$.准确率 $- AvgSimAcc|$
  循环结束
循环结束
$Sensitivity \leftarrow Sensitivity \div Cop$.长度

表5：在前置任务中使用批量大小为256时，CIFAR-10下游任务的最佳准确率。SO：单次优化。MO：多次优化。SSL（优化后）：使用最佳搜索到的配置进行更长训练轮次后的最佳结果。粗体斜体值表示SO、MO和两个基线所找到的最佳结果。粗体值表示采用最佳SO方法并延长训练轮次后取得的提升结果。

algo	NNCLR	SimSiam	SwaV	BYOL
SSL（SO）	***84.36***	***84.43***	***84.57***	***84.20***
SSL（MO）	84.24	83.85	83.97	84.20
SSL（默认）	83.86	83.23	83.22	84.01
监督学习	83.24	83.24	83.24	83.24
自监督学习（已优化）	**85.90**	**86.06**	**85.93**	**85.81**

表6：SVHN 在前置任务批量大小为 32 时，下游任务中找到的最佳准确率。SO：单次优化。MO：多次优化。自监督学习（已优化）：将找到的最佳配置训练更长轮次后取得的最佳结果。粗斜体值表示 SO、MO 和两个基线中找到的最佳结果。粗体值表示对找到的最佳 SO 方法进行更长轮次训练后取得的提升结果。

algo	NNCLR	SimSiam	SwaV	BYOL
自监督学习（SO）	***93.35***	***93.41***	***93.29***	***93.39***
自监督学习（MO）	93.07	93.07	93.07	92.70
自监督学习（默认）	92.34	92.52	91.88	92.85
监督学习	91.28	91.28	91.28	91.28
自监督学习（已优化）	**93.67**	**93.75**	**93.58**	**93.55**

表 7：在前置任务中批量大小为 256 时，SVHN 下游任务上找到的最佳准确率。SO：单次优化。MO：多次优化。自监督学习（已优化）：训练找到的最佳配置更长训练轮次后取得的最佳结果。粗斜体值表示 SO、MO 和两个基线中找到的最佳结果。粗体值表示对找到的最佳 SO 方法延长训练轮次后得到的改进结果。

algo	NNCLR	SimSiam	SwaV	BYOL
自监督学习（SO）	***93.41***	***93.55***	***93.50***	***93.37***
自监督学习（多次优化）	92.94	92.94	92.83	92.94
自监督学习（默认）	92.75	91.87	91.95	92.34
监督学习	83.24	83.24	83.24	83.24
自监督学习（优化后）	**93.70**	**93.68**	**93.63**	**93.67**

表8：所有单目标实验相对于自监督学习基线的 T 检验结果。

批量大小	数据集	algo	p值	95%置信区间	99%置信区间
32	CIFAR-10	BYOL	4.926368e-07	True	True
32	CIFAR-10	NNCLR	2.658124e-05	True	True
32	CIFAR-10	SimSiam	2.917474e-06	True	True
32	CIFAR-10	SwaV	2.240725e-06	True	True
32	svhn	BYOL	1.658260e-05	True	True
32	svhn	NNCLR	1.515886e-04	True	True
32	svhn	SimSiam	4.092746e-04	True	True
32	svhn	SwaV	4.330569e-04	True	True
256	CIFAR-10	BYOL	1.990317e-12	True	True
256	CIFAR-10	NNCLR	3.736046e-12	True	True
256	CIFAR-10	SimSiam	1.584149e-12	True	True
256	CIFAR-10	SwaV	1.235250e-15	True	True
256	svhn	BYOL	5.714354e-25	True	True
256	svhn	NNCLR	5.901998e-21	True	True
256	svhn	SimSiam	3.431059e-19	True	True
256	svhn	SwaV	1.401669e-18	True	True

6.2 增强重要性

在本分析中，对于每个算子和每种自监督学习方法，我们统计该算子在所有代中排名前50的染色体（依据测试准确率）中出现的次数。我们将该值除以算子数量和染色体数量进行归一化。算法2展示了这一过程的具体细节。该方法使我们能够找出增强算子在生成已发现的最佳染色体时的使用程度。借助这一指标，我们可以了解不同数据增强对排名靠前的染色体有何影响。

