用于黑箱优化的扩散大语言模型

Ye Yuan[1], [2*], **Can (Sam) Chen**[2*], **Zipeng Sun**[1, 2], **Dinghuai Zhang**[2, 3], **Christopher Pal**[2], [4], [5†], **Xue Liu**[1], [2†]

[1]麦吉尔大学, [2]MILA—魁北克人工智能研究所, [3] 微软研究院, [4]蒙特利尔理工学院, [5]加拿大 CIFAR 人工智能讲席

ye.yuan3@mail.mcgill.ca, can.chen@mila.quebec, zipeng.sun@mail.mcgill.ca,
dinghuai233@gmail.com, chris.j.pal@gmail.com, xueliu@cs.mcgill.ca

arXiv:2601.14446v1 [cs.CE] 20 Jan 2026

摘要

离线黑盒优化（BBO）旨在仅根据由设计及其标签构成的离线数据集寻找最优设计。这类场景常见于 DNA 序列设计和机器人学等领域，在这些领域中，可用的标注数据点寥寥无几。传统方法通常依赖特定任务的代理模型或生成模型，却忽视了预训练大语言模型（LLM）的上下文学习能力。近期研究将任务描述和离线数据集转化为自然语言提示，以此调整自回归大语言模型来执行 BBO，从而实现直接生成设计。然而，这些设计往往包含双向依赖关系，左至右的模型难以捕捉这些关系。本文探索将扩散大语言模型用于 BBO，利用其双向建模和迭代细化能力。基于此，我们提出上下文去噪模块：以任务描述和离线数据集为条件（两者均采用自然语言格式），并提示模型将掩码设计去噪为更优的候选项。为引导生成过程产出高性能设计，我们提出掩码扩散树搜索，将去噪过程建模为逐步进行的蒙特卡洛树搜索，动态平衡探索与利用。每个节点代表一个部分掩码的设计，每个去噪步骤对应一个动作，而候选项则通过在离线数据集上训练的高斯过程计算期望改进来评估。我们的方法*dLLM*在 design-bench 的少样本设置中取得了最先进的结果。

1 引言

设计新对象或实体以优化特定属性，是 DNA 序列设计和机器人学等领域面临的一项根本挑战 Trabucco 等（2022）。由于查询这些属性通常代价高昂 Hamidieh（2018）；Angerm¨uller 等（2020）；Barrera 等（2016）；Sample 等（2019），近期研究开始聚焦于避免在线评估的离线设置。由此产生了离线黑盒优化（黑盒优化）（Kim 等，2025），其目标是仅利用包含候选项及其相关标签的离线数据集，找出性能优异的设计。当可用的标注数据点很少时，这一挑战尤为突出。

传统方法通常采取以下两种做法之一：（1）训练代理模型（Trabucco 等人，2021；Yuan 等人，2023；Hoang 等人，2024；Yu 等人，2021；Fu 和 Levine，2021），为现有设计提供明确的梯度引导；或（2）训练条件生成模型（Kumar 和 Levine，2020；Krishnamoorthy 等，2023；Brookes 等人，2019），将目标值作为输入，直接生成高分设计。然而，这类针对特定任务且受数据限制的方法容易出现分布外泛化问题，也忽视了预训练大语言模型（LLM）（Brown 等人，2020）所具备的通用上下文学习能力，而后者提供了一种灵活的替代方案。

近期的工作将任务描述和离线数据集构造成自然语言提示，以此将自回归大语言模型应用于黑盒优化。这些模型通过上下文学习直接生成改进后的候选设计

*贡献相同（资深作者列于后面）。
†资深作者贡献相同。

（Yang 等人，2024年；Zhang 等人，2023年；Liu 等人，2024年；Nie 等人，2024年；Veli˘ckovi´c 等人，2024年；Novikov 等人，2025年）。然而，许多设计任务都涉及双向依赖关系——例如，DNA 单元不仅受其前缀影响，也受其后缀影响。由于自回归大语言模型以从左到右的方式生成序列，因此难以充分建模此类依赖关系，尤其是在高维且复杂的设计空间中。

本文探索了扩散大语言模型在黑盒优化中的应用。这些模型具有两项关键优势：双向建模和迭代细化，因此非常适合处理具有复杂依赖关系的设计任务。如图 1 所示，我们引入了**上下文去噪**模块：将任务描述和离线数据集整理为自然语言提示。随后，将这些提示与一条生成改进设计的指令一同提供给扩散大语言模型，由其通过迭代去噪，将掩码设计转化为改进后的候选项。

