Computer Science and Application 计算机科学与应用, 2026, 16(2), 405-413
Published Online February 2026 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2026.162069

Hans汉斯

LLM-NAS：一种基于大型语言模型的神经架构搜索进化框架

卢　烨，陈　磊*

广东工业大学数学与统计学院，广东　广州

收稿日期：2026年1月10日；录用日期：2026年2月7日；发布日期：2026年2月24日

摘　要

可微架构搜索(DARTS)已成为神经架构搜索的一种流行方法，然而，这种方法面临过早收敛到局部最优的情形。同时大语言模型(LLMs)已经成为能够完成广泛任务的强大工具。本文提出了一种基于大型语言模型的进化搜索框架LLM-NAS，用于神经架构搜索。它将LLM作为黑箱生成新的架构，允许在架构搜索过程中探索各种优化方向。在不同数据集和搜索空间上进行的大量实验表明，所提出的方法取得了与最先进的技术相当甚至更优的性能。

关键词

神经架构搜索，进化算法，大语言模型

LLM-NAS: An Evolutionary Framework for Neural Network Architecture Search Based on Large Language Model

Ye Lu, Lei Chen*

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: January 10, 2026; accepted: February 7, 2026; published: February 24, 2026

Abstract

Differentiable Architecture Search (DARTS) has become a popular method for Neural Architecture Search (NAS). However, this method faces the problem of premature convergence to local optima.

*通讯作者。

文章引用: 卢烨, 陈磊. LLM-NAS: 一种基于大型语言模型的神经架构搜索进化框架[J]. 计算机科学与应用, 2026, 16(2): 405-413. DOI: 10.12677/csa.2026.162069

Meanwhile, Large Language Models (LLMs) have evolved into powerful tools capable of accomplishing a wide range of tasks. This paper proposes an evolutionary search framework called LLM-NAS based on LLMs for Neural Architecture Search. It employs LLMs as a black box to generate new architectures, allowing the exploration of various optimization directions during the architecture search process. Extensive experiments conducted on different datasets and search spaces demonstrate that the proposed method achieves performance comparable to or even superior to state-of-the-art techniques.

Keywords

Neural Architecture Search (NAS), Evolutionary Algorithm (EA), Large Language Models (LLMs)

1. 引言

在过去十年中，深度神经网络(DNN)的迅速发展在计算机视觉领域取得了显著成果，例如图像分类、目标检测、目标跟踪、语义分割、深度估计等任务。其中，人工设计的网络架构在对应任务的特征提取方面发挥了关键作用。然而，随着各种任务的结果不断改善，人工设计网络架构的难度和成本也逐渐增加。为了减少为特定任务获取更优模型的需求并减少人为干预，神经架构搜索(NAS)作为一种自动搜索网络结构的方法引起广泛关注。

NAS 所采用的主要策略包括强化学习[1]、进化算法[2]和基于梯度[3]的方法。在早期的研究中，NAS 方法由于方法的计算成本过高而难以被广泛采用。基于梯度的架构搜索方法“可微架构搜索”(DARTS)作为一种替代方案，在 0.4 个 GPU 日内就能完成架构搜索任务，这极大地提高了搜索效率。然而这种方法容易收敛到局部最优解，从而严重影响了搜索阶段架构获取的性能。

本文提出了基于大型语言模型(LLMs)的神经架构搜索进化框架，称为 LLM-NAS。这种方法利用 LLMs 强大的生成能力提供创新方案，从而促进对搜索空间更有效的探索。具体而言，我们将进化策略与 LLMs 相结合，其中进化策略用于建立有效的搜索框架，而 LLMs 作为黑箱优化器生成具有多样性和先进性能的架构。我们从实验上展示了 LLM-NAS 在数据集 CIFAR-10、CIFAR-100，NAS-bench-201 基准测试上的性能，结果显示 LLM-NAS 搜索的结构性能与最先进结果相似。

