arXiv:2602.01090v1 [cs.AI] 1 Feb 2026

硬约束与软生成相遇：确保基于大语言模型的组合优化具有可行性

Yang Liu [1] **Chuan Zhou** [1 2] **Yancheng Chen** [1] **Shuai Zhang** [1] **Xixun Lin** [3] **Xiaoqing Wang** [4]

摘要

大语言模型（LLM）已成为颇具潜力的通用组合优化（CO）求解器，但从根本上缺乏确保解可行性的机制，而这对实际部署至关重要。在本文中，我们提出 FALCON 框架，通过三项关键创新确保 100% 的可行性：（i）语法约束解码确保语法有效性；（ii）可行性修复层 纠正语义约束违反；（iii）自适应 *Best-of-N* 采样高效分配推理计算资源。为训练底层大语言模型，我们在大语言模型训练中提出 Best-anchored Objective-guided Preference Optimization（BOPO），根据目标差距对偏好对加权，从而在无需人工标注的情况下提供密集监督信号。理论上，我们证明了 BOPO 的收敛性，并给出了修复导致的质量损失上界。实验结果表明，在七个 NP难组合优化问题上，FALCON 实现了完美可行性，同时达到或超越最先进的神经网络及基于大语言模型的求解器的解质量。

1. 引言

组合优化（CO）是众多关键现实系统的基础，涵盖物流规划（Bao 等，2018）和制造业调度（Zhang 等，2023），以及应急资源分配（Jain 和 Bharti，2023）。这类问题需要在指数级庞大的搜索空间中选择最优配置，同时满足复杂约束。传统上，解决这类问题需要依赖特定领域的启发式方法或精确算法（Liu 等，2024d；2023；2024c），这需要大量专家工程投入，并且缺乏跨问题领域的普适性。

大语言模型（LLM）的出现带来了范式转变。通过将自然语言问题描述直接映射为解序列，大语言模型为组合优化提供了统一且易于使用的接口（Jiang 等，2025）。近期研究表明，大语言模型具有出色的模式识别和序列生成能力；通过监督学习或强化学习进行微调后，它们在各种组合优化问题上取得了有竞争力的性能（Jiang 等，2025）。这表明，迈向通用、由语言驱动的优化是一条颇具前景的道路。

然而，一个根本性的矛盾依然存在：大语言模型本质上是不受约束的生成模型。面对哈密顿回路、容量限制或优先关系等严格的组合约束时，它们缺乏任何内在机制来保证满足约束。现有方法将可行性视为软目标，在训练过程中通过奖励塑形加以激励，却从未在推理阶段强制执行。因此，即使对于复杂度适中的问题，单样本可行率也会大幅波动，且仍低于 100%（Jiang 等人，2025年）。对于物流（McGarvey 等人，2025年）、制造业（Meurer 等人，2025年）或其他高风险领域中的安全关键型部署，这种不可靠性是不可接受的。因此，核心科学挑战十分明确：如何在不牺牲问题求解能力的前提下，赋予表达能力强的生成模型可验证的、*100%* 满足约束的保证？

这一挑战体现在两个层面。首先，大语言模型可能生成语法有效但语义上不可行的输出——格式正确的车辆路径可能超出容量，或者独立集可能包含相邻顶点。仅凭语法无法编码每类问题固有的语义约束。其次，训练时的激励（通过奖励或偏好）并不能转化为推理时保证。模型可能学会经常生成可行解，但无法被强制如此，从而在概率性激励与确定性可靠性之间造成关键差距。

为弥合这一差距，我们提出 FALCON（可行性-

[1]中国科学院数学与系统科学研究院 [2]中国科学院大学网络空间安全学院 [3]中国科学院信息工程研究所 [4]阿里巴巴集团。通讯作者：Chuan Zhou <zhouchuan@amss.ac.cn>。

预印本。工作仍在进行中。

感知的基于语言的组合优化与自适应推理），这是首个具有可证明100% 可行性保证的基于大语言模型的组合优化框架。FALCON 建立在一项关键概念洞见之上：将挑战分离为语法有效性 与语义可行性，并以专门且可验证的层分别加以解决。具体而言，我们整合了三个协同组件：（1）语法约束解码，借助问题特定的上下文无关文法确保输出格式正确性；（2）可行性修复算子，将任何违反约束的解转化为可行解；（3）自适应 *Best-of-*N采样，根据实例难度动态分配推理算力。这些组件采用最佳锚定的目标函数引导偏好优化（BOPO）进行训练；这是一种新颖的目标函数，按目标值差距为偏好对加权，从而提供稠密的自动监督。