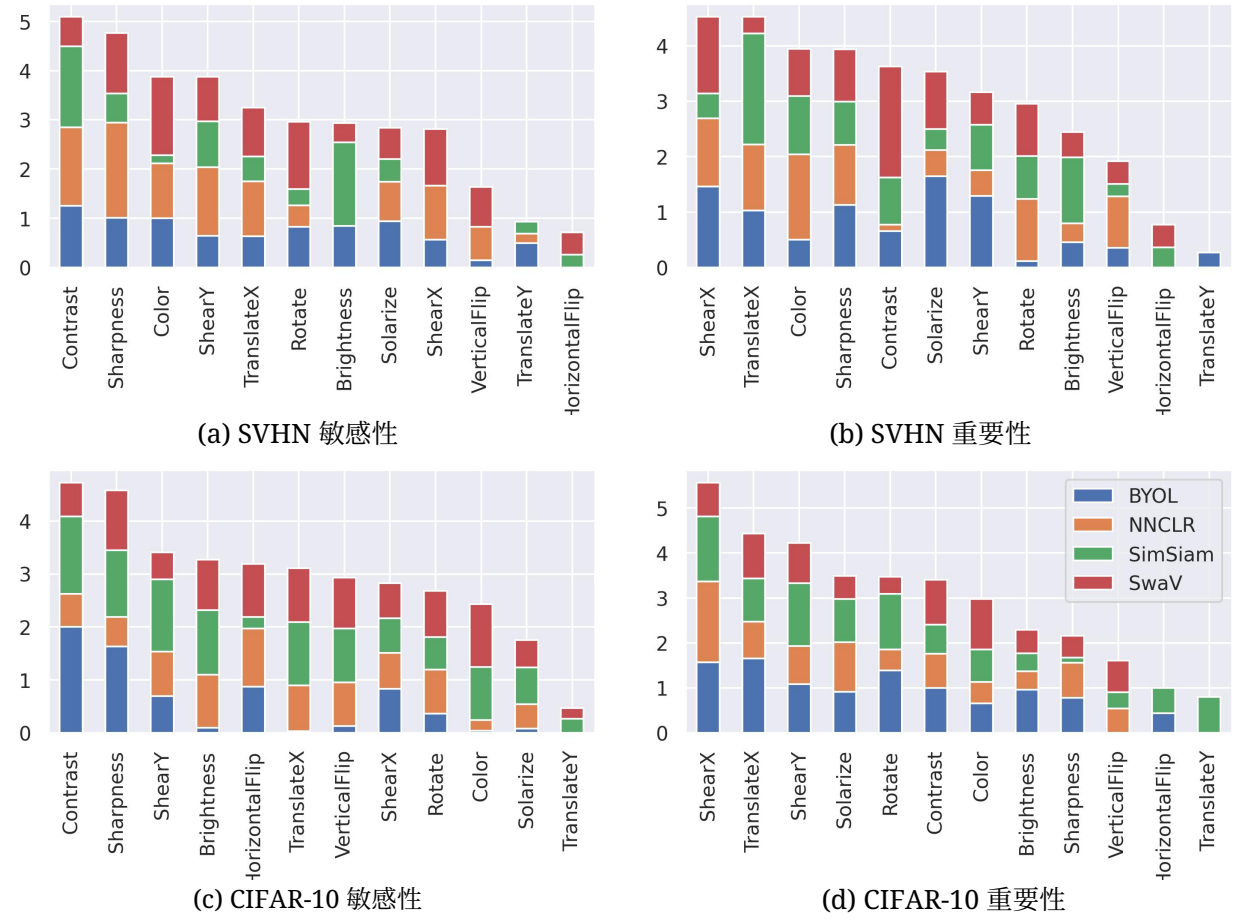
算法2计算算子重要性

$op \leftarrow$目标算子
$C \leftarrow$所有染色体的集合
$N \leftarrow$待考虑的染色体数量
$Importance \leftarrow 0$
$C \leftarrow$sorted(C)
对于 c_i 在 C 中[: N] **执行**
  如果 op 在 c_i.operations **则**
    $Importance \leftarrow Importance + 1$
  结束如果
结束循环