2. 相关工作

2.1. 可微神经网络架构搜索

现有的可微网络搜索方法主要通过权重共享和连续松弛来实现梯度下降，以用于寻找离散架构。DARTS [3]是其中最具代表性的成果之一。尽管基于梯度的方法简单且计算效率高，但有研究表明，DARTS 经常会出现性能崩溃的问题[4]，从而导致性能下降。为应对这些挑战，已提出了多种解决方案。包括直接限制跳跃连接的数量[5]、利用架构参数的海森矩阵范数[6]，上述方法有效地提高了架构搜索的性能，但它们容易过早地收敛到局部最优解。

2.2. 进化神经网络架构搜索

进化神经架构搜索(ENAS)利用进化算法的全局搜索能力来避免过早收敛于局部最优解。然而，这种

方法所涉及的大量计算成本一直是显著的瓶颈[2]。Lu 等人[7]引入了 NSGA 算法以降低搜索成本。尽管上述工作有效地平衡了性能和搜索成本，但由于搜索空间巨大，其搜索效果仍面临挑战。

2.3. 大语言模型用于神经网络架构搜索

大型语言模型通过无监督学习对大量文本数据进行预训练，其表现远超传统语言模型，凭借强大的生成能力和上下文理解能力，大型语言模型在神经架构搜索任务中的应用得到了广泛研究。GENIUS [8]使用 GPT-4 [9]作为黑箱优化器，以便快速探索架构搜索空间并识别有潜力的候选方案。Yu 等人提出的 GPT-NAS [10]方法在神经网络架构数据集上对 GPT 模型进行微调，并将其与进化算法相结合，以优化个体并获得性能更优的后代与以往的方法不同，本文引入了利用大型语言模型与进化策略相结合的新方法，以在 DARTS 空间内生成架构个体，从而提高了搜索效率。

3. 方法论

在本节中，我们将详细介绍所提出的方法。首先，我们简要介绍 DARTS 以及协方差矩阵自适应进化策略(CMA-ES)在优化架构参数方面的应用，然后提出基于 CMA-ES 和大型语言模型的 NAS 框架。

3.1. 可微分架构搜索

DARTS 会探索最优单元架构，并通过重复堆叠普通单元(normal cells)和缩减单元(reduction cells)来构建超级网络(supernet)。需要注意的是，与普通单元相比，缩减单元位于网络总深度的 1/3 和 2/3 处，此处所有与输入节点步幅相邻的操作均设置为 2。在搜索过程中，每个单元被视为一个包含 N 个节点和 E 条边的有向无环图(DAG)，其中每个节点 $x^{(i)}$ 由特征图表示，每条边 (i,j) 代表在不同节点间信息流传递过程中的操作 $o^{(i,j)}$ 。在 DARTS 中，所有候选操作均采用连续松弛方法，以执行基于梯度的搜索。具体而言，中间节点通过候选操作的 softmax 混合计算得出：

$$\bar{o}^{(i,j)}\left(x^{(i)}\right)=\sum_{o\in\mathcal{O}}\frac{\exp\left(a_o^{(i,j)}\right)}{\sum_{o'\in\mathcal{O}}\exp\left(a_{o'}^{(i,j)}\right)}o\left(x^{(i)}\right) \tag{1}$$

其中 $i<j$，所有候选操作均存储于集合 $\mathcal{O}$ 中，$a_o^{(i,j)}$ 代表操作 $o^{(i,j)}$ 在超级网络构建过程中的混合权重。在松弛过程中，架构搜索以可微方式优化网络权重 ω 和架构参数 α ，为 NAS 建立了双层优化模型：

$$\min_{\alpha}\mathcal{L}_{val}\left(w^*,\alpha\right)\quad\text{s.t.}\quad w^*=\arg\min_w\mathcal{L}_{train}\left(w,\alpha\right). \tag{2}$$