我们的贡献如下：

1. 我们提出了 FALCON，这是首个基于大语言模型的组合优化求解器，通过由文法约束、语义修复和自适应推理构成的分层架构，保证 100% 的可行性。我们给出了关于可行性、格式有效性以及修复后解质量退化界限的形式化证明。2. 我们将 BOPO 引入大语言模型训练，这是一种将目标引导加权融入偏好优化的原则性训练方法。我们证明了其在标准假设下的 $O(-1/\sqrt{T})$ 收敛性，并展示了其相较于奖励塑形和优势归一化的优越性。3. 我们针对涵盖路径规划、图和调度领域的七个 NP难问题进行了广泛评估。结果表明，FALCON 在保持有竞争力的最优性差距的同时，实现了稳定的 100% 可行性，表现优于通用大语言模型和近期的神经组合优化求解器。

2. 预备知识

2.1. 问题表述

令 $\mathcal{P}$ 表示一类组合优化问题。每个实例 $p \in \mathcal{P}$ 定义为一个元组 $(\mathcal{X}_p,\ f_p,\ \mathcal{C}_p)$，其中 $\mathcal{X}_p$ 是包含所有格式规范的解表示的解空间，$f_p\colon \mathcal{X}_p \to \mathbb{R}$ 是需要最小化的目标函数，而 $\mathcal{C}_p = \{c_{p,1}, \ldots, c_{p,m_p}\}$ 是一组约束，其中每个 $c_{p,i}\colon \mathcal{X}_p \to \{0,1\}$ 用于指示某个解是否满足约束 i。

定义 2.1 (可行解)。 解 $x \in \mathcal{X}_p$ 是可行的，当且仅当它满足所有约束：$\forall c_{p,i} \in \mathcal{C}_p : c_{p,i}(x) = 1$。可行域为：

$$\mathcal{X}_{\mathcal{C}_p} = \{x \in \mathcal{X}_p \mid \forall c_{p,i} \in \mathcal{C}_p : c_{p,i}(x) = 1\}.$$

优化目标是求 $x_p^* = \arg\min_{x \in \mathcal{X}_{\mathcal{C}_p}} f_p(x)$。

2.2. 大语言模型作为端到端求解器

令 $\pi_\theta\colon \mathcal{T} \to \mathcal{T}$ 为一个参数为 θ的大语言模型，其中$\mathcal{T}$是文本序列空间。我们定义两个映射函数：$\phi\colon \mathcal{P} \to \mathcal{T}$ 将问题实例映射为文本描述，而 $\psi_p\colon \mathcal{T} \rightharpoonup \mathcal{X}_p$ 是一个偏函数，当文本序列符合有效语法时，将其解析为解。端到端解生成流程尝试生成 $\hat{x}_p = \psi_p(\pi_\theta(\phi(p)))$，只有当生成文本可解析且解析所得的解可行时，流程才会成功，即 $\hat{x}_p \in \mathcal{X}_{\mathcal{C}_p}$。

3. 基于大语言模型的组合优化框架：FALCON

3.1. 语法约束解码

可行性保证的第一层确保格式有效性——输出必须在语法上正确，括号配对完整、分隔符有效，且索引位于有效范围内。

3.1.1. 问题特定文法

我们为每个组合优化问题定义上下文无关文法（CFG），以精确规定有效的输出格式。

定义 3.1(组合优化输出文法)。 组合优化输出文法是一个元组 $G = (V, \Sigma, R, S)$，其中 V 是非终结符集合（例如，NODELIST、NUMBER），Σ是对应于大语言模型词汇表中词元的终结符集合，R 是规定有效推导的产生式规则集合，S 是开始符号。

定义 3.2 （输入相关文法）。 例如，对于一个包含 n 个节点的实例 p，输入相关文法 $G_p = (V, \Sigma_p, R_p, S)$ 通过限制节点索引的产生式规则，使其仅生成有效索引 $\{0, 1, \ldots, n-1\}$，从而避免越界引用，进而特化基础文法 G。

七个问题的完整文法规范见第 D 节。

3.1.2. 约束解码算法

在每个解码步骤中，我们维护一个与当前解析状态对应的下推自动机（PDA）配置，并屏蔽会导致无效后续内容的词元。对于给定的文法 G，我们使用标准的 CFG 到 PDA 的转换，构造一个等价的 PDA $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$，使其能够识别 $\mathcal{L}(G)$。

定理 3.3 （格式有效性保证）。 对于算法 y *1* 生成的任何输出，当其在文法G下终止时，我们有 $y \in \mathcal{L}(G)$ 其中 $\mathcal{L}(G)$ 是由 G 定义的语言。