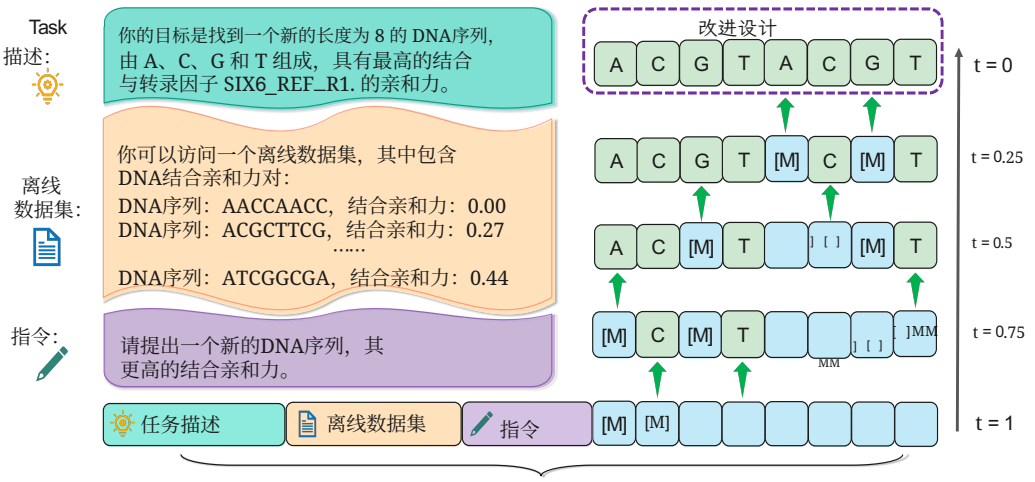

图 1：上下文去噪。

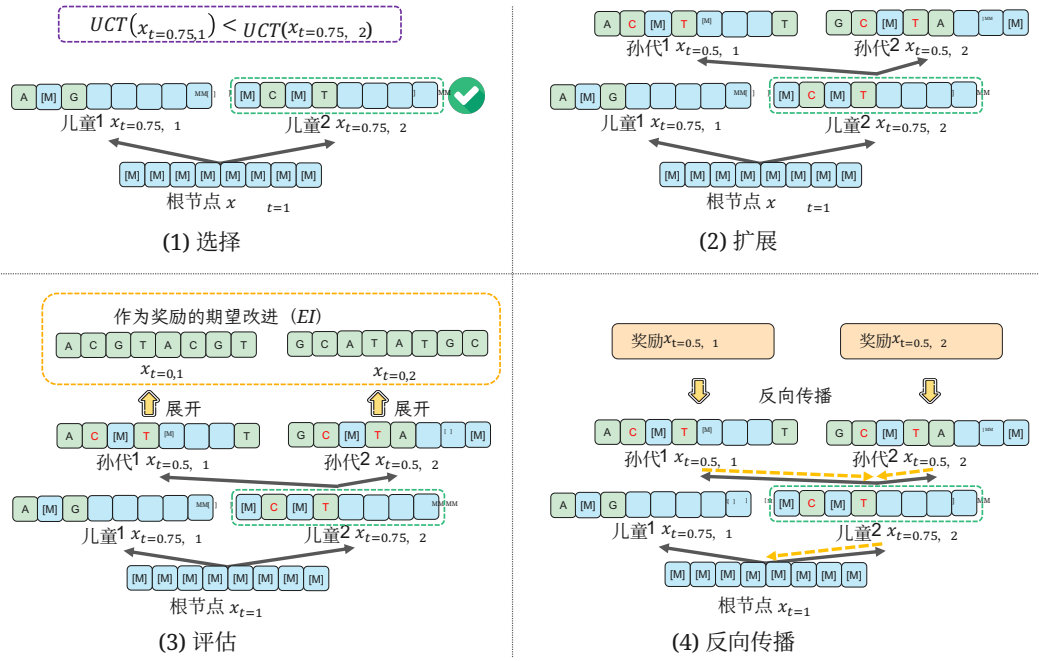

图 2：掩码扩散树搜索。

仅靠上下文去噪往往不足以引导模型生成高性能设计。为此，我们引入**掩码扩散树搜索**模块（图 2），将去噪过程建模为逐步进行的蒙特卡洛树搜索（MCTS）：每个掩码设计都是一个树节点，每个去噪步骤都是一个动作。搜索分为四个阶段，以动态平衡探索与利用：(1) 选择：从根节点 $\boldsymbol{x}_{t=1}$（完全掩码的设计）出发，使用 UCT 分数（Kocsis & Szepesv´ari, 2006）遍历树。该分数在利用高价值节点与探索访问次数较少的节点之间进行权衡，从而确定一个有前景的节点（例如，由于 $\boldsymbol{x}_{t=0.75,2}$ 的 UCT 分数更高，因此选择它而非 $\boldsymbol{x}_{t=0.75,1}$ ）；(2) 扩展：对选定节点（例如 $\boldsymbol{x}_{t=0.75,2}$）应用扩散大语言模型去噪，以生成新的子节点（例如 $\boldsymbol{x}_{t=0.5,1}$, $\boldsymbol{x}_{t=0.5,2}$）；(3) 评估：将每个子节点（例如 $\boldsymbol{x}_{t=0.5,1}$）展开为完整设计（例如 $\boldsymbol{x}_{t=0,1}$），并使用高斯过程下的期望改进来估算其奖励；(4) 反向传播：将奖励信号沿树向上传播（例如 $\boldsymbol{x}_{t=0.5,1} \rightarrow \boldsymbol{x}_{t=0.75,2} \rightarrow \boldsymbol{x}_{t=1}$），从而更新祖先节点并为后续决策提供信息。

总而言之，我们的贡献主要有三点：

- 据我们所知，我们是首个探索将扩散大语言模型用于黑盒优化的研究，并利用其双向建模和迭代细化能力。
- 我们提出了上下文去噪，通过自然语言任务描述和离线数据集对扩散大语言模型进行条件化，将掩码设计去噪为更优的候选项。
- 我们提出了掩码扩散树搜索，将扩散大语言模型去噪视为逐步执行的蒙特卡洛树搜索，以自适应地平衡探索与利用。

2 预备知识

2.1 离线黑盒优化

离线黑盒优化（BBO）旨在找到使未知目标函数 $f(\cdot)$ 最大化的最优设计 $\boldsymbol{x}^*$：

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} f(\boldsymbol{x}). \tag{1}$$

为解决这一问题，我们假设可以访问一个离线数据集 $\mathcal{D} = \{(\boldsymbol{x}_i,\ y_i)\}_{i=1}^N$，其中包含 N 个数据点。每个 $\boldsymbol{x}_i \in \mathbb{R}^d$ 表示一个 d 维设计，而 y_i 是其对应的得分。在许多实际场景中，可用的带标签数据点很少；因此，我们关注 N 较小的少样本设置（例如，$N = 10$）。

一种常见方法是以监督方式训练深度神经网络（DNN）模型来近似 $f(\cdot)$（Chen 等，2023），然后利用该模型指导设计优化。然而，在少样本设置中，由于数据有限，而且高斯过程（GP）模型能够提供有理论依据的不确定性估计，因此它们通常比训练得到的深度神经网络更有效。

具体而言，我们采用带有径向基函数（RBF）核的高斯过程模型（MacKay 等人，1998年）$K(\cdot,\ \cdot)$。给定离线数据集 $\boldsymbol{X} \in \mathbb{R}^{N \times d}$及其对应的标签 $\boldsymbol{y} \in \mathbb{R}^N$，我们通过计算预测后验分布来对新的设计 $\boldsymbol{x}$ 建模：

$$\hat{y}(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})). \tag{2}$$

2.2 扩散大语言模型

大语言模型（LLM）广泛应用于自然语言处理任务。与 GPT 等传统自回归模型（ARM）（Brown 等人，2020年）相比，扩散大语言模型（Nie 等人，2025b；Gong 等人，2025年；Arriola 等人，2025年）已成为一种颇具前景的替代方案，并以相当的模型规模取得了有竞争力的结果。与 ARM 相比，扩散大语言模型具有两大优势：（1）能够利用双向上下文；（2）采用迭代采样过程，可在测试时自然扩展。

扩散大语言模型通过两个步骤定义分布 $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$：前向破坏过程和反向去噪过程。在前向过程中，原始序列 $\boldsymbol{x}_0$ 中的词元会逐步被掩码，直到在 $t = 1$时，整个序列都被完全掩码。对于任意中间时间步 $t \in (0,1)$，序列 $\boldsymbol{x}_t$ 会被部分掩码，其中每个词元都独立地以概率 $p_{\boldsymbol{\theta}}$被掩码，

并以概率 $1-t$保持未掩码状态。随后，随着 t 从 1 逐渐减小到 0，反向过程通过迭代恢复被掩码的词元，重建原始数据分布。

扩散大语言模型的核心是一个掩码预测器，即一个参数化模型 $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_t)$，它以部分掩码序列 $\boldsymbol{x}_t$ 为输入，并同时预测所有掩码词元（记为 M）。模型通过最小化仅在掩码位置计算的交叉熵损失进行训练：

$$\mathcal{L}(\theta) = -\mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}[\boldsymbol{x}_t^i = \mathrm{M}]\log p_\theta(\boldsymbol{x}_0^i|\boldsymbol{x}_t)\right], \tag{3}$$

其中，$\boldsymbol{x}_0$ 从训练语料库中采样，t 从区间 $[0,1]$, 中均匀抽取，而 $\boldsymbol{x}_t$由正向掩码过程生成。指示函数 $\mathbf{1}[\cdot]$ 确保只有掩码词元会计入损失。训练完成后，掩码预测器便可模拟反向去噪过程，最终生成模型分布 $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$，作为最终的边缘分布。

3 方法

算法 1 概述了我们的方法，该方法结合了上下文去噪 （第 3.1 节）与掩码扩散树搜索 （第 3.2 节）。

3.1 上下文去噪

In-上下文提示我们通过串联以下内容来构造提供给扩散大语言模型的提示：

• **任务描述**：说明设计的含义和格式、对应的标签以及优化目标（最大化或最小化）。例如：

> 任务描述
>
> 你的目标是找到一个新的长度为 8 的 DNA序列，由 A、C、G 和 T 组成，并使其与转录因子SIX6 REF R1的结合亲和力最大。

• **离线数据集**：提供少样本集合，其中包含设计–标签对。例如：

> 离线数据集
>
> 你可以访问一个包含 DNA结合亲和力对的离线数据集：DNA序列：GGCCGGCC，结合亲和力：0.00 DNA序列：GCCCTTCG，结合亲和力：0.27 ..... DNA序列：GTGGGCGA，结合亲和力：0.44

• **指令**：提示模型生成更优的候选项。例如：

> 指令
>
> 请提出一个结合亲和力更高的新 DNA序列。

去噪 生成的结构化提示（任务描述、离线数据集、指令）被提供给扩散大语言模型，随后该模型将掩码候选迭代去噪为性能更高的设计。具体而言，设 $\boldsymbol{x}_t$ 表示步骤 t时的掩码设计，从完全掩码的设计 $t = 1$开始。给定采样间隔 Δt，下一步为 $s = t - \Delta t$。在每一步中，扩散大语言模型预测 $\boldsymbol{x}_t$ 中的所有掩码词元，以生成完整候选项 $\hat{}\boldsymbol{x}_0$。然后，我们使用 s/t的掩码比例，对 $\hat{}\boldsymbol{x}_0$中置信度最低的词元重新掩码，从而形成新的子节点$\boldsymbol{x}_s$ （Nie 等，2025b）。我们将采样限制为仅使用有效的 logits。例如，在 DNA 序列设计中，扩散大语言模型仅限于使用字母表 {A、C、G、T}；而在数值设计任务中，只允许使用数字和基本符号（“0–9、-、.”）。任何其他无效设计（例如无法构成有效数字的字符串）都会被排除。最终的 $\boldsymbol{x}_0$ 将作为所需设计返回。

算法 1 扩散大语言模型用于黑盒优化

输入： 离线数据集 $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$、预训练扩散大语言模型 $p_{\boldsymbol{\theta}}(\cdot)$、迭代次数 M 1：在 $\mathcal{D}$上拟合径向基函数核高斯过程 $f(\cdot)$。2：按照第（3.1）节构建上下文提示（任务描述、离线数据集、指令）。3：使用根节点 $\boldsymbol{x}_1$初始化搜索树。4：**对于**迭代 $= 1$至 M **执行**5：**选择：** 使用公式（4）中的 UCT 分数从 $\boldsymbol{x}_1$ 遍历树，并选择节点 $\boldsymbol{x}_t$。6：**扩展：** 扩展 $\boldsymbol{x}_t$，通过补全和重新掩码生成多样的子节点 $\boldsymbol{x}_{s,i}$。7：**评估：** 将每个子节点 $\boldsymbol{x}_{s,i}$ 展开为完整设计，并通过公式（5）计算奖励 $r_{s,i}$。8：**反向传播：** 使用公式（6）和（7）将$r_{s,i}$ 传播至所有祖先节点。9：**结束循环**10：返回已探索候选项中期望改进分数最高的完全去噪后的设计 $\boldsymbol{x}_0$。