其中，优化变量 α 和 ω 通过梯度下降法更新。搜索结束后，最终的体系结构由每条边上体系结构参数 α 最大的操作组成，$o^{(i,j)}=\arg\max_{o'\in\mathcal{O}}a_o^{(i,j)}$ 。

3.2. 协方差矩阵自适应进化策略

最近有研究提出采用协方差矩阵自适应进化策略对 DARTS 搜索空间中的架构参数进行优化[11]。首先从高斯分布中采样 N 个架构 $\boldsymbol{x}_n,n=1,2,\cdots,N$，该高斯分布以 α 为均值向量 $\boldsymbol{m}^0$、单位矩阵 $\boldsymbol{I}$ 为协方差矩阵 $\boldsymbol{C}^0$。也就是说，对于 $n=1,2,\cdots,N$ ，有 $\boldsymbol{x}_n=\alpha+\sigma\boldsymbol{y}$，其中 $\boldsymbol{y}\sim\mathcal{N}\left(0,\boldsymbol{I}\right)$。随后，利用 $\boldsymbol{x}_n$ 初始化 N 个基于 CMA-ES 的搜索过程，具体而言，第 n 个进化策略的初始搜索种群采样方式如下：

$$\begin{aligned}\boldsymbol{z}_i^t&=\boldsymbol{m}^t+\sigma_i\boldsymbol{y}_i,\\ \boldsymbol{y}_i&\sim\mathcal{N}\left(\boldsymbol{0},\boldsymbol{C}^t\right)\end{aligned} \tag{3}$$

其中 $t=0,\boldsymbol{C}^0=\boldsymbol{I},i=1,\cdots,\lambda$ ，σ 为步长，之后通过以下公式对 $\boldsymbol{m}$ 和 $\boldsymbol{C}^{t+1}$ 进行优化，从而生成新的个体 $\boldsymbol{z}_{i+1}^t$。

$$\boldsymbol{m}^{t+1}=\sum_{i=1}^{\lfloor \lambda/2 \rfloor}\beta_i \boldsymbol{z}_i^t \tag{4}$$

其中 β_i 表示分配给每个个体 x_i 的适应度权重。

$$\begin{aligned}\boldsymbol{C}^{t+1}=&\left(1-c_1-c_{\lfloor \lambda/2 \rfloor}\right)\cdot\boldsymbol{C}+c_1\cdot\left(p\cdot p^{\mathrm{T}}\right)\\&+c_{\lfloor \lambda/2 \rfloor}\cdot\sum_{i=1}^{\lfloor \lambda/2 \rfloor}\beta_i\cdot\left(\boldsymbol{z}_i^t-\boldsymbol{m}^{t+1}\right)\cdot\left(\boldsymbol{z}_i^t-\boldsymbol{m}^{t+1}\right)^{\mathrm{T}}\end{aligned} \tag{5}$$

p 是进化路径，在种群中，每个个体通过适应度函数进行评估并根据每个个体的适应度值进行排序。随后，采用最优的前$[\lambda/2]$个解来更新搜索参数。在多次迭代中，根据每个种群的适应度值将其存储在集合 P 中在最后的迭代阶段，我们从集合 P 中选择适应度值最高的个体 $\boldsymbol{z}_{best}^{*}$ 作为最佳输出个体代表第 n 次采样的最优搜索方向，对 DARTS 中的结构参数 α 进行更新：

$$\boldsymbol{a}_{t+1}=\boldsymbol{a}_t+\xi\boldsymbol{s}_{t+1} \tag{6}$$

其中 ξ 为超参数控制变化步长。

3.3. LLM-NAS

受到用于 NAS 应用的大语言模型研究的启发，我们提出了一种基于大型语言模型的神经架构搜索进化框架，称为 LLM-NAS。我们的方法旨在利用 LLM 强大的生成能力来解决 DARTS 空间中的搜索效率问题，框架图如图 1 所示。总的来说，可以分为以下几个部分：