备注 3.4 （计算开销）。 每个词元的开销为

算法 1 语法约束解码

输入： 文法 G、提示词 x、大语言模型 π_θ
输出： 有效解文本 $y \in \mathcal{L}(G)$

从 $\mathcal{A}$ $y \leftarrow$ "" 初始化下推自动机配置 $(q_0,\ Z_0)$
当不处于接受状态**时**
  逻辑值 $\leftarrow \pi_\theta(x,\ y)$ // 前向传播
  $\mathcal{V}_{\text{valid}} \leftarrow \{v \in \Sigma \mid \delta(q, v, \text{top}(\gamma)) \neq \emptyset\}$
  对于每个 $v \notin \mathcal{V}_{valid}$ **执行**
    逻辑值$[v] \leftarrow -\infty$ // 屏蔽无效词元
  $v_{\text{next}} \leftarrow$采样(softmax(逻辑值))
  $y \leftarrow y + v_{\text{next}}$
  通过 δ 更新下推自动机配置 $(q,\ \gamma)$
返回 y

词元为 $O(|\Sigma| \cdot |Q|)$，通过下推自动机转移查找计算有效词元，其中 $|\Sigma|$ 是词汇表大小，$|Q|$是下推自动机状态的数量。对于结构简单的组合优化文法，$|Q| = O(1)$，因此每个词元的开销为 $O(|\Sigma|)$。对于现代大语言模型，$|\Sigma| \approx 32K$，与 $O(d^2)$ 注意力计算相比，这可以忽略不计，其中 d 是隐藏维度。

3.2. 可行性修复层

文法约束确保格式有效性，但不保证语义可行性。语法有效的容量约束车辆路径问题路线可能超出车辆容量；格式良好的最大独立集可能包含相邻顶点。修复层旨在应对这一挑战。

3.2.1. 修复算子设计

定义 3.5（修复算子）。修复算子 $\mathcal{R}: \mathcal{X}_p \to \mathcal{X}_{C_p}$ 对于约束集 C_p 满足：
1. 可行性：$\forall x \in \mathcal{X}_p$：$\mathcal{R}(x) \in \mathcal{X}_{C_p}$*2.* 幂等性：$\forall x \in \mathcal{X}_{C_p}$：$\mathcal{R}(x) = x$*3.* 有界局部性：存在一个距离度量 $d: \mathcal{X}_p \times \mathcal{X}_p \to \mathbb{R}_{\geq 0}$、一个违反程度函数 $v: \mathcal{X}_p \to \mathbb{R}_{\geq 0}$，以及一个常数 $\alpha > 0$，使得：

$$d(\mathcal{R}(x), x) \leq \alpha \cdot v(x),$$

其中 $v(x) = 0$ 当且仅当 $x \in \mathcal{X}_{C_p}$。

性质（1）保证任意输入都被映射为一个可行解。性质（2）确保已经可行的解保持不变，从而避免不必要的质量下降。性质（3）为修改距离提供了一个定量界，该界与约束违反幅度成正比。

表*1.* 各问题类型对应的修复算子及其最坏情况下的时间复杂度。这里 n 表示节点/作业的数量，m 表示机器的数量，$|E|$ 表示边的数量。

问题	修复策略	复杂度
TSP	删除重复项，并贪心插入缺失节点	$O(n^2)$
CVRP	对每条路径进行 TSP修复 + 拆分超载路径	$O(n^2)$
OP	移除奖励/距离比最差的节点†	$O(n$日志 $n)$
MIS	从每个冲突中移除较高度数顶点	$O(\|E\| \cdot n)$
MVC	添加未覆盖边中度数较高的端点	$O(\|E\|)$
PFSP	通过贪心插入最小化最大完工时间	$O(n^2m)$
JSSP	排列修复 +拓扑排序	$O\ (n^2m)$

†假设使用优先队列；$O\ (n^2)$ 则不使用。

3.2.2. 特定问题的修复策略

GIES

我们为每种问题类型设计了高效的修复算子，汇总于表1。核心洞见在于，每个算子针对特定的约束违反模式：排列问题（旅行商问题、置换流水车间调度问题）需要去除重复元素并插入缺失元素；容量问题（容量约束车辆路径问题、定向越野问题）需要拆分路径或移除节点；图问题（最大独立集、最小顶点覆盖）则基于度进行贪心的顶点添加或移除。详情见附录E。

3.2.3. 理论保证

定理 3.6（100% 可行性保证）。对于修复算子 $\mathcal{R}$满足定义 *3.5*时，对于任意数量的样本 $N \geq 1$:

$$\mathbb{P}(\textit{all } N \textit{ outputs are feasible}) = 1.$$

该保证可直接由定义 3.5 的性质 **(1)** 得出，该性质确保 $\mathcal{R}$ 的所有输出都位于可行域 $\mathcal{X}_{C_p}$ 中，而不受输入影响。

定理 3.7（修复质量界）。设 d：$\mathcal{X}_p \times \mathcal{X}_p \to \mathbb{R}_{\geq 0}$ 是定义在解上的距离度量，v：$\mathcal{X}_p \to \mathbb{R}_{\geq 0}$表示约束违反幅度（例如，容量溢出、冲突数量）。假设修复算子满足 d（$\mathcal{R}$（x） x） $\leq \alpha \cdot v$（x），其中某个常数为 $\alpha > 0$，且目标函数 f_p 关于 d 是 L_f-利普希茨连续的。那么：

$$f_p(\mathcal{R}(\hat{x}_p)) \leq f_p(\hat{x}_p) + L_f \cdot \alpha \cdot v(\hat{x}_p).$$