6.3 可解释性发现

图7展示了两种批量大小下增强重要性和敏感性的总体情况。从图中可以有趣地观察到，两种分析的趋势几乎相同，并揭示了有关各个算子的相似信息。请注意，重要性值反映了在最佳策略中使用频率较高的算子，而敏感性分析则考虑了对准确率影响更大的算子。至关重要的是，我们发现某些增强算子在整体上普遍存在，而其他算子则并非如此。此外，BYOL、NNCLR、SwAV 和 SimSiam 这四种算法的结果显示，各个算子的敏感性值和重要性值呈现出非均匀分布。这支持了这样一种观点：特定的数据增强会使自监督学习算法的前置任务更强。如果数据增强对结果没有影响，我们预计分布会更加均匀。在我们的结果中可以看到，所有情况下都有一种或多种数据增强占据主导地位，而其他增强则很少出现，甚至完全没有出现在条形图中。图7中的 a 图和 c 图对应敏感性分析，表明算法对对比度 和锐度 最敏感，对垂直平移最不敏感。两种翻转 操作的总体敏感性值分别位居倒数第二和倒数第三，这意味着所有算法似乎都不会对翻转 操作过度敏感。水平平移、颜色、亮度、旋转、曝光反转 以及两种剪切 算子的敏感性值均处于中等水平。这些发现表明，所有四种自监督学习算法对对比度 和锐度等非几何变换数据增强更敏感，而对平移和翻转等几何变换的敏感度较低。从堆叠条形图中还可以看出，某些算子在各个数据集中影响更大。例如，SimSiam 在 SVHN 中始终对亮度 算子敏感，在 CIFAR-10 中则对对比度 和Y轴剪切 敏感；SwAV 在 SVHN 中对颜色 和旋转 的敏感性相对较高，在 CIFAR-10 中则对颜色 、锐度 等若干算子敏感。BYOL 在 SVHN 和 CIFAR-10 中对不同增强算子的敏感性范围始终较广，而且该算法对对比度 和锐度 算子最敏感。对于 NNCLR，我们发现使用 SVHN 的实验对锐度最敏感。对于 CIFAR-10，模式并不那么明显，不过水平翻转显示出相对较高的敏感性。

图7中图组d的增强重要性条形图表明，对于两个数据集，X轴剪切和水平平移都主要出现在排名靠前的染色体中，而垂直翻转、水平翻转和垂直平移是应用最少的算子。对于 SVHN，其余增强在重要性方面的分布比 CIFAR-10 更为均匀。总体来看，对于两个数据集，在重要性最低和最高的增强之间，我们观察到的增强相同，但排序略有不同。可以看出，锐度在 SVHN 中的总体重要性高于 CIFAR-10，