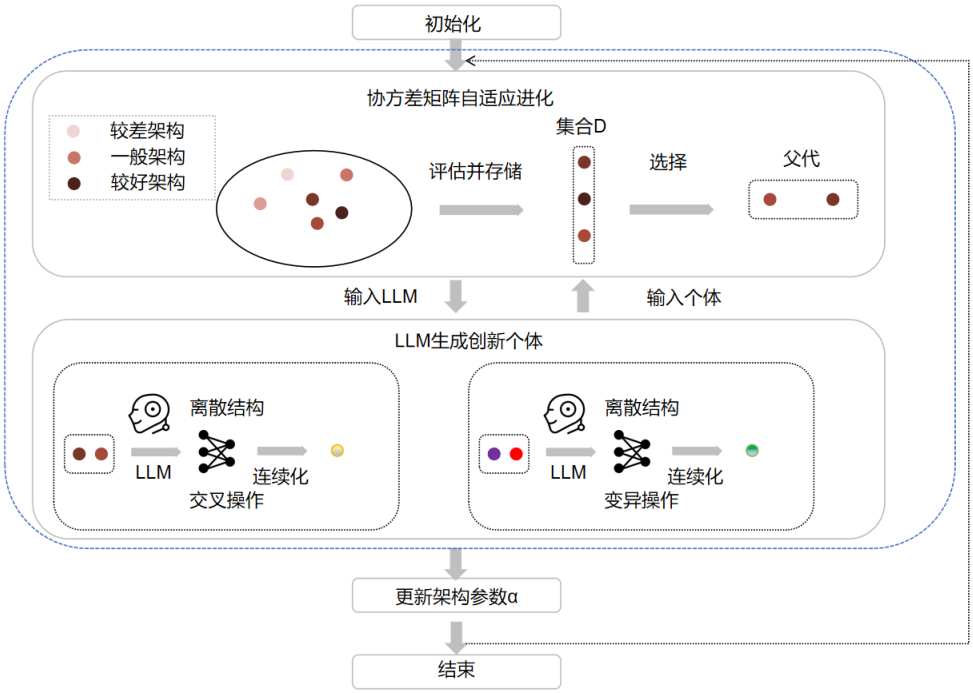

Figure 1. The framework of LLM-NAS model
图 1. LLM-NAS 模型框架图

3.3.1. 个体编码及种群初始化

LLM-NAS 首先对架构个体进行重新编码，以便大型语言模型能够识别神经架构。这是通过将个体的

架构参数的连续形式转换为对应的离散形式来完成的，然后再将该个体输入到 LLM 中。随后使用 CMA-ES 算法搜索结构并更新相关参数。随后，它会将每个样本最优架构存储进集合 D 作为输入 LLM 的初始种群。

3.3.2. 选择

从初始种群中选择两个个体作为父代，将连续的架构参数转换为对应的离散架构，同时避免两个离散架构相同的个体进行配对。

3.3.3. 交叉

给定任务规范、一对父代架构、它们的相对性能以及生成指令作为上下文，LLM-NAS 提示 LLM 生成离散形式的子代架构。

3.3.4. 变异

从精英架构中进行随机采样，并对其进行改进。变异线索包括任务规范、精英架构以及生成指令。

3.3.5. 基于 LLM 生成个体的 CMA-ES

将 LLM 生成的离散形式个体架构转化为连续形式的架构参数，操作为以现有模型架构参数 α 为基础，将 α 中将对应于离散架构的操作的坐标修改为最大值。具体而言，对每一架构：

$$architecture = \left\{ o^{(i,j)} \mid i \in \mathcal{N}, j \in \mathcal{N}', o \in \mathcal{O} \right\} \tag{7}$$

我们提取到每个节点 i 上的边 (i, j) 以及其对应的操作信息 $o^{(i,j)}$，信息可转换为对应参数矩阵上索引，此后我们将 α 中每个节点 i 对应边最大两个权重提取出来，记录如下：

$$w_i = top_2 \left\{ \alpha_o^{(i,j)} \mid i \in \mathcal{N}, j \in \mathcal{N}', o \in \mathcal{O} \right\} \tag{8}$$