3.2 掩码扩散树搜索

我们提出掩码扩散树搜索 以实现更有效的引导生成。该模块将去噪过程视为逐步决策：树中的每个节点都对应一个部分掩码设计 $\boldsymbol{x}_t$，可进一步去噪以生成新的子节点 $\boldsymbol{x}_{s,i}$，其中 $s < t$和 i索引第 i 个子节点。我们遵循标准的蒙特卡洛树搜索步骤：选择、扩展、评估和反向传播，以动态平衡探索与利用。

选择 每次迭代时，我们从父节点 $\boldsymbol{x}_u$ 开始遍历树，并使用 UCT 分数（Kocsis & Szepesv´ari, 2006年）选择一个子节点 $\boldsymbol{x}_{s,i}$：

$$\text{UCT}(\boldsymbol{x}_{s,i}) = V(\boldsymbol{x}_{s,i}) + \omega\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{s,i}|\boldsymbol{x}_u)\sqrt{\frac{\log N(\boldsymbol{x}_u)}{N(\boldsymbol{x}_{s,i})}}. \tag{4}$$

这里，$V(\boldsymbol{x}_{s,i})$ 是节点 $\boldsymbol{x}_{s,i}$ 当前的价值估计，有助于利用有前景的分支。模型似然 $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{s,i}|\boldsymbol{x}_u)$ 引导搜索转向更可能的解掩码步骤（Silver 等，2016）。 $N(\boldsymbol{x}_u)$ 表示父节点的访问次数，而 $N(\boldsymbol{x}_{s,i})$ 表示子节点的访问次数。因此，探索项 $\sqrt{\frac{\log N(\boldsymbol{x}_u)}{N(\boldsymbol{x}_{s,i})}}$ 鼓励选择访问次数较少的节点。系数 $\omega = 1.0$ 控制利用与探索之间的权衡。重复此过程，直到到达一个尚未扩展的叶节点。

扩展 选择节点 $\boldsymbol{x}_t$ 进行扩展后，使用扩散大语言模型生成设计补全。随后，通过按照比例 s/t对词元重新掩码来构建新的子节点 $\boldsymbol{x}_{s,i}$。根据分支因子对不同补全进行采样，可以从同一父节点生成多样的子节点 $\boldsymbol{x}_{s,i}$。

评估 为了评估新的子节点 $\boldsymbol{x}_{s,i}$，我们执行多次去噪运行，以获得 J完全未掩码的设计 $\boldsymbol{x}_{0,i}^{(j)}$。然后，使用高斯过程中的期望改进（EI）准则计算其奖励：

$$\text{EI}(\boldsymbol{x}_{0,i}^{(j)}) = \left(\mu(\boldsymbol{x}_{0,i}^{(j)}) - f_{\text{best}}\right)\Phi(z_j) + \sigma(\boldsymbol{x}_{0,i}^{(j)})\phi(z_j), \quad \text{where} \quad z_j = \frac{\mu(\boldsymbol{x}_{0,i}^{(j)}) - f_{\text{best}}}{\sigma(\boldsymbol{x}_{0,i}^{(j)})}. \tag{5}$$

其中，$\mu(\cdot)$ 和 $\sigma(\cdot)$ 分别为高斯过程预测均值和标准差，Φ 和 ϕ 分别为标准正态累积分布函数和概率密度函数，f_{best} 为离线数据集中的最佳得分。最后，我们将节点的奖励设为平均 EI：$r_{s,i} = \frac{1}{J}\sum_{j=1}^{J}\text{EI}(\boldsymbol{x}_{0,i}^{(j)})$。

反向传播 获得奖励 $r_{s,i}$ 对于$\boldsymbol{x}_{s,i}$后，我们将其沿整个轨迹进行反向传播。对于每个具有 $\tau > s$ 的祖先节点 $\boldsymbol{x}_\tau$，我们更新其访问次数和价值估计：

$$N_{\text{new}}(\boldsymbol{x}_\tau) = N_{\text{old}}(\boldsymbol{x}_\tau) + 1, \tag{6}$$

$$V_{\text{new}}(\boldsymbol{x}_\tau) = \frac{V_{\text{old}}(\boldsymbol{x}_\tau)\cdot N_{\text{old}}(\boldsymbol{x}_\tau) + r_{s,i}}{N_{\text{new}}(\boldsymbol{x}_\tau)}. \tag{7}$$

最后，我们返回从已探索的候选项中获得高期望改进分数的完全去噪设计 $\boldsymbol{x}_0$。

我们提出的掩码扩散树搜索模块是一种实用的、基于 UCT 的启发式方法，用于探索高维设计空间。经典 UCT 在理想化条件下具有理论保证，但要将此类分析扩展到我们的设置——其中扩展来自扩散大语言模型，奖励则依赖于基于高斯过程的期望改进——需要额外作出严格假设，这超出了本文的讨论范围。我们转而关注其经验表现；第 (4.5) 节的大量消融实验和第 (4.6) 节的敏感性研究表明，该模块在各项任务中都有效且稳健。

4 实验

我们通过在 design-bench 的少样本设置下进行大量实验，评估所提出的*dLLM* 的有效性。具体而言，我们在第 (4.4) 节与成熟的基线方法进行比较，在第 (4.5) 节评估各组件的贡献，并在第 (4.6) 节分析超参数敏感性。

4.1 基准测试

数据集按照 Nguyen 等人，2023年的做法，我们在两个连续任务和两个离散任务上进行评估。连续任务包括：**(1) 蚂蚁形态（Ant）**Brockman 等人（2016），该任务需要优化一个 60 维的蚂蚁形态，以实现快速爬行；以及**(2) D'Kitty 形态（D'Kitty）**Ahn 等人（2020年），该任务需要优化一个 56 维的 D'Kitty 形态，以实现相同目标。离散任务包括：**(1) TF Bind 8（TF8）**Barrera 等（2016），该任务需要设计一个长度为 8 的 DNA序列，以最大化其与 SIX6 REF R1 转录因子的结合活性；以及**(2) TF Bind 10（TF10）**Barrera 等（2016），该任务将序列长度扩展至 10。我们通过从离线数据集中均匀抽取**10** 个样本来构建少样本设置。

评估 对于每个生成的设计，我们采用 *Design-Bench* 基准任务 中描述的预言机（Trabucco 等（2022））进行评估。与以往工作（Trabucco 等人，2021）一致，我们对每种方法评估 128 个候选项，并报告归一化真实分数的最大值（即 100th 百分位数）。归一化分数 y_n 的计算方式为 $y_n = \frac{y - y_{\min}}{y_{\max} - y_{\min}}$，其中 y 表示设计的原始得分，$y_{\min}$ 和 $y_{\max}$ 分别表示整个未观测数据集中的最低分和最高分。为进行更全面的比较，我们还报告了各竞争方法的平均排名和中位排名，以及归一化分数的中位数。

4.2 基线方法

我们将所提出的方法与一系列成熟的基线方法进行比较。

基于代理模型的方法 这些方法着重于学习可靠的代理模型，以指导设计更新。**(1) Grad (mean)**：使用高斯过程预测均值 $\mu(\boldsymbol{x})$ 作为代理预测器，并利用其梯度更新离线设计。**(2) Grad (EI)**：使用公式（5）中的期望改进 EI($\boldsymbol{x}$) 作为代理预测器，并利用其梯度更新离线设计。**(3)COMs**（Trabucco 等人，2021）：通过对分布外设计上的神经代理模型预测值设置下界，采用保守目标，随后进行代理模型梯度上升。**(4) ICT**（Yuan 等人，2023）：采用轮换式伪标签器和三个代理模型之间的协同教学策略，随后进行元学习样本重加权。**(5) MATCH-OPT**（Hoang 等人，2024）：通过代理模型与梯度不匹配来约束性能差距，从而提升代理模型质量和优化性能。**(6) UniSO-T**（Tan 等人，2025）：训练序列到序列自回归代理模型来预测编码为词元的标签，并使用训练好的代理模型指导设计生成。

生成模型方法 这些方法使用变分自编码器、生成对抗网络、自回归模型或扩散模型，直接对有前景的设计的分布进行建模。**(1) CbAS**（Brookes 等人，2019年）：训练一个变分自编码器，并逐步使其适应高得分设计。**(2) ExPT**（Nguyen 等人，2023年）：在多样化的合成函数上预训练基于变换器的变分自编码器，并通过上下文学习生成来采样改进的设计。**(3) MIN**（Kumar 和 Levine，2020年）：使用生成对抗网络对从得分到设计的逆映射进行建模，然后通过以高得分为条件来查询改进的设计。**(4) BONET**（Mashkaria 等人，2023年）：在从低得分到