该定理给出了一个明确界限，表明质量退化与违反程度 $v(\hat{x}_p)$ 成正比。实践中，经 BOPO 训练的模型生成的解大多可行，因此违反情况罕见且程度较小，导致的退化也微乎其微。当 $v(\hat{x}_p) = 0$（输入已可行）时，该界限变得平凡，且性质 **(2)** 确保不会发生任何修改。

推论 3.8 (修复的样本效率)。若仅通过采样（不进行修复）达到可行性速率 $1-\delta$，

所需的期望样本数为：

$$\mathbb{E}[N] = \frac{1}{p_f} \quad \textit{(rejection sampling)},$$

或者针对置信度为 $1-\delta$ 的固定预算采样：

$$N \geq \frac{\log(1/\delta)}{\log(1/(1-p_f))} = \frac{\log(1/\delta)}{p_f} + O(p_f) \quad \textit{as } p_f \to 0,$$

其中，p_f 是单样本可行率。采用修复时，$N=1$ 即可实现 *100%* 的可行率，在达到目标置信度方面可提速至多~~—~~$\log(1/\delta)\ p_f$。

3.3. 自适应 Best-of-N 采样

固定-N 采样效率低下：简单实例可能只需采样一次，而困难实例则可能受益于广泛探索。我们提出一种基于解一致性的自适应采样方法。

3.3.1. 基于一致性的难度估计

定义 3.9(解一致性)。 给定从模型中独立采样得到的 $K \geq 2$ 解 $\{y_1, \ldots, y_K\}$，一致性度量定义为：

$$\textit{Cons} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i} \mathbb{1}[y_i = y_j],$$

即计算有序对 (y_i, y_j) 中相同对所占的比例。

高一致性表明模型对解有把握——大多数样本意见一致，说明这是一个“简单”实例。低一致性表明模型存在不确定性——样本彼此多样，说明这是一个“困难”实例，需要更多探索。

引理 3.10(一致性-难度关系)。 对于从解空间 $\mathcal{Y}$ 上的分布 p 中独立采样得到的 K解，期望一致性满足：

$$\mathbb{E}[\textit{Cons}] = \sum_{y \in \mathcal{Y}} p(y)^2.$$

此外，将阶数为 *2* 的 *Rényi* 熵定义为$H_2(p) = -\log \sum_y p(y)^2$,，我们有：

$$\mathbb{E}[\textit{Cons}] = e^{-H_2(p)}.$$

这一结果确立了模型输出分布的集中程度与一致性之间的精确关系。低熵（集中分布）会带来高一致性，而高熵（弥散分布）

算法 2 自适应 Best-of-N 采样 输入：实例 p，边界 $N_{\min}$、$N_{\max}$，阈值 τ 输出：最佳可行解 x^* $n \leftarrow 0$, 解 $\leftarrow [\,]$ 当 $n < N_{\max}$ 时执行 从 π_θ 中采样 y_n，并使用文法约束 $y_n \leftarrow \mathcal{R}(y_n)$ // 确保可行性 solutions.append(y_n) $n \leftarrow n+1$ 如果 $n \geq N_{\min}$ 则 best $\leftarrow \arg\min_{y \in \text{solutions}} f_p(y)$ conf $\leftarrow$ 贝叶斯置信度(solutions, best) 如果 conf $\geq \tau$ 则 返回 best // 提前终止 **返回** $\operatorname{argmin}_{y \in \text{solutions}} f_p(y)$

会导致一致性较低。对于经过充分训练的模型，早期样本之间的一致性较高，表明模型对某个特定解有信心，这说明该实例较简单，无需大量采样。相反，一致性较低表明模型对最优解存在不确定性，这说明该实例较难，适合进行更多探索。

3.3.2. 自适应采样算法

算法2 使用贝叶斯置信度估计来确定何时停止采样。收集 n个样本后，使用贝塔-二项模型估计当前最佳解 y^* 在所有样本中为最优解的置信度：

$$\text{Conf}(y^*) = \frac{\alpha_0 + n_{y^*}}{\alpha_0 + \beta_0 + n},$$

其中，n_{y^*} 是与 y^*匹配的样本数量，α_0, β_0是先验参数。我们采用 $\alpha_0 = \beta_0 = 1$（均匀先验），对所有解赋予相同的先验权重。高置信度表明解已稳定收敛，从而触发提前终止。