我们将与该节点相关的不同边的最大权重值表示为 w_i^1，而这些边的第二大权重值则表示为 w_i^2。对应于这些权重值的索引分别表示为 $o_1^{(i,j)}$ 和 $o_2^{(i,j)}$。然后我们得到了所需要的连续架构参数 β：

$$\beta_o^{(i,j)} = \begin{cases} w_i^1 + \epsilon & \text{if } o^{(i,j)} \in o_1^{(i,j)} \\ w_i^2 + \epsilon & \text{if } o^{(i,j)} \in o_2^{(i,j)} \\ \alpha_o^{(i,j)} & \text{else?} \end{cases} \tag{9}$$

对于不同的架构 $architecture_i$ 有不同的架构参数 β_i，选择适应度值最高的个体作为均值向量以进行新一轮的协方差矩阵自适应进化以搜索架构。

4. 实验结果与分析

在这一部分，我们进行了大量的实验，以评估我们的方法在 DARTS 搜索空间上对 CIFAR-10、CIFAR-100 [12]和 ImageNet [13]图像分类任务的表现，同时也对 NAS-Bench-201 [14]搜索空间上的 CIFAR-10、CIFAR-100 和 ImageNet-16-120 进行了评估。丰富的对比实验和消融实验证明了我们方法的有效性。本文中提到的 ImageNet 数据集指的是 ImageNet 1K 或 ILSVRC2012 数据集，除了 NAS-Bench-201 中的 ImageNet-16-120 之外。

4.1. 数据集和实现细节

CIFAR10 与 CIFAR100：CIFAR-10 数据集包含 60,000 张不同类别的彩色图像，例如飞机和汽车。每个类别包含 6000 张图像，每张图像的尺寸为 32 × 32 像素。相比之下，CIFAR-100 数据集更为复杂，

包含 100 个不同的类别，其中包括多个细粒度的子类别。我们采用了与 DARTS 相同的操作空间，并在 PC-DARTS 中使用了部分连接策略以减少内存开销。我们使用批大小为 128 的方式进行超网的 50 个周期的训练(前 20 个周期用于预热)。随后，我们以 128 的批大小从头开始重新训练网络，总共进行 600 个周期的训练。在进化搜索过程中，群体规模设为 100，精英组中的个体数量设为 3，步长、样本数量 N 和初始通道数量分别设为 1、10 和 16。为了进行公平的比较，其余搜索和评估阶段的设置与 DARTS 相一致。

ImageNet：ImageNet 包含约 120 万张用于训练的图像和 5 万张用于验证的图像，涵盖了 1000 个类别，其难度远高于 CIFAR-10。在我们的研究中，通过在 CIFAR-10 和 CIFAR-100 数据集上进行搜索所发现的架构在 ImageNet 数据集上进行了评估。在评估阶段，我们从头开始训练网络，一共进行了 250 个周期。我们使用了一个带有线性衰减学习率(初始值为 0.5)、动量为 0.9 和权重衰减为 3×10^{-5} 的 SGD 优化器。

NAS-Bench-201：NAS-Bench-201 是一个用于评估 NAS 算法的常用基准测试工具，因为它提供了所有候选架构的性能数据，这些数据可通过查询直接获取。在 NAS-Bench-201 的搜索空间中，操作集包含 5 个元素，每个单元包含 4 个节点，从而形成了一个总共有 15,625 种架构的搜索空间。NAS-Bench-201 在 CIFAR-10、CIFAR-100 和 ImageNet16-120 数据集上提供了标准化的测试环境。在这个搜索空间中，我们可以通过直接评估数据集来获得特定任务的性能。因此，我们将特定任务的性能用作适应度指标，并通过使用四个不同的随机种子进行独立运行来得出最佳架构的平均值和标准差。