表1：四项任务上第100百分位归一化分数的对比实验结果。

方法	蚂蚁形态	D'Kitty 形态	TF 绑定8	TF 结合 10	平均排名	排名中位数
$\mathcal{D}$（最佳）	0.565	0.884	0.439	0.467	–	–
梯度均值	0.628 ± 0.019	0.922 ± 0.006	0.700 ± 0.103	0.603 ± 0.033	10.3/16	9.5/16
Grad-EI	0.620 ± 0.022	$\underline{0.939} \pm \underline{0.001}$	0.701 ± 0.109	0.570 ± 0.036	9.5/16	11.0/16
COMs	0.629 ± 0.022	0.934 ± 0.002	0.681 ± 0.086	0.596 ± 0.038	10.0/16	9.5/16
ICT	0.639 ± 0.015	0.925 ± 0.009	0.752 ± 0.050	0.605 ± 0.024	8.0/16	8.5/16
MATCH-OPT	0.619 ± 0.018	0.920 ± 0.004	0.696 ± 0.050	0.598 ± 0.065	11.5/16	10.5/16
UniSO-T	$\underline{0.642} \pm \underline{0.010}$	$\underline{0.939} \pm \underline{0.022}$	0.844 ± 0.033	0.631 ± 0.018	$\underline{2.8}/16$	$\underline{2.5}/16$
CbAS	0.594 ± 0.036	0.910 ± 0.007	0.738 ± 0.026	0.614 ± 0.008	11.3/16	11.5/16
ExPT	0.641 ± 0.021	0.935 ± 0.004	0.826 ± 0.036	0.640 ± 0.014	3.3/16	3.5/16
MIN	0.588 ± 0.065	0.892 ± 0.009	0.776 ± 0.011	0.567 ± 0.010	13.5/16	14.5/16
BONET	0.638 ± 0.031	0.927 ± 0.003	0.782 ± 0.115	0.595 ± 0.035	8.0/16	7.0/16
ORPO	0.596 ± 0.014	0.878 ± 0.009	0.778 ± 0.033	0.499 ± 0.033	13.0/16	14.0/16
GTG	0.597 ± 0.021	0.904 ± 0.002	0.795 ± 0.079	0.630 ± 0.035	8.8/16	9.0/16
DDOM	0.589 ± 0.016	0.902 ± 0.002	0.760 ± 0.064	0.615 ± 0.033	11.3/16	11.5/16
CMA-ES	0.603 ± 0.142	0.725 ± 0.002	0.811 ± 0.057	0.628 ± 0.044	9.5/16	8.5/16
MCTS-transfer	0.633 ± 0.017	0.933 ± 0.020	$\underline{0.848} \pm \underline{0.023}$	$\underline{0.638} \pm \underline{0.008}$	4.3/16	4.5/16
dLLM(ours)	$\mathbf{0.652 \pm 0.030}$	$\mathbf{0.942 \pm 0.012}$	$\mathbf{0.876 \pm 0.038}$	$\mathbf{0.642 \pm 0.013}$	**1.0/16**	**1.0/16**

高得分样本的轨迹上训练自回归模型，并将其展开以生成候选项。**(5) OPRO**（Yang 等人，2024年）：将设计–标签历史输入自回归大语言模型，直接采样新的设计。由于 LLaMA3-8B-Instruct 的规模与 LLaDA-8B-Instruct 相当，我们采用前者（Dubey 等人，2024年）。**(6) GTG**（Yun 等人，2024年）：在来自离线数据的合成轨迹上学习条件扩散模型，引导设计趋向高得分区域。**(7)DDOM**（Krishnamoorthy 等，2023）：在离线数据集上训练条件扩散模型，并应用无分类器引导来获取候选项。

我们还与其他方法进行了比较：**(1) CMA-ES** (Hansen，2006年)：通过调整协方差矩阵，使其朝向得分较高的区域。**(2) MCTS-transfer** (Wang 等人，2024年)：通过蒙特卡洛树搜索迭代划分并探索子空间，自适应地生成改进设计。

4.3 实现细节

我们采用预训练的 dLLM LLaDA-8B-Instruct 作为基础模型（Nie 等，2025b），并在离线数据集上拟合一个使用 RBF核的高斯过程（GP）模型。搜索树的深度限制为 4，分支因子为 5。树搜索的总步数记为 M，见算法 1，设为 16；完全解除掩码的设计数量 J 设为 5。dLLM 的采样温度设为 1。0。完整的上下文提示见附录 A.2。所有实验均在一块配备 80GB 内存的 NVIDIA A100 GPU 上进行。运行时间方面，在 Ant 上完成一次完整的搜索过程约需 30 分钟，在 TF8 上约需 40 分钟。

4.4 结果与分析

我们在表1中报告了第100百分位数归一化分数，并分别以**粗体**和下划线标出最佳和次佳结果，同时汇总了各任务的均值/中位排名。第50百分位数的结果见附录A.3。

我们的主要观察结果如下：**(1)** *dLLM* 在15种基线方法中的平均排名和中位排名均为最高。在全部四项任务中，它都排名第一，在连续和离散设计空间中均展现出稳定的性能提升。**(2)**纯基于梯度的方法（Grad-mean、Grad-EI）表现明显较差，这表明仅凭代理模型无法可靠地识别优质设计。将强大的生成模型（如 dLLM）与树搜索相结合，对于有效探索设计空间至关重要。**(3)** 在基于代理模型的方法中，UniSO-T表现最佳，这可能归功于它利用了具有强大无监督知识的预训练T5模型。不过，它仅适用于回归任务。**(4)** ExPT是生成式基线方法中表现最强的，这得益于其在合成函数上的预训练。然而，它的表现仍不及*dLLM*，后者利用了更广泛的预训练知识。DDOM（不使用外部预训练的扩散模型）的表现更差，也进一步印证了这一点。**(5)**ORPO将LLaMA3-8B-Instruct与高斯过程模型的反馈相结合，但其表现不如*dLLM*。其从左到右的自回归建模方式本身限制了它全面捕捉设计中双向依赖关系的能力，在TF8和TF10中尤为明显。

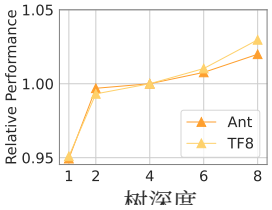

图3：对不同树深度的敏感性。

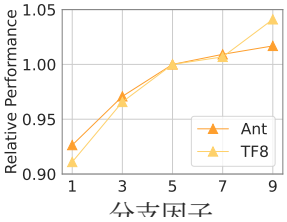

图4：对不同分支因子的敏感性。

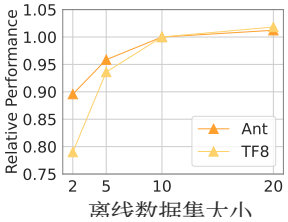

图5：对不同离线数据集规模的敏感性。

4.5 消融实验

我们通过替换或移除*dLLM* 的关键模块进行消融实验，详情见表2。

原始扩散 我们将扩散大语言模型替换为直接在离线数据集上训练的连续扩散模型（Krishnamoorthy 等，2023）。这会导致性能明显下降，尤其是在离散序列任务上，因为该模型仅使用少样本进行训练，缺乏 dLLM 丰富的预训练知识，因此难以泛化到未见设计。

无 MDTS 我们移除掩码扩散树搜索 （MDTS）模块，让扩散大语言模型直接根据上下文提示生成候选项，然后使用 GP 预测器选出最佳候选项。这导致所有任务的性能均大幅下降，凸显了基于树的探索对于在设计空间中导航的重要性。

无模板我们通过移除任务描述来对提示构造进行消融实验，仅保留离线数据集和指令作为输入。缺少这种明确的任务框架后，模型在所有任务上的性能均有所下降，这表明，提示中的清晰语义引导有助于扩散大语言模型更好地理解目标并生成更契合目标的候选项。

表2：消融实验。

方法	Ant	D'Kitty	TF8	TF10
原始扩散	0.590 ± 0.018	0.897 ± 0.008	0.764 ± 0.068	0.603 ± 0.014
无 MDTS	0.604 ± 0.003	0.892 ± 0.001	0.798 ± 0.012	0.503 ± 0.013
无模板	0.630 ± 0.005	0.934 ± 0.015	0.846 ± 0.027	0.633 ± 0.017
dLLM(ours)	**0.652** ± **0.030**	**0.942** ± **0.012**	**0.876** ± **0.038**	**0.642** ± **0.013**