(a) SVHN 敏感性 (b) SVHN 重要性

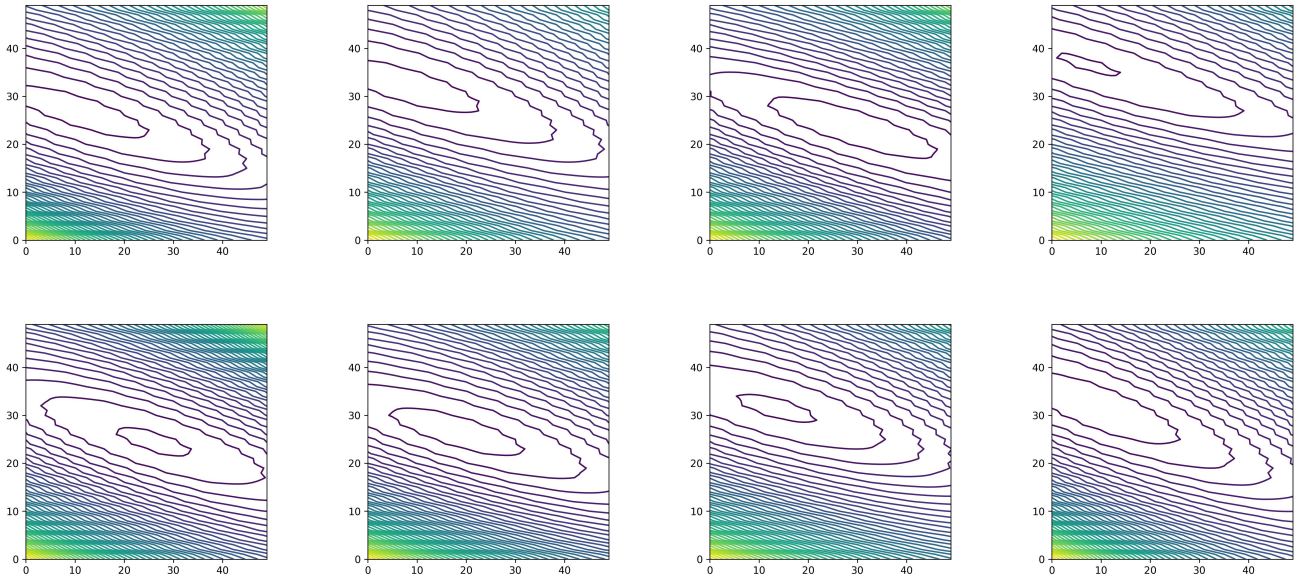
(c) CIFAR-10 敏感性 (d) CIFAR-10 重要性

图7：增强敏感度与重要性的全局视图

颜色也是如此。对比度、曝光反转、旋转、*Y*轴剪切和亮度对于两个数据集而言，其总体重要性都相对接近。考察图7中所有四种算法的重要性指标，可以清楚地看出，某些增强最为重要。对于 BYOL，SVHN 中最重要的是曝光反转，而 CIFAR-10 中最重要的是水平平移。此外，可以看到，对于 BYOL，两个剪切操作在两个数据集中始终具有相对较高的重要性。对于 NNCLR，SVHN 中最重要的是颜色，而 CIFAR-10 中最重要的是*X*轴剪切。对于 SimSiam，最重要的算子是 SVHN 中的水平平移，以及 CIFAR-10 中的*X*轴剪切和*Y*轴剪切。最后，对于 SwaV，对比度是两个数据集中最重要的增强。

6.4 景观分析

为了更好地比较自监督学习算法，并衡量批量大小在实验中的影响，我们采用一种分析最终下游模型损失的策略，称为损失景观分析。损失景观分析旨在通过对优化后的参数化附近的模型参数进行微小扰动，更深入地理解优化后的模型参数及其在相应参数空间中的位置。这会为给定的参数空间生成损失函数的景观，从而让我们了解解附近的损失空间是凸的还是混沌的。针对给定网络架构和问题生成损失景观并非易事，因此存在许多不同的方法。本研究采用 Li 等人 Li 等人（2018）提出的滤波器归一化方法来可视化损失景观。本研究选择滤波器归一化损失景观可视化方法，是因为该方法被发现能够准确捕捉极小化点的局部尖锐度和平坦度。理解



(a) BYOL (b) NNCLR (c) SimSiam (d) SwAV

图 8：比较使用批量大小为 32（上排）和批量大小为 256（下排）的自监督学习方法时，下游模型的损失景观。可以看到，自监督预训练中较小的批量大小会使下游任务中的最小化过程不那么尖锐。

损失景观的局部几何结构，是理解损失函数在当前问题中行为的重要组成部分。正如 Li 等人（2018）所述，逐滤波器归一化技术的一个关键组成部分，是 Goodfellow 等人（2014）和 Im 等人（2016）使用的随机向量技术。在该技术中，从随机高斯分布中采样两个随机向量 δ 和η，以及一个中心点 $\theta*$，其中 $\theta*$ 是优化后的参数配置。然后，通过绘制函数4生成一个二维等高线图:

$$f(\alpha, \beta) = L(\theta * + \alpha\delta + \beta\eta) \tag{4}$$

其中， α和 β 分别是随机向量 δ 和 η的长度。与早期随机向量方法不同，这种方法的关键在于对采样方向逐滤波器进行归一化。也就是说，将随机向量缩放到卷积神经网络中的滤波器尺度，以确保随机向量覆盖的区域对于给定参数而言不会过小或过大。我们主要发现，CIFAR-10 上的结果在批量大小的影响方面更有意思。因此，我们选择进一步探索在 CIFAR-10 数据集上进行的实验的损失景观。对于四种自监督学习算法 NNCLR、BYOL、SimSiam 和 SwaV，以及批量大小 32 和 256，我们采用滤波器归一化技术， α和 β 均设为 10，样本量设为 50，从而得到一个 $50x50$ 损失景观。Li 等人Li 等人（2018）发现，使用滤波器归一化方法可以直观地展示批量大小对极小值尖锐度的影响；较大的批量会产生视觉上更尖锐的极小值。我们将进化搜索机制找到的最优解的损失景观可视化。值得注意的是，对于全部四种自监督学习算法，在前置任务中使用 32 和 256 的批量大小进行训练、并在下游任务中保持批量大小为 32 时，我们都观察到了尖锐度差异。如图 8 所示，当在前置任务中使用 256 的批量大小时，BYOL、NNCLR 和 SimSiam 在下游任务中都收敛到了尖锐的极小值；而在前置任务中使用 32 的批量大小时，则收敛到了明显更平坦的极小值。SwaV 中观察到的结果恰好相反。