4.2. 实验结果

4.2.1. CIFAR10 与 CIFAR100

表 1 对 LLM-NAS 在 CIFAR-10 和 CIFAR-100 数据集上的表现与当前最先进的 NAS 方法进行了比较。可以看到 LLM-NAS 不仅在 CIFAR-10 数据上达到了最高表现，在 CIFAR-100 上达到了第二名的表现，搜索成本大幅超过 DARTS 的基准线，有较大竞争优势。

Table 1. Evaluation results on CIFAR10 and CIFAR100 dataset
表 1. CIFAR10 和 CIFAR100 数据集上的评估结果

架构	搜索成本 GPU-天数	CIFAR-10		CIFAR-100	
		参数量(M)	准确率(%)	参数量(M)	准确率(%)
ResNet [15]	-	**1.7**	95.39	1.7	77.9
ENAS + cutout [16]	0.5	4.6	97.11	4.6	-
AmoebaNet-A	3150	3.3	96.66	3.3	82.37
NSGA-Net	4.0	3.3	97.25	3.3	79.26
DARTS (1st)	0.4	3.4	97.00	3.4	82.46
DARTS (2nd)	1.0	3.3	97.24	3.3	-
GDAS [17]	0.2	3.4	97.07	3.4	81.62
P-DARTS + cutout	0.3	3.4	97.50	3.6	82.51
PC-DARTS + cutout	**0.1**	3.6	97.43	3.6	83.10
DARTS [18]	0.4	3.4	97.41	3.4	82.49
DrNAS [19]	0.4	4.0	97.46	4.0	-
EG-NAS [11]	**0.1**	3.2	97.47	3.2	**83.78**
LLM-NAS	**0.1**	4.0	**97.52**	3.6	83.38

4.2.2. ImageNet

表 2 给出了 LLM-NAS 与其它方法在 ImageNet 数据集上的比较，需要注意的是，我们使用的架构是在 CIFAR-10 上搜索得到的最佳架构。正如表 2 显示，LLM-NAS 在 0.1 GPU 天数的时间成本下测试误差表现与最先进的启发式算法相当。

Table 2. Evaluation results on ImageNet dataset
表 2. ImageNet 数据集上的评估结果

架构	测试误差 Top-1 (%)	搜索成本 GPU-天数	参数量(M)
DARTS (2nd)	26.7	1.0	4.7
SNAS [20]	27.3	1.5	2.8+
GDAS	26.0	0.3	3.4
PC-DARTS	25.1	0.1	4.7
DrNAS	24.2	4.6	5.7
NASNet-A	26.0	2000	3.3
NASNet-B	27.2	2000	3.3
NASNet-C	27.5	2000	3.3
AmoebaNet-A	25.5	3150	3.2
AmoebaNet-B	26.0	3150	3.2
AmoebaNet-C	24.3	3150	3.2
EAEPSO [21]	26.9	4.0	4.9
EG-NAS	24.9	0.1	5.3
LLM-NAS	25.1	0.1	5.1

4.2.3. NAS-Bench-201

表 3 给出了 LLM-NAS 在 NAS-Bench-201 基准集上的测试结果，值得注意的是，此时我们的适应度函数采用基准集上已有的测试数据，结果为使用随机种子 4 次后的平均值，实验结果显示优于当前的算法。

Table 3. Evaluation results on NAS-Bench-201 benchmark
表 3. NAS-Bench-201 基准集上的评估结果