3.3.3. 理论分析

定理 3.11 (采样带来的质量提升)。 设$\hat{}x_N$ 为从 N 个独立同分布样本中得到的最佳解，x^* 为实例 p的最优解。定义最优性差距$g_N = f_p(\hat{}x_N) - f_p(x^*)$。若单样本差距 g_1 的累积分布函数为 F，则：

$$\mathbb{E}[g_N] = \int_0^\infty (1 - F(\epsilon))^N d\epsilon.$$

该结果由期望的尾积分公式得出：$\mathbb{E}[g_N] = \int_0^\infty \mathbb{P}(g_N > \epsilon) d\epsilon$，结合 $\mathbb{P}(g_N > \epsilon) = (1 - F(\epsilon))^N$，因为 g_N 是 N个独立样本中的最小值。

推论 3.12 (指数间隙分布)。 如果差距服从速率为 $\lambda > 0$ 的指数分布，即$F(\epsilon) = 1 - e^{-\lambda\epsilon}$ 对于 $\epsilon \geq 0$，那么：

$$\mathbb{E}[g_N] = \frac{1}{N\lambda} = \frac{\mathbb{E}[g_1]}{N}.$$

也就是说，在该分布假设下，期望差距随样本数量线性下降。

定理 3.13（自适应采样复杂度）。 令 q 表示来自 π_θ 的单个样本产生差距至多为 δ 的解的概率，对于某个目标差距阈值$\delta > 0$。算法2的期望样本数满足：

$$\mathbb{E}[N_{adaptive}] \leq N_{\min} + \frac{(N_{\max} - N_{\min})(1-q)^{N_{\min}}}{q}.$$

对于 q 较大的简单实例（例如 $q = 0.8$），项（$1 - q$）$^{N_{\min}}$ 呈指数衰减，从而得到 $\mathbb{E}[N_{\text{adaptive}}] \approx N_{\min}$。对于 q 较小的困难实例，该算法需要更多样本，直至最大预算 $N_{\max}$。该界表明，自适应采样能够根据实例难度高效分配计算资源。完整证明见附录 C。

3.4. BOPO：最佳锚定目标引导偏好优化

3.4.1. 动机

传统的神经组合优化强化学习方法受到稀疏奖励的困扰：只有在完整生成解之后才能获得反馈，因此对中间令牌决策的指导有限。GRPO（Shao 等，2024年）通过在组内归一化奖励并计算相对优势来解决这一问题。然而，在 GRPO 中，梯度更新由奖励最高的样本主导，其他样本通过优势归一化所作的贡献很小。我们观察到，组合优化问题具有天然的偏好结构：对于任意两个可行解 y_i 和 y_j，当且仅当 $y_i \succ y_j$ 时，$f_p(y_i) < f_p(y_j)$（对于最小化问题）。这提供了稠密的目标引导监督信号，无需明确的人工标注。

3.4.2. 偏好对构造

对于每个问题实例 p，我们从当前策略 π_θ 中采样 K 解$\{y_1, \ldots, y_K\}$。令 $\mathcal{F}_p = \{y_i \mid y_i \in \mathcal{X}_{\mathcal{C}_p}\}$ 表示可行解的子集，并令 $y^* = \arg\min_{y \in \mathcal{F}_p} f_p(y)$ 表示样本中的最佳可行解。我们采用“最佳锚定”策略构造偏好对：

$$\mathcal{D}_p = \{(y^*, y_i) \mid y_i \in \mathcal{F}_p, y_i \neq y^*\}.$$

这种构造确保所有较差的可行解都能参与学习，这不同于 GRPO，在后者中梯度会被单个最佳样本主导。令 $K' = |\mathcal{F}_p|$ 表示可行样本的数量。

3.4.3. 目标引导损失函数

并非所有偏好对都同样有信息量。最优性差距为 50% 的解，应比差距仅为 1% 的解提供更强的学习信号。我们通过目标引导加权来体现这一点：

定义 3.14 (BOPO 损失)。 对于由实例 p 构造的偏好对，*BOPO* 损失 $\mathcal{L}_{BOPO}(\theta)$ 定义如下：

$$-\mathbb{E}_{p\sim\mathcal{D}}\left[\frac{1}{K'-1}\sum_{y_i\in\mathcal{F}_p\setminus\{y^*\}} w(\Delta_i)\cdot\log\sigma\left(\beta\cdot s_\theta(y^*, y_i)\right)\right]$$

其中，$\Delta_i = f_p(y_i) - f_p(y^*)$ 表示目标差距，$s_\theta(y_w, y_l) = \log \frac{\pi_\theta(y_w|\phi(p))}{\pi_\theta(y_l|\phi(p))}$ 表示对数概率比，$w(\Delta) = \Delta/\bar{\Delta}$ 表示目标引导缩放因子，其中 $\bar{\Delta} = \frac{1}{K'-1}\sum_{y_i\in\mathcal{F}_p\setminus\{y^*\}}\Delta_i > 0$ 为平均差距，$\sigma(\cdot)$ 表示 *Sigmoid* 函数，而 $\beta > 0$ 是控制偏好强度的温度超参数。