我们的消融实验表明，每个组成部分都不可或缺。我们还测试了无分类器引导（Nie 等人，2025a），结果相近，但推理成本翻倍，因此未采用。

4.6 超参数敏感性

我们在两个有代表性的任务 Ant 和 TF8 上进行了敏感性分析。所有结果均以相对性能报告，即将性能相对于默认配置进行归一化。

树深度在掩码扩散树搜索中，树深度对应于达到完全解除掩码的设计所需的去噪步骤数。默认值为 4，我们将其取值范围设为 $\{1, 2, 4, 6, 8\}$。如图3所示，随着深度增加，性能稳步提升，体现了 dLLM 去噪过程中迭代细化的益处。

分支因子 接下来，我们考察分支因子的影响，即每个节点扩展的候选子节点数量。默认值为 5，我们在 $\{1, 3, 5, 7, 9\}$ 范围内对其进行调整。如图4所示，较大的分支因子通常能带来更好的性能，凸显了更广泛探索的优势。值得一提的是，dLLM 天然支持通过树深度实现顺序扩展，以及通过分支因子实现并行扩展，不过其收益最终受限于高斯过程模型的认知不确定性。

离线数据集大小 接下来，我们调整离线数据集的大小，这既会影响高斯过程预测器的训练，也会影响上下文提示中的离线数据集。默认大小为 10，我们测试$\{2, 5, 10, 20\}$。如图5所示，当可用样本仅有 2 个时，性能会急剧下降，

因为高斯过程和提示都缺乏足够的信息。除这一极端情况外，随着数据增多，性能稳步提升并趋于稳定。在附录 A.4 中，我们进一步研究了使用不同上下文提示的影响，发现我们的方法对提示变化具有鲁棒性。

dLLM 在主要实验中，我们采用 LLaDA-8B-Instruct（Nie 等，2025b）作为扩散大语言模型骨干。我们还测试了 MMaDA-8B-MixCoT（Yang 等，2025），这是一个规模相当、采用统一多模态预训练的模型。在 Ant（0.649 ± 0.023 使用 MMaDA，0.652 ± 0.030 使用 LLaDA）和 TF8（0.862 ± 0.027 使用 MMaDA，0.876 ± 0.038使用 LLaDA）上的结果非常接近，表明我们的方法对扩散大语言模型骨干的选择具有鲁棒性。

5 相关工作

用于黑盒优化的大语言模型 大语言模型凭借强大的模式识别和上下文学习能力，近期在黑盒优化领域受到关注（Song 等（2024））。目前已形成两条主要研究路线：（1）使用大语言模型预测给定黑盒设计的属性 y $\boldsymbol{x}$，通常采用微调；（2）直接提示大语言模型——不更新参数——以生成有潜力的设计 $\boldsymbol{x}$。由于能够捕捉双向依赖关系，T5 编码器（Raffel 等，2020年）等双向模型已广泛应用于第一条路线：Nguyen 等（2024）；Tang 等（2024）研究了将大语言模型嵌入用于回归，并证明其在高维设计空间中的有效性。Tan 等（2025）进一步提出了一种统一的基于字符串的表示，用于编码元数据和设计值。自回归大语言模型也得到探索：Zhao 等（2024）分析了它们在二元分类任务中的决策边界，而 Nguyen 和 Grover（2024）则通过加入分子嵌入和属性预测器，使其能够自回归地预测分子属性。

在第二条研究路线中，自回归大语言模型已被直接用作黑盒优化中的优化器。Zhang 等（2023）将任务描述和带标签的设计作为提示，以引导设计生成；Liu 等（2024）则以进化方式利用大语言模型选择用于交叉和变异的父代。除了通用设计任务之外，程序合成也已成为一个具体且前景广阔的领域，因为代码可以用语言自然地描述，并通过编译器进行评估。FunSearch（Veli$\check{}$ckovi$\acute{}$c 等，2024）和 AlphaEvolve（Novikov 等人，2025）将代码视为设计空间，并利用大语言模型引导的进化策略来优化目标函数，取得了出色的实证结果。我们的工作遵循第二条研究路线：我们利用大语言模型的上下文学习能力来生成改进设计。不过，与主要使用自回归模型的既有工作（Zhang 等，2023）不同，我们研究扩散大语言模型；这类模型具备双向建模和迭代采样能力，因此尤其适用于黑盒设计问题。

蒙特卡洛树搜索 蒙特卡洛树搜索（MCTS）已被应用于黑盒优化（BBO），以提升样本效率和搜索质量。LA-MCTS（Wang 等，2020）通过学习空间划分，引导黑箱优化器将采样集中于有潜力的局部区域。Wang 等（2024）利用蒙特卡洛树搜索，通过从相似的源任务中迁移知识，自适应地构建和优化新任务的搜索空间。Liu 等（2025）在 ESM3 中应用奖励引导的蒙特卡洛树搜索框架，以发现能够折叠成给定骨架结构的蛋白质序列。
蒙特卡洛树搜索也已应用于扩散模型领域，其中去噪过程被重新诠释为一种基于树的序贯搜索——每个节点代表一个部分去噪的样本，每个动作对应一个去噪步骤。Yoon 等（2025a）提出了蒙特卡洛树扩散，Yoon 等（2025b）则通过并行展开和轨迹粗化进一步加速该方法。Zhang 等（2025）将混合蒙特卡洛树搜索策略融入扩散去噪流程，实现了可扩展的测试时推理。这些研究主要聚焦于迷宫导航等规划任务。Jain 等（2025）提出一种基于树的方法，利用预训练扩散模型从与奖励对齐的目标分布中采样，并在文生图和语言建模任务中展现出优异效果。与我们的工作最为接近的是 Tang 等（2025），他们提出了一种基于扩散语言模型的分子优化蒙特卡洛树搜索框架。然而，他们的模型处理的是 SMILES 表示，因此适用范围局限于小分子，且不具备大语言模型的上下文学习能力。

6 结论

自回归大语言模型近期已应用于黑箱设计生成，但难以捕捉双向依赖关系。在本研究中，我们探索了扩散大语言模型在离线黑箱优化中的应用，利用其双向建模和迭代细化能力。我们引入了两个关键模块：上下文去噪，该模块以任务描述和离线数据集为条件，引导扩散大语言模型细化掩码设计；以及掩码扩散树搜索，该模块将去噪过程建模为蒙特卡洛树搜索，以平衡探索与利用。这两个模块相结合，使我们的方法在少样本设置下于 design-bench 上取得了最先进的结果。我们的结果展现了扩散大语言模型作为通用优化器的潜力，并为科学与工程领域的数据高效设计生成开辟了新方向。

可复现性声明

为确保结果可复现，我们在第4.3节中提供了全面的实现细节。

伦理声明

我们提出的方法 *dLLM* 有望通过实现对复杂设计空间的更有效优化，加速生物医学和机器人学等领域的科学发现。这些优势有望推动多个领域取得实质性进展。与此同时，与许多强大的优化工具一样，它也存在被滥用的风险。尤其是，生成和改进高性能设计的能力可能会被用于有害目的，例如加速生物武器或其他恶意技术的研发。为降低此类风险，必须建立适当的防护措施、监督机制和监管框架，确保该方法得到负责任的使用，且严格用于有益目的。

参考文献

Michael Ahn、Henry Zhu、Kristian Hartikainen、Hugo Ponte、Abhishek Gupta、Sergey Levine 和 Vikash Kumar。《Robel：用于低成本机器人学习的机器人学基准》。载于机器人学习会议（*CoRL*），2020年。

Christof Angerm¨uller、David Dohan、David Belanger、Ramya Deshpande、Kevin Murphy 和 Lucy Colwell。基于模型的生物序列设计强化学习。载于 第八届国际学习表征会议，*ICLR 2020*，埃塞俄比亚亚的斯亚贝巴，*2020*年*4*月*26*日至*30*日。OpenReview.net，2020年。

Marianne Arriola、Aaron Gokaslan、Justin T Chiu、Zhihan Yang、Zhixuan Qi、Jiaqi Han、Subham Sekhar Sahoo 和 Volodymyr Kuleshov。块扩散：在自回归语言模型与扩散语言模型之间进行插值。*ArXiv*预印本，abs/2503.09573，2025年。

Luis A Barrera 等。人类转录因子变异调查揭示了普遍存在的 DNA结合变化。《科学》，2016年。

Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 Wojciech Zaremba。OpenAI Gym。*ArXiv*预印本，abs/1606.01540，2016年。

David H. Brookes、Hahnbeom Park 和 Jennifer Listgarten。通过自适应采样进行稳健设计。在 Kamalika Chaudhuri 和 Ruslan Salakhutdinov（编）所编的 第*36*届国际机器学习会议论文集，*ICML 2019*，*2019*年*6*月*9*日至*15*日，美国加利福尼亚州长滩中，第97卷，机器学习研究论文集，第773–782页。PMLR，2019年。