架构	CIFAR-10		CIFAR-100		ImageNet16-120	
	Valid	Test	Valid	Test	Valid	Test
ResNet	90.83	93.97	70.42	70.86	44.53	43.63
Random (baseline)	90.93 ± 0.36	93.70 ± 0.36	70.60 ± 1.37	70.65 ± 1.38	42.92 ± 2.00	42.96 ± 2.15
ENAS	37.51 ± 3.19	53.89 ± 0.58	13.37 ± 2.35	13.96 ± 2.33	15.06 ± 1.95	14.57 ± 2.10
SETN [22]	84.04 ± 0.28	87.64 ± 0.00	58.86 ± 0.06	59.05 ± 0.24	33.06 ± 0.02	32.52 ± 0.21
DARTS (1st)	39.77 ± 0.00	54.30 ± 0.00	15.03 ± 0.00	15.61 ± 0.00	16.43 ± 0.00	16.32 ± 0.00
DARTS (1st)	39.77 ± 0.00	54.30 ± 0.00	15.03 ± 0.00	15.61 ± 0.00	16.43 ± 0.00	16.32 ± 0.00
GDAS [17]	90.01 ± 0.46	93.23 ± 0.23	24.05 ± 8.12	24.20 ± 8.08	40.66 ± 0.00	41.02 ± 0.00

续表

PC-DARTS	89.96 ± 0.15	93.41 ± 0.30	67.12 ± 0.39	67.48 ± 0.89	40.83 ± 0.08	41.31 ± 0.22
DSNAS [23]	89.66 ± 0.29	93.08 ± 0.13	30.87 ± 16.40	31.01 ± 16.38	40.61 ± 0.09	41.07 ± 0.09
DARTS−	91.03 ± 0.44	93.80 ± 0.40	71.36 ± 1.51	71.53 ± 1.51	44.87 ± 1.46	45.12 ± 0.82
iDARTS [24]	91.03 ± 0.44	93.80 ± 0.40	71.36 ± 1.51	71.53 ± 1.51	44.87 ± 1.46	45.12 ± 0.82
EG-NAS	90.12 ± 0.05	93.56 ± 0.02	70.78 ± 0.12	70.91 ± 0.07	44.89 ± 0.29	46.13 ± 0.46
LLM-NAS	**91.27 ± 0.11**	**94.02 ± 0.12**	**72.25 ± 0.48**	**72.25 ± 0.52**	**45.74 ± 0.18**	**46.31 ± 0.07**

5. 结论

本文提出了一种基于大语言模型(LLMs)的进化搜索框架 LLM-NAS，通过结合协方差矩阵自适应进化策略较快的迭代能力以及 LLMs 强大的创新生成能力，有效突破了传统 DARTS 方法的搜索局限，实现了架构搜索过程中的多样性探索。大量实验表明，相较于传统的神经网络架构搜索算法，LLM-NAS 在性能上得到显著提升，在各个数据集上都保持了竞争性的结果，同时还保持较低的搜索成本，为求解神经网络架构搜索问题提供了一种创新且高效的解决方案。

参考文献

[1] Zoph, B. and Le, Q.V. (2016) Neural Architecture Search with Reinforcement Learning.

[2] Real, E., Aggarwal, A., Huang, Y. and Le, Q.V. (2019) Regularized Evolution for Image Classifier Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 4780-4789. https://doi.org/10.1609/aaai.v33i01.33014780

[3] Liu, H., Simonyan, K. and Yang Y. (2018) DARTs: Differentiable Architecture Search. arXiv preprint arXiv:1806.09055. https://arxiv.org/abs/1806.09055

[4] Zela, A., Elsken, T., Saikia, T., *et al*. (2019) Understanding and Robustifying Differentiable Architecture Search.

[5] Chen, X., Xie, L., Wu, J. and Tian, Q. (2019) Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. 2019 *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 27 October-2 November 2019, 1294-1303. https://doi.org/10.1109/iccv.2019.00138

[6] Chen, X. and Hsieh, C.J. (2020) Stabilizing Differentiable Architecture Search via Perturbation-Based Regularization. 2020 *International Conference on Machine Learning*, Vienna, 18 July 2020, 1554-1565.

[7] Lu, Z., Whalen, I., Boddeti, V., *et al*. (2019) NSGA-Net: Neural Architecture Search Using Multi-Objective Genetic Algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference*, Prague, 13-17 July 2019, 419-427.