缩放因子 $w(\Delta)$ 确保质量差异较大的配对对梯度的贡献按比例更大。这在整个训练过程中提供了密集且信息丰富的监督信号，因为每个较差解都会根据其质量差距做出贡献。

3.4.4. 收敛性分析

我们在标准假设下为 BOPO 建立收敛保证：(i) L光滑的损失函数，(ii) 方差为 σ^2的随机梯度有界方差，以及 (iii) 有界缩放因子 $w(\Delta) \in [w_{\min}, w_{\max}]$，对某个 $0 < w_{\min} \leq w_{\max} < \infty$。完整陈述见附录 C.1。

定理 3.15（BOPO 收敛）。 在光滑性、有界方差和有界缩放假设下，当学习率为 $\eta = \frac{1}{L\sqrt{T}}$ 时，*BOPO* 满足：

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}_{BOPO}(\theta_t)\|^2] \leq \frac{C_1 + C_2}{\sqrt{T}} = O\left(\frac{1}{\sqrt{T}}\right),$$

其中 $C_1 = 2L(\mathcal{L}_{BOPO}(\theta_0) - \mathcal{L}^*)$ 取决于初始化，而 $C_2 = w_{\max}^2\sigma^2$ 刻画了由目标函数引导的缩放所带来的方差。

关键洞见在于，目标函数引导的缩放因子 $w(\Delta)$仅影响方差常数 C_2，而不影响$O(1/\sqrt{T})$的收敛速率，因为有界缩放假设确保 $w(\Delta)$保持受控。

推论 3.16 (迭代复杂度)。为找到满足$\min_{t\in\{0,\ldots,T-1\}}\mathbb{E}[\|\nabla\mathcal{L}_{BOPO}(\theta_t)\|^2] \leq \epsilon$ 的 ϵ-驻点，*BOPO* 需要 $T = O(1/\epsilon^2)$ 次迭代，与光滑非凸优化中标准随机梯度下降的速率相匹配。

表2。 与基线方法的比较。可行性：可行性（%）。差距：最优性差距（%）。 和 高亮显示。

方法	TSP		OP		CVRP		MIS		MVC		PFSP		JSSP		时间（秒）
	可行性（%）	差距（%）	可行性(%)	差距(%)	可行性(%)	差距(%)	可行性(%)	差距(%)	可行性(%)	差距(%)	可行性(%)	差距(%)	可行性(%)	差距(%)	
通用大语言模型（零样本）															
GPT-4o	39	33.79	59	55.19	15	76.62	8	11.70	6	16.67	88	20.57	7	97.85	5.3
GPT-4o-mini	28	53.38	80	70.70	3	68.10	0	0.00	3	29.52	78	21.47	8	212.00	4.5
Claude-3.5-Sonnet	66	24.53	49	34.62	30	38.34	13	12.51	2	6.25	100	18.42	10	90.00	5.4
Claude-3.5-Haiku	45	38.60	26	51.26	4	41.48	2	23.33	6	43.01	73	20.58	9	95.51	5.1
DeepSeek-V3	73	35.75	50	46.10	21	58.22	5	12.05	15	37.15	58	20.81	52	103.19	26.4
Llama-3.3-70B	50	69.08	27	48.98	31	97.31	8	37.12	20	22.86	98	21.97	29	105.01	2.1
Qwen2.5-72B	20	36.89	32	49.36	61	180.91	14	29.56	5	63.20	98	21.13	53	103.90	12.5
推理模型															
GPT-o3-mini	91	306.00	8	43.93	50	139.00	66	9.23	33	2.98	98	16.97	20	77.86	84.0
GPT-o1	54	276.00	31	40.90	24	154.00	82	8.03	47	3.58	89	14.86	29	81.90	192.0
DeepSeek-R1	48	70.99	60	40.54	26	30.46	41	1.60	38	4.17	**100**	16.65	5	26.29	390.0
基于大语言模型的优化															
OPRO	83	35.98	85	53.96	21	37.03	7	5.95	9	41.67	99	18.40	65	83.35	126.0
LMEA	77	265.00	48	66.18	24	61.24	5	25.00	13	34.22	98	14.31	44	83.19	318.0
PHP	84	33.84	43	36.08	33	58.11	5	11.67	13	19.84	92	17.23	56	104.04	96.0
SGE	98	29.66	93	24.49	92	3.62	94	3.83	94	3.83	95	4.48	87	38.58	216.0
神经组合优化求解器															
仅监督微调	89	2.30	54	2.32	59	6.02	80	1.71	98	2.41	**100**	2.22	**100**	11.01	5.6
GRPO	91	2.32	92	4.25	80	8.27	83	1.34	98	2.39	**100**	2.12	**100**	10.94	5.6
LLMCoSolver（N=1）	91	2.32	92	4.25	80	8.27	83	1.34	98	2.39	**100**	2.12	**100**	10.94	5.6
LLMCoSolver（N=8）	**100**	1.07	**100**	1.85	**100**	4.53	94	1.04	**100**	1.29	**100**	1.03	**100**	8.20	9.8
FALCON（我们的方法）															
FALCON（N=1）	**100**	1.89	**100**	2.14	**100**	5.37	**100**	1.18	**100**	1.85	**100**	1.76	**100**	9.23	5.8
FALCON（N=8）	**100**	0.95	**100**	1.42	**100**	3.68	**100**	0.87	**100**	1.12	**100**	0.89	**100**	7.15	10.4
FALCON（自适应）	**100**	**0.92**	**100**	**1.38**	**100**	**3.52**	**100**	**0.84**	**100**	**1.08**	**100**	**0.86**	**100**	**6.98**	6.3