7 结论

本文的贡献有两个方面：一方面，我们致力于通过寻找最佳增强算子来优化自监督学习算法的性能。为此，我们提出了一种基于进化优化的方法，能够自动寻找最优增强算子及其强度，以最大限度地提高下游任务的准确率。为此，我们评估并比较了采用所提方法优化的四种最先进的自监督学习算法的性能。结果表明，所提方法提高了下游分类任务的准确率。另一方面，我们提出了用于解释优化后解的算法，以分析并确定应用于自监督学习算法前置任务的增强算子的影响。因此，我们在两个标准视觉数据集上设计了可解释性实验，以找出影响最大的增强算子。我们还进一步分析了批量大小和训练轮数等不同参数的影响，并对损失景观进行可视化，以便更好地理解最佳增强策略和性能最佳的模型。据我们所知，本研究首次尝试优化自监督学习算法前置任务中使用的超参数。我们的结果证实，自监督学习算法对增强策略的选择十分敏感。我们还证明，所提出的进化式增强优化方法能够通过高效搜索增强策略空间，找到可提升自监督学习算法性能的解。

利益冲突声明

作者声明，本研究是在不存在任何可能被视为潜在利益冲突的商业或财务关系的情况下开展的。

资金支持

本研究得到加拿大自然科学与工程研究委员会的资助。

数据可用性声明

本研究中分析的数据集可在 [https://www.cs.toronto.edu/ kriz/cifar.html] 和[http://ufldl.stanford.edu/housenumbers/] 找到。

参考文献

[数据集] （2020）。Bootstrap your own latent：一种新的自监督学习方法。 doi:10.48550/arXiv.2006.07733。ArXiv:2006.07733 [cs, stat]Bäck, T.、Eiben, A. E. 和 van der Vaart, N. A.（2000）。关于“无参数”气体的实证研究。载于自然启发式并行问题求解 *PPSN VI*：第六届国际会议，法国巴黎，*2000*年*9*月*18*日至*20*日，论文集 *6*（施普林格），315–324 Brazdil, P.、van Rijn, J. N.、Soares, C. 和 Vanschoren, J.（2022）。元学习：自动化机器学习与数据挖掘中的应用（施普林格·自然）Chen, X. 和 He, K.（2021）。探索简单的孪生表示学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。15750–15758 Chiong, R.、Weise, T. 和 Michalewicz, Z.（2012）。面向现实世界应用的进化算法变体，第2卷（施普林格）Cubuk, E. D.、Zoph, B.、Mane, D.、Vasudevan, V. 和 Le, Q. V.（2019）。AutoAugment：从数据中学习增强策略。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。113–123