[8] Zheng, M., Su, X., You, S., *et al*. (2023) Can GPT-4 Perform Neural Architecture Search? arXiv preprint arXiv:2304.10970. http://arxiv.org/abs/2304.10970

[9] Achiam, J., Adler, S., Agarwal, S., *et al*. (2023) GPT-4 Technical Report. arXiv preprint arXiv:2303.08774. https://arxiv.org/abs/2303.08774

[10] Yu, C., Liu, X., Wang, Y., *et al*. (2023) GPT-NAS: Evolutionary Neural Architecture Search with the Generative Pre-trained Model.

[11] Cai, Z., Chen, L., Liu, P., Ling, T. and Lai, Y. (2024) EG-NAS: Neural Architecture Search with Fast Evolutionary Exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 11159-11167. https://doi.org/10.1609/aaai.v38i10.28993

[12] Krizhevsky, A. and Hinton, G. (2009) Learning Multiple Layers of Features from Tiny Images.

[13] Deng, J., Dong, W., Socher, R., Li, L., Li, K. and Li, F.F. (2009) ImageNet: A Large-Scale Hierarchical Image Database. 2009 *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 248-255. https://doi.org/10.1109/cvpr.2009.5206848

[14] Dong, X. and Yang, Y. (2020) Nas-Bench-201: Extending the Scope of Reproducible Neural Architecture Search.

[15] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778.

https://doi.org/10.1109/cvpr.2016.90

[16] Pham, H., Guan, M., Zoph, B., *et al*. (2018) Efficient Neural Architecture Search via Parameters Sharing. *International Conference on Machine Learning*, *Stockholm*, 10-15 July 2018, 4095-4104.

[17] Dong, X. and Yang, Y. (2019) Searching for a Robust Neural Architecture in Four GPU Hours. 2019 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 1761-1770. https://doi.org/10.1109/cvpr.2019.00186

[18] Chu, X., Wang, X., Zhang, B., *et al*. (2020) DARTS−: Robustly Stepping out of Performance Collapse without Indicators. arXiv preprint arXiv:2009.01027. https://arxiv.org/abs/2009.01027

[19] Chen, X., Wang, R., Cheng, M., *et al*. (2020) Drnas: Dirichlet Neural Architecture Search. arXiv preprint arXiv:2006.10355. https://arxiv.org/abs/2006.10355

[20] Xie, S., Zheng, H., Liu, C., *et al*. (2018) SNAS: Stochastic Neural Architecture Search. arXiv preprint arXiv:1812.09926. https://arxiv.org/abs/1812.09926

[21] Yuan, G., Wang, B., Xue, B. and Zhang, M. (2024) Particle Swarm Optimization for Efficiently Evolving Deep Convolutional Neural Networks Using an Autoencoder-Based Encoding Strategy. *IEEE Transactions on Evolutionary Computation*, **28**, 1190-1204. https://doi.org/10.1109/tevc.2023.3245322

[22] Dong, X. and Yang, Y. (2019) One-Shot Neural Architecture Search via Self-Evaluated Template Network. 2019 *IEEE*/*CVF International Conference on Computer Vision (ICCV)*, Seoul, 27 October-2 November 2019, 3681-3690. https://doi.org/10.1109/iccv.2019.00378

[23] Hu, S., Xie, S., Zheng, H., Liu, C., Shi, J., Liu, X., *et al*. (2020). DSNAS: Direct Neural Architecture Search without Parameter Retraining. 2020 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 12084-12092. https://doi.org/10.1109/cvpr42600.2020.01210

[24] Zhang, M., Su, S.W., Pan, S., *et al*. (2021) iDARTS: Differentiable Architecture Search with Stochastic Implicit Gradients. 2021 *International Conference on Machine Learning*, Online, 18-24 July 2021, 12557-12566.