3.5. 训练流程

完整的训练流程由两个阶段组成：

阶段 1：监督微调（SFT）。我们使用领域专用求解器（TSP/CVRP 使用 LKH，MIS/MVC 使用 Gurobi 等）生成的专家解答对基础大语言模型进行微调。这使模型学会解的格式和基本的问题求解模式。

阶段 2：BOPO 优化。从 SFT 检查点出发，我们为每个实例采样 K 个解，利用最佳锚定策略构造偏好对，并使用 BOPO 损失更新模型参数。这使策略逐步生成质量更高的解，同时通过修复层维持可行性。

4. 实验

我们进行了广泛的实验，在七类 NP难组合优化问题上，将 FALCON 与最先进的基线方法进行比较。实验旨在回答以下研究问题：

- **研究问题1**：FALCON 能否在保持具有竞争力的解质量的同时，实现 100% 的可行性?
- **研究问题2**：FALCON 与通用大语言模型和特定领域求解器相比表现如何?
- **研究问题3**：各个组件（BOPO、文法、修复、自适应采样）分别有何贡献?
- **研究问题4**：FALCON 在不同问题规模和实例分布下的扩展性如何?

4.1. 实验设置

问题 我们在三个领域的七类 NP难问题上进行评估： 路径规划(旅行商问题、定向越野问题、容量约束车辆路径问题)、图 (最大独立集、最小顶点覆盖)，以及 调度 (置换流水车间调度问题、作业车间调度问题)。每个问题都涉及不同类型的约束，例如哈密顿回路、容量限制、独立性和优先关系，从而实现全面覆盖。遵循Jiang 等人(2025年)，我们使用成熟的领域专用求解器生成合成实例（统计数据见表 9）。正式的问题定义见附录B。

基线方法 我们将 FALCON 与四类基线方法进行比较。(1)通用大语言模型在零样本设置下，包括专有模型（GPT-4o、GPT-4o-mini、Claude-3.5-Sonnet、Claude-3.5-Haiku）和开源模型（DeepSeek-V3（Liu 等，2024年a）、Llama-3.3-70B（Dubey 等，2024年）、Qwen2.5-72B（Yang 等，2025年））。(2) 推理增强型大语言模型 具备扩展推理能力：GPT-o3-mini、GPT-o1 和 DeepSeek-R1。(3)基于大语言模型的优化方法使用大语言模型进行迭代

表3.与领域启发式的比较。不同问题规模下的最优性差距（%）。小型：10–30个节点/作业；中型：40–60；大型：70–100。

方法	TSP			CVRP		
	小型	Med.	大型	小型	Med.	大型
最近邻	19.36	24.13	26.19	18.36	20.59	22.07
OR-Tools	0.82	2.59	3.59	3.60	7.87	8.84
ACO	1.98	17.98	36.69	2.52	17.07	29.49
LLMCoSolver（N=8）	0.14	0.70	1.34	1.70	4.57	7.24
FALCON（N=8）	0.12	0.61	1.15	1.48	3.89	6.28
FALCON（自适应）	**0.10**	**0.57**	**1.08**	**1.42**	**3.71**	**6.05**

方法	PFSP			JSSP		
	小型	Med.	大型	小型	Med.	大型
NEH	1.33	2.78	3.56	–	–	–
FIFO	–	–	–	24.38	32.97	39.00
LLMCoSolver（N=8）	0.45	0.89	1.56	4.23	7.89	12.34
FALCON（N=8）	0.39	0.74	1.28	3.71	6.92	10.85
FALCON（自适应）	**0.36**	**0.71**	**1.21**	**3.58**	**6.68**	**10.52**