Tom B. Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhari- wal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel M. Ziegler、Jeffrey Wu、Clemens Winter、Christopher Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。语言模型是少样本学习者。在 Hugo Larochelle、Marc'Aurelio Ranzato、Raia Hadsell、Maria-Florina Balcan 和 Hsuan-Tien Lin（编）所编的 神经信息处理系统进展*33*：神经信息处理系统年度会议论文集*2020*，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上中，2020年。

Can Chen、Christopher Beckham、Zixuan Liu、Xue (Steve) Liu 和 Chris Pal。用于离线模型优化的并行指导。载于 Alice Oh、Tristan Naumann、Amir Globerson、Kate Saenko、Moritz Hardt 和 Sergey Levine（编），神经信息处理系统进展第*36*卷：*2023*年神经信息处理系统年会，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023*年*12*月*10*日至*16*日，2023年。

Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Amy Yang、Angela Fan 等。《Llama 3 模型家族》。*ArXiv* 预印本，第 arXiv–2407 页，2024年。

Justin Fu 和 Sergey Levine。通过归一化最大似然估计进行离线模型优化。载于第九届国际学习表征会议，*ICLR 2021*，线上活动，奥地利，*2021*年*5*月*3*日至*7*日。OpenReview.net，2021年。

Shansan Gong、Ruixiang Zhang、Huangjie Zheng、Jiatao Gu、Navdeep Jaitly、Lingpeng Kong 和 Yizhe Zhang。Diffucoder：理解并改进用于代码生成的掩码扩散模型。*ArXiv* 预印本，abs/2506.20639，2025年。

Kam Hamidieh。一种用于预测超导体临界温度的数据驱动统计模型。计算材料科学，2018年。

Nikolaus Hansen. 协方差矩阵自适应进化策略：一篇比较综述。迈向新的进化计算：分布估计算法的进展，2006年。

Minh Hoang、Azza Fadhel、Aryan Deshwal、Jana Doppa 和 Trong Nghia Hoang。通过梯度匹配学习离线黑盒优化的代理模型。发表于第四十一届国际机器学习大会，*ICML 2024*，奥地利维也纳，*2024*年*7*月*21*日至*27*日。OpenReview.net，2024年。

Vineet Jain、Kusha Sareen、Mohammad Pedramfar 和 Siamak Ravanbakhsh。扩散树采样：扩散模型的可扩展推理时对齐。*ArXiv*预印本，abs/2506.20701，2025年。

Minsu Kim、Jiayao Gu、Ye Yuan、Taeyoung Yun、Zixuan Liu、YoshuaBengio 和 CanChen。离线模型优化：全面综述。*ArXiv*预印本，abs/2503.17286，2025年。

Levente Kocsis 和 Csaba Szepesv´ari。基于老虎机的蒙特卡洛规划。载于欧洲机器学习会议，第282–293页。Springer，2006年。

Siddarth Krishnamoorthy、Satvik Mehul Mashkaria 和 Aditya Grover。用于黑箱优化的扩散模型。载于 Andreas Krause、Emma Brunskill、Kyunghyun Cho、Barbara Engelhardt、Sivan Sabato 和 Jonathan Scarlett（编），国际机器学习会议，*ICML 2023*，*2023*年*7*月*23*日至*29*日，檀香山，夏威夷，美国，第202卷，机器学习研究论文集，第17842–17857页。PMLR，2023年。

Aviral Kumar 和 Sergey Levine。用于基于模型的优化的模型反演网络。载于 Hugo Larochelle、Marc' Aurelio Ranzato、Raia Hadsell、Maria-Florina Balcan 和 Hsuan-Tien Lin（编），神经信息处理系统进展*33*：神经信息处理系统年度会议 *2020*，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上，2020年。

Mengdi Liu、Xiaoxue Cheng、Zhangyang Gao、Hong Chang、Cheng Tan、Shiguang Shan 和 Xilin Chen。Protinvtree：通过奖励引导树搜索审慎地进行蛋白质逆折叠。*ArXiv*预印本，abs/2506.00925，2025年。

Shengcai Liu、Caishun Chen、Xinghua Qu、Ke Tang 和 Yew-Soon Ong。作为进化优化器的大语言模型。载于*2024*年 *IEEE*进化计算大会 *(CEC)*，第 1–8 页。IEEE，2024年。

David JC MacKay 等人。高斯过程简介。北约高级研究院系列 *F* 计算机与系统科学，168:133–166，1998年。

Satvik Mehul Mashkaria、Siddarth Krishnamoorthy 和 Aditya Grover。面向黑盒优化的生成式预训练。载于 Andreas Krause、Emma Brunskill、Kyunghyun Cho、Barbara Engel- hardt、Sivan Sabato 和 Jonathan Scarlett（编），国际机器学习会议，*ICML 2023*，*2023*年*7*月*23*日至*29*日，美国夏威夷檀香山，机器学习研究论文集第202卷，第24173–24197页。Pmlr，2023年。

Tung Nguyen 和 Aditya Grover。Lico：用于上下文分子优化的大语言模型化。*ArXiv*预印本，abs/2406.18851，2024年。

Tung Nguyen、Sudhanshu Agrawal 和 Aditya Grover。Expt：面向少样本实验设计的合成预训练。载于 Alice Oh、Tristan Naumann、Amir Globerson、Kate Saenko、Moritz Hardt 和 Sergey Levine（编），神经信息处理系统进展第*36*卷：*2023*年神经信息处理系统年会，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023*年*12*月*10*日至*16*日，2023年。

Tung Nguyen、Qiuyi Zhang、Bangding Yang、Chansoo Lee、Jorg Bornschein、Yingjie Miao、Sagi Perel、Yutian Chen 和 Xingyou Song。从字符串进行预测：用于贝叶斯优化的语言模型嵌入。*ArXiv*预印本，abs/2410.10190，2024年。

Allen Nie、Ching-An Cheng、Andrey Kolobov 和 Adith Swaminathan。定向反馈对基于大语言模型的优化器的重要性。*ArXiv*预印本，abs/2405.16434，2024年。

Shen Nie、Fengqi Zhu、Chao Du、Tianyu Pang、Qian Liu、Guangtao Zeng、Min Lin 和 Chongxuan Li。在《第十三届国际学习表征会议》上发表，2025a。

Shen Nie、Fengqi Zhu、Zebin You、Xiaolu Zhang、Jingyang Ou、Jun Hu、Jun Zhou、Yankai Lin、Ji- Rong Wen 和 Chongxuan Li。大语言扩散模型。*ArXiv*预印本，abs/2502.09992，2025b。

Alexander Novikov、NgˆanV˜u、Marvin Eisenberger、Emilien Dupont、Po-SenHuang、AdamZsolt Wagner、Sergey Shirobokov、Borislav Kozlovskii、Francisco JR Ruiz、Abbas Mehrabian 等。Alphaevolve：用于科学和算法发现的编码代理。*ArXiv*预印本，abs/2506.13131，2025年。

Colin Raffel、NoamShazeer、Adam Roberts、KatherineLee、SharanNarang、MichaelMatena、Yanqi Zhou、Wei Li 和 Peter J. Liu。探索采用统一文本到文本转换器的迁移学习极限。《机器学习研究杂志》，21:140:1–140:67，2020年。

Paul J Sample、Ban Wang、David W Reid、Vlad Presnyak、Iain J McFadyen、David R Morris 和 Georg Seelig。基于大规模并行翻译测定的人类5' UTR设计和变异效应预测。《自然·生物技术》，2019年。

David Silver、Aja Huang、Chris J Maddison、Arthur Guez、Laurent Sifre、George Van Den Driessche、Julian Schrittwieser、Ioannis Antonoglou、Veda Panneershelvam、Marc Lanctot 等。使用深度神经网络和树搜索掌握围棋。《自然》，529(7587):484–489，2016年。

Xingyou Song、Yingtao Tian、Robert Tjarko Lange、Chansoo Lee、Yujin Tang 和 Yutian Chen。立场：利用基础模型进行黑盒优化。载于 第四十一届机器学习国际会议，*ICML 2024*，奥地利维也纳，*2024*年*7*月*21*日至*27*日。OpenReview.net，2024年。

Rong-Xi Tan、Ming Chen、Ke Xue、Yao Wang、Yaoyuan Wang、Sheng Fu 和 Chao Qian。通过学习语言模型嵌入实现通用离线黑盒优化。*ArXiv*预印本，abs/2506.07109，2025年。

Eric Tang、Bangding Yang 和 XingyouSong。理解用于回归的大语言模型嵌入。*ArXiv*预印本，abs/2411.14708，2024年。