Deng, L.（2012）。用于机器学习研究的手写数字图像MNIST数据库。*IEEE*信号处理杂志 29，141–142 Doersch, C.、Gupta, A. 和 Efros, A. A.（2015）。通过上下文预测进行无监督视觉表征学习。载于*IEEE*国际计算机视觉会议论文集。1422–1430 Donahue, J.、Krähenbühl, P. 和 Darrell, T.（2016）。对抗特征学习。*arXiv*预印本 *arXiv:1605.09782*Dosovitskiy, A.、Springenberg, J. T.、Riedmiller, M. 和 Brox, T.（2014）。利用卷积神经网络进行判别式无监督特征学习。神经信息处理系统进展27 Dwibedi, D.、Aytar, Y.、Tompson, J.、Sermanet, P. 和 Zisserman, A.（2021）。朋友的一点帮助：视觉表征的最近邻对比学习。载于*IEEE/CVF*国际计算机视觉会议论文集。9588–9597 Eiben, Á. E.、Hinterding, R. 和 Michalewicz, Z.（1999）。进化算法中的参数控制。*IEEE*进化计算汇刊3，124–141 Fortin, F.-A.、De Rainville, F.-M.、Gardner, M.-A.、Parizeau, M. 和 Gagné, C.（2012）。DEAP：让进化算法变得简单。机器学习研究期刊13，2171–2175 Gidaris, S.、Singh, P. 和 Komodakis, N.（2018）。通过预测图像旋转进行无监督表示学习。*arXiv*预印本 *arXiv:1803.07728*[数据集] Goodfellow, I. J.、Vinyals, O. 和 Saxe, A. M.（2014）。神经网络优化问题的定性刻画。doi:10.48550/ARXIV.1412.6544 Ho, D.、Liang, E.、Chen, X.、Stoica, I. 和 Abbeel, P.（2019）。基于种群的增强：高效学习增强策略调度。载于国际机器学习会议 （PMLR），2731–2741 Im, D. J.、Tao, M. 和 Branson, K.（2016）。深度网络损失景观的实证分析。*CoRR* abs/1612.04010 Isola, R. Z. P. 和 Efros, A. A.（2016）。彩色图像上色。*ECCV*。*Richard Zhang Phillip Isola* 和 *Alexei A Efros*Jenni, S.、Jin, H. 和 Favaro, P.（2020）。引导自监督特征学习超越局部像素统计。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。6408–6417 Katoch, S.、Chauhan, S. S. 和 Kumar, V.（2021）。遗传算法综述：过去、现在与未来。多媒体工具与应用80，8091–8126 Krizhevsky, A.（2009）。从小图像中学习多层特征。技术报告。Li, C.-L.、Sohn, K.、Yoon, J. 和 Pfister, T.（2021）。剪切粘贴：用于异常检测与定位的自监督学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。9664–9674 Li, H.、Xu, Z.、Taylor, G.、Studer, C. 和 Goldstein, T.（2018）。神经网络损失景观的可视化。技术报告 arXiv:1712.09913，arXiv。ArXiv:1712.09913 [cs, stat]类型：文章 Lim, S.、Kim, I.、Kim, T.、Kim, C. 和 Kim, S.（2019）。快速自动增强。神经信息处理系统进展 32 Misra, I.、Zitnick, C. L. 和 Hebert, M.（2016）。洗牌与学习：使用时间顺序验证的无监督学习。载于欧洲计算机视觉会议 （施普林格），527–544 Netzer, Y.、Wang, T.、Coates, A.、Bissacco, A.、Wu, B. 和 Ng, A. Y.（2011）。通过无监督特征学习读取自然图像中的数字。载于*NIPS*深度学习与无监督特征学习研讨会 *2011*Noroozi, M. 和 Favaro, P.（2016）。通过解决拼图进行视觉表征的无监督学习。载于欧洲计算机视觉会议（施普林格），69–84 Oord, A. v. d.、Li, Y. 和 Vinyals, O.（2018）。利用对比预测编码进行表示学习。*arXiv*预印本 *arXiv:1807.03748*

[数据集] O’Neill, M. (2009)。Riccardo poli、William B. Langdon、Nicholas F. McPhee：《遗传编程指南》：Lulu.com，2008年，250页，ISBN 978-1-4092-0073-4。Paszke, A.、Gross, S.、Massa, F.、Lerer, A.、Bradbury, J.、Chanan, G.等（2019）。PyTorch：一种命令式风格的高性能深度学习库。载于神经信息处理系统进展*32*（Curran Associates, Inc.），8024–8035。Pathak, D.、Krahenbuhl, P.、Donahue, J.、Darrell, T.和Efros, A. A.（2016）。上下文编码器：通过图像修复进行特征学习。载于*IEEE*计算机视觉与模式识别会议论文集，2536–2544。Reeves, C.（2003）。遗传算法。载于元启发式算法手册，F. Glover和G. A. Kochenberger编（波士顿，马萨诸塞州：Springer US），55–82。doi:10.1007/0-306-48056-5_3Shorten, C.和Khoshgoftaar, T. M.（2019）。深度学习图像数据增强综述。《大数据期刊》6，1–48。Srinivas, M.和Patnaik, L. M.（1994）。遗传算法中的交叉与变异自适应概率。*IEEE*《系统、人类与控制论汇刊》24，656–667。Such, F. P.、Madhavan, V.、Conti, E.、Lehman, J.、Stanley, K. O.和Clune, J.（2017）。深度神经进化：遗传算法是训练用于强化学习的深度神经网络的有竞争力替代方案。*arXiv*预印本 *arXiv:1712.06567*Susmelj, I.、Heller, M.、Wirth, P.、Prescott, J.和等，M. E.（2020）。Lightly。*GitHub*。注：*https://github.com/lightly- ai/lightly*Von Kügelgen, J.、Sharma, Y.、Gresele, L.、Brendel, W.、Schölkopf, B.、Besserve, M.等（2021）。利用数据增强从理论上将内容与风格分离的自监督学习。神经信息处理系统进展34，16451–16467。Yang, L.和Shami, A.（2020）。机器学习算法的超参数优化：理论与实践。《神经计算》415，295–316