Sophia Tang、Yinuo Zhang 和 Pranam Chatterjee。Peptune：通过多目标引导离散扩散从头生成治疗性肽。*ArXiv*，第 arXiv–2412 页，2025年。

Brandon Trabucco、Aviral Kumar、Xinyang Geng 和 Sergey Levine。用于有效离线模型优化的保守目标模型。载于 Marina Meila 和 Tong Zhang（编），第*38*届国际机器学习会议论文集，*ICML 2021*，*2021*年*7*月*18*日至*24*日，线上活动，机器学习研究论文集第139卷，第10358–10368页。PMLR，2021年。

Brandon Trabucco、Xinyang Geng、Aviral Kumar 和 Sergey Levine。design-bench：数据驱动的离线基于模型的优化基准测试。载于 Kamalika Chaudhuri、Stefanie Jegelka、Le Song、Csaba Szepesv´ari、Gang Niu 和 Sivan Sabato（编），国际机器学习会议，*ICML 2022*，*2022*年*7*月*17*日至*23*日，美国马里兰州巴尔的摩，机器学习研究论文集第162卷，第21658–21676页。PMLR，2022年。

Petar Veli˘ckovi´c、Alex Vitvitskyi、Larisa Markeeva、Borja Ibarz、Lars Buesing、Matej Balog 和 Alexander Novikov。增强人类在组合竞争性编程中的表现。*ArXiv* 预印本，abs/2411.19744，2024 年。

Linnan Wang、Rodrigo Fonseca 和 Yuandong Tian。使用蒙特卡洛树搜索学习用于黑盒优化的搜索空间划分。载于 Hugo Larochelle、Marc'Aurelio Ranzato、Raia Hadsell、Maria-Florina Balcan 和 Hsuan-Tien Lin（编），神经信息处理系统进展 *33*：神经信息处理系统年会 *2020*，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上，2020年。

Shukuan Wang、Ke Xue、Song Lei、Xiaobin Huang 和 Chao Qian。基于蒙特卡洛树搜索的黑盒优化空间迁移。载于 Amir Globersons、Lester Mackey、Danielle Belgrave、Angela Fan、Ulrich Paquet、Jakub M. Tomczak 和 Cheng Zhang（编），《神经信息处理系统进展》第*38*卷：*2024*年神经信息处理系统年会，*NeurIPS 2024*，温哥华，不列颠哥伦比亚省，加拿大，*2024*年*12*月*10*日至*15*日，2024年。

Chengrun Yang、Xuezhi Wang、Yifeng Lu、Hanxiao Liu、Quoc V. Le、Denny Zhou 和 Xinyun Chen。作为优化器的大语言模型。载于第十二届国际学习表征会议，*ICLR 2024*，维也纳，奥地利，*2024*年*5*月*7*日至*11*日。OpenReview.net，2024年。

Ling Yang、YeTian、BowenLi、Xinchen Zhang、KeShen、Yunhai Tong 和 MengdiWang。Mmada：多模态大型扩散语言模型。*arXiv*预印本 *arXiv:2505.15809*，2025年。

Jaesik Yoon、Hyeonseo Cho、Doojin Baek、Yoshua Bengio 和 Sungjin Ahn。用于系统2规划的蒙特卡洛树扩散。*ArXiv*预印本，abs/2502.07202，2025a。

Jaesik Yoon、Hyeonseo Cho、Yoshua Bengio 和 Sungjin Ahn。快速蒙特卡洛树扩散：通过并行稀疏规划实现100倍加速。*ArXiv*预印本，abs/2506.09498，2025b。

Sihyun Yu、Sungsoo Ahn、Le Song 和 Jinwoo Shin。Roma：用于离线基于模型优化的鲁棒模型适应。载于 Marc'Aurelio Ranzato、Alina Beygelzimer、Yann N. Dauphin、Percy Liang 和 Jennifer Wortman Vaughan（编），神经信息处理系统进展第*34*卷：神经信息处理系统年会*2021*，*NeurIPS 2021*，*2021*年*12*月*6*日至*14*日，线上，第4619–4631页，2021年。

Ye Yuan、Can Chen、Zixuan Liu、Willie Neiswanger 和 Xue (Steve) Liu。面向离线模型优化的重视重要性的协同教学。载于 Alice Oh、Tristan Naumann、Amir Globerson、Kate Saenko、Moritz Hardt 和 Sergey Levine（编），神经信息处理系统进展第*36*卷：*2023*年神经信息处理系统年会，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023*年*12*月*10*日至*16*日，2023年。

Taeyoung Yun、Sujin Yun、Jaewoo Lee 和 Jinkyoo Park。利用扩散模型为离线模型优化生成引导式轨迹。载于 Amir Globersons、Lester Mackey、Danielle Belgrave、Angela Fan、Ulrich Paquet、Jakub M. Tomczak 和 Cheng Zhang（编），神经信息处理系统进展 *38*：*2024*年神经信息处理系统年会，*NeurIPS 2024*，加拿大不列颠哥伦比亚省温哥华，*2024*年*12*月*10*日至*15*日，2024年。

Michael R Zhang、Nishkrit Desai、Juhan Bae、Jonathan Lorraine 和 Jimmy Ba。使用大型语言模型进行超参数优化。*ArXiv*预印本，abs/2312.04528，2023年。

Tao Zhang、Jia-Shu Pan、Ruiqi Feng 和 Tailin Wu。T-scend：测试时可扩展的蒙特卡洛树搜索增强型扩散模型。*arXiv* 预印本，第 arXiv–2502 页，2025年。

Siyan Zhao、Tung Nguyen 和 Aditya Grover。《探究大语言模型中上下文学习的决策边界》。载于 Amir Globersons、Lester Mackey、Danielle Belgrave、Angela Fan、Ulrich Paquet、Jakub M. Tomczak 和 Cheng Zhang（编），神经信息处理系统进展 *38*：神经信息处理系统 *2024* 年会，*NeurIPS 2024*，加拿大不列颠哥伦比亚省温哥华，*2024*年*12*月*10*日至*15*日，2024年。

A 附录

A.1 大语言模型的使用

我们仅使用大语言模型协助润色本文写作。具体而言，我们使用大语言模型修改措辞并检查语法，以提升文章的清晰度和可读性。方法设计、实现或实验结果分析均未借助大语言模型。

A.2 上下文提示

为确保内容清晰且便于复现，我们提供了实验中使用的上下文提示。

Ant 的上下文提示

你是一位乐于助人的优化助手，将帮助我们生成新的机器人形态设计。目标是优化仿真机器人 Ant（来自 OpenAI Gym）的形态结构。对于 Ant 形态，我们需要优化四足机器人的形态，使其尽可能快速地奔跑。

对于每项设计，我们有以下信息，即形态参数的 60 个连续值，分为 4 条腿，每条腿有 15 个参数：x：髋关节的 x 坐标，y：髋关节的 y 坐标，z：髋关节的 z 坐标，a：髋关节的角度，b：大腿关节的角度，c：踝关节的角度，髋关节中心：髋关节中心的位置，髋关节范围：髋关节的范围，大腿关节中心：大腿关节中心的位置，大腿关节范围：大腿关节的范围，踝关节中心：踝关节中心的位置，踝关节范围：踝关节的范围，髋关节尺寸：髋关节的尺寸，大腿关节尺寸：大腿关节的尺寸，踝关节尺寸：踝关节的尺寸

给定以下现有设计及其对应的性能评分：

机器人形态设计： [0.01, 0.01, 0.01, 15.03, 15.02, 58.04, -24.89, 4.57, -16.5, 28.86, 61.91, 23.22, 0.2, 0.21, 0.38, 0.01, 0.0, 0.01, 12.08, 27.69, 108.36, -0.24, 8.41, 5.2, 31.64, 41.59, 18.12, 0.22, 0.19, 0.36, 0.0, 0.0, -0.01, -14.78, 2.05, 195.67, 8.69, 2.64, -3.88, 29.51, 32.58, 19.1, 0.19, 0.2, 0.4, 0.01, -0.0, -0.02, 35.92, -12.99, -61.83, -20.47, 4.67, -24.25, 29.19, 22.72, 19.29, 0.22, 0.21, 0.43],性能得分：-386.9

...
机器人形态设计： [-0.01, -0.0, 0.0, 7.64, -7.49, 1.64, 10.86, 5.26, 25.13, 28.7, 33.16, 21.44, 0.19, 0.2, 0.44, -0.02, -0.01, 0.01, 8.55, 7.52, 78.95, -11.6, 5.98, 13.54, 29.48, 41.69, 19.64, 0.2, 0.23, 0.38, -0.0, -0.01, 0.0, -13.43, -2.04, 170.37, 9.33, 6.51, -0.43, 31.12, 54.72, 22.07, 0.19, 0.18, 0.4, 0.0, -0.0, -0.0, 5.91, -20.12, -92.43, 17.22, 6.47, 10.4, 30.79, 45.95, 18.89, 0.18, 0.21, 0.44], 性能得分：165.33

请提出一种新的机器人形态设计，以最大化性能得分。每项特征都应为浮点数，且每个数值都应四舍五入到小数点后两位。

D’Kitty 的上下文提示

你是一位乐于助人的优化助手，将协助我们生成一种新的机器人形态设计。我们的目标是优化仿真机器人 D’Kitty 的形态结构。对于 D’Kitty 形态，我们旨在优化机器人的身体和腿部结构，最大限度地提升其运动能力，使机器人能够移动到固定位置。

对于每个设计，我们都有以下信息，包括 56 个连续形态参数，分为 4 条腿，每条腿有 14 个参数：x：髋关节的 x 坐标 y：髋关节的 y 坐标 z：髋关节的 z 坐标 a：髋关节的角度 b：膝关节的角度 髋关节中心：髋关节的中心 髋关节范围：髋关节的范围 膝关节中心：膝关节的中心 膝关节范围：膝关节的范围 髋关节尺寸：髋关节的尺寸 膝关节尺寸：膝关节的尺寸 足部关节中心：足部关节的中心 足部关节范围：足部关节的范围 足部关节尺寸：足部关节的尺寸

现有设计及其对应的性能评分如下：

机器人形态设计： [0.11, 0.14, 0.0, 0.46, -2.88, 0.14, -0.34, 0.44, 0.37, 0.79, -1.21, 1.19, 0.09, 0.1, -0.09, 0.11, 0.0, -0.35,3.76, -0.25,0.23,0.4, 0.58, 0.79, -0.31, 1.03, 0.1, 0.1, -0.09, -0.11, 0.0, 0.26, 2.66, 0.18, 0.47, 0.34, 0.45, 0.72, -0.64, 1.04, 0.1, 0.09, 0.11, -0.1, 0.0, -0.51, -2.27, 0.26, -1.03, 0.38, 0.43, 0.76, -0.9, 0.93, 0.09, 0.09], 性能得分：-880.46

...

机器人形态设计： [0.1, 0.13, 0.0, -0.04, -3.55, -0.09, -0.17, 0.42, 0.51, 0.78, -1.24, 0.99, 0.1, 0.1, -0.11, 0.11, 0.0, -0.03, 4.01, 0.2, 0.23, 0.43, 0.73, 0.82, -0.33, 0.98, 0.1, 0.1, -0.09, -0.12, 0.0, 0.06, 3.37, -0.03, 0.36, 0.31, 0.61, 0.61, -0.78, 0.92, 0.1, 0.09, 0.1, -0.09, 0.0, -0.22, -2.74, -0.18, -0.71, 0.42, 0.32, 0.76, -0.43, 0.94, 0.1, 0.09], 性能得分：199.36

请提出一种新的机器人形态设计，以最大化性能得分。每项特征都应为浮点数，且每个数值应四舍五入到小数点后两位。

TF8 的上下文提示

你是一位乐于助人的优化助手，将帮助我们生成一条与特定转录因子SIX6 REF R1具有最高结合亲和力的长度为8的最优DNA序列。

现提供以下已有DNA序列及其对应的结合亲和力：

DNA序列： [’G’, ’G’, ’C’, ’C’, ’G’, ’G’, ’C’, ’C’], 结合亲和力：0.0

... DNA序列： [’G’, ’T’, ’G’, ’G’, ’G’, ’C’, ’G’, ’A’],结合亲和力：0.44

请提出一个新的DNA序列，该序列应与现有DNA序列不同，并且具有比现有DNA序列更高的结合亲和力。DNA序列应采用 A、C、G、T 的格式。新DNA序列至少有一个位置与现有DNA序列不同。

表3：用于比较的四项任务第50百分位归一化分数实验结果。

方法	蚂蚁形态	D'Kitty 形态	TF 绑定 8	TF 绑定 10	平均排名	排名中位数
$\mathcal{D}$（最佳）	0.565	0.884	0.439	0.467	–	–
梯度均值	0.378 ± 0.006	0.888 ± 0.001	0.439 ± 0.000	0.477 ± 0.000	5.8/16	5.0/16
Grad-EI	0.373 ± 0.008	0.902 ± 0.007	$\underline{0.451} \pm \underline{0.007}$	0.450 ± 0.005	6.8/16	7.0/16
COMs	0.372 ± 0.012	0.896 ± 0.004	0.439 ± 0.000	0.473 ± 0.008	6.5/16	5.0/16
ICT	0.384 ± 0.001	$\underline{0.908} \pm \underline{0.001}$	0.395 ± 0.009	0.425 ± 0.009	8.8/16	8.5/16
MATCH-OPT	0.377 ± 0.007	0.888 ± 0.003	0.436 ± 0.001	0.433 ± 0.000	9.3/16	9.0/16
UniSO-T	$\mathbf{0.400 \pm 0.003}$	0.906 ± 0.004	0.443 ± 0.005	0.457 ± 0.006	3.8/16	3.0/16
CbAS	0.351 ± 0.001	0.893 ± 0.001	0.390 ± 0.001	0.431 ± 0.005	12.8/16	13.5/16
ExPT	$\mathbf{0.400 \pm 0.008}$	0.903 ± 0.002	0.350 ± 0.000	0.464 ± 0.005	6.8/16	5.5/16
MIN	0.366 ± 0.001	0.861 ± 0.004	0.328 ± 0.008	0.448 ± 0.009	13.5/16	14.0/16
BONET	0.389 ± 0.006	0.895 ± 0.004	0.413 ± 0.009	0.472 ± 0.007	6.3/16	6.0/16
ORPO	0.362 ± 0.003	0.787 ± 0.004	0.339 ± 0.005	0.399 ± 0.003	11.5/16	12.5/16
GTG	0.374 ± 0.013	0.888 ± 0.000	0.418 ± 0.013	0.443 ± 0.011	9.8/16	9.5/16
DDOM	0.348 ± 0.008	0.886 ± 0.001	0.401 ± 0.003	0.448 ± 0.006	12.5/16	12.0/16
CMA-ES	0.368 ± 0.002	0.695 ± 0.003	0.426 ± 0.003	0.471 ± 0.002	10.5/16	10.0/16
MCTS-transfer	0.374 ± 0.006	0.895 ± 0.008	0.359 ± 0.007	$\underline{0.478} \pm \underline{0.004}$	8.0/16	8.0/16
dLLM(ours)	$\underline{0.398} \pm \underline{0.014}$	$\mathbf{0.910 \pm 0.021}$	$\mathbf{0.463 \pm 0.015}$	$\mathbf{0.482 \pm 0.009}$	**1.5/16**	**1.0/16**

TF10 的上下文提示

你是一位乐于助人的优化助手，将帮助我们生成一个长度为 10、与特定转录因子 Pho4 具有最大结合亲和力的最优DNA序列。

给定以下已有的 DNA 序列及其对应的结合亲和力：

DNA序列： ['T', 'C', 'C', 'A', 'C', 'G', 'A', 'A', 'G', 'A'], 结合亲和力：-1.86
...
DNA序列： ['G', 'C', 'T', 'T', 'G', 'G', 'A', 'A', 'C', 'A'], 结合亲和力：0.01

请提出一个新的 DNA 序列，该序列应不同于现有 DNA 序列，且结合亲和力高于现有 DNA 序列。DNA序列的格式应为 A、C、G、T。新 DNA 序列至少应有一个位置不同于现有 DNA 序列。

A.3 中位数结果

我们还在表3中报告了第50百分位归一化分数，其中最佳和次佳结果分别以**粗体**和下划线标出。我们的方法 *dLLM* 取得了最佳的总体平均
并在16种方法中位列中位排名第一，在 D'Kitty、TF8 和 TF10 上排名第一，在 Ant 上排名第三。这表明 *dLLM* 始终保持着出色的性能。

A.4 不同的任务描述

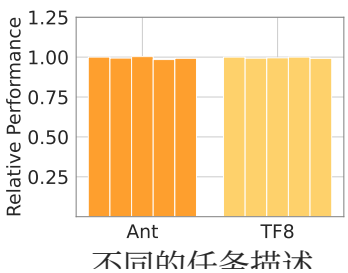

图6：不同任务描述下的相对性能。

我们通过用五种由 GPT-5 生成的模板替换默认任务描述，评估 dLLM 对上下文提示变化的稳健性。如图6所示，归一化分数保持稳定——在 Ant 上介于 0.642 至 0.654 之间，在 TF8 上介于 0.869 至 0.876 之间——这表明模型能够稳健应对任务描述的风格变化。