表4. 旅行商问题和容量约束车辆路径问题上的消融研究。

配置	TSP			CVRP		
	Fea.	Gap	Time	Fea.	Gap	Time
FALCON（完整）	**100**	**0.92**	6.3s	**100**	**3.52**	6.8s
不含语法	**100**	0.95	6.5s	**100**	3.67	7.1s
无修复	94	0.89	6.1s	85	3.41	6.5s
无自适应	**100**	0.91	10.4秒	**100**	3.48	11.2秒
无 BOPO	**100**	1.85	6.3s	**100**	5.89	6.8s
监督微调 +修复	**100**	2.18	5.9s	**100**	5.72	6.2s
仅监督微调	89	2.30	5.6s	59	6.02	5.8s

式解细化：OPRO（Yang 等，2023年）、LMEA（Liu 等，2024年b）、PHP（Zheng 等，2023年）和 SGE（Iklassov 等，2024年）。(4) 神经组合优化求解器：仅监督微调（不使用强化学习的监督微调）、GRPO（组相对策略优化）和 LLMCoSolver（Jiang 等，2025年）（结合了 SFT 与 FOARL）。

指标 我们使用两项主要指标评估模型。可行率衡量满足全部问题约束的生成解所占百分比：$M_f(\mathcal{P}_s) = \frac{|\{p \in \mathcal{P}_s \mid x_p \in X_{C_p}\}|}{|\mathcal{P}_s|}$。最优性差距 衡量相对于领域特定求解器所得参考解的解的质量：$M_o(\mathcal{P}_s) = \frac{1}{|\mathcal{P}_s|} \sum_{p \in \mathcal{P}_s} \frac{f_p(\hat{x}_p) - f_p(x_p^*)}{|f_p(x_p^*)|}$。我们还报告每个实例的平均推理时间。对于FALCON，我们设置最小样本数 $N_{\min} = 8$、最大样本数 $N_{\max} = 64$、置信度阈值 $\tau = 0.85$以及采样温度 $T = 0.7$。

4.2. 主要结果（研究问题1与研究问题2）

所有基线方法和问题的全面比较见表2（正文中已提及）。

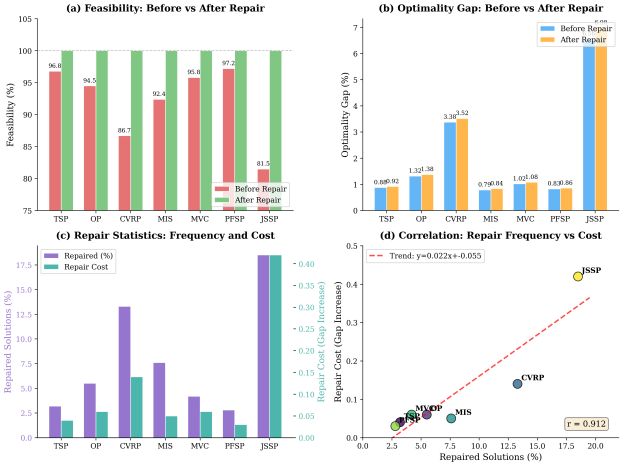


图1. 七个组合优化问题上的修复层统计。(a) 可行率。(b) 最优性差距。(c) 每个问题的修复频率与成本。(d) 修复频率与成本之间的强相关（$r = 0.912$）。

研究问题1：100% 可行性保证。FALCON 在全部七个问题上均实现了 100% 可行性，即使仅使用一次采样（$N = 1$），这验证了我们的理论保证（定理3.6）。相比之下，通用大语言模型的可行率差异很大（2%–100%），其中容量约束车辆路径问题和图问题尤其具有挑战性。LLMCoSolver 的可行率达到 80%–100%（$N = 1$），但即使使用 $N = 8$，它在最大独立集问题上也仅达到 94%——这表明单靠拒绝采样无法保证可行性。

研究问题2：具有竞争力的解质量。尽管施加了硬可行性约束，FALCON 仍能保持具有竞争力的最优性差距。采用自适应采样时，FALCON 在 7 个问题中的 5 个上取得了最小差距，且平均使用的样本比固定的 $N = 8$ 设置少 40%。质量退化极小，这证实了：(1) 经 BOPO 训练的模型在修复前就能生成很大比例的可行解；(2) 修复算子能够保持解局部性；(3) 在某些情况下，修复甚至可以通过移除低效重复项来提升解的质量。

4.2.1. 与领域启发式方法的比较

表3在不同问题规模上将 FALCON 与经典启发式方法进行了比较。FALCON 始终以大幅优势优于简单启发式方法（例如最近邻、SPT/先进先出），并相较更复杂的方法（例如 OR-Tools、2-opt、NEH）取得具有竞争力或更优的性能。值得注意的是，随着问题规模增大，FALCON 仍保持稳定性能，而蚁群优化等经典启发式方法在大型实例上显著退化（例如在大型旅行商问题上差距达 36.69%）。这表明具有强泛化能力的学习型求解器具有优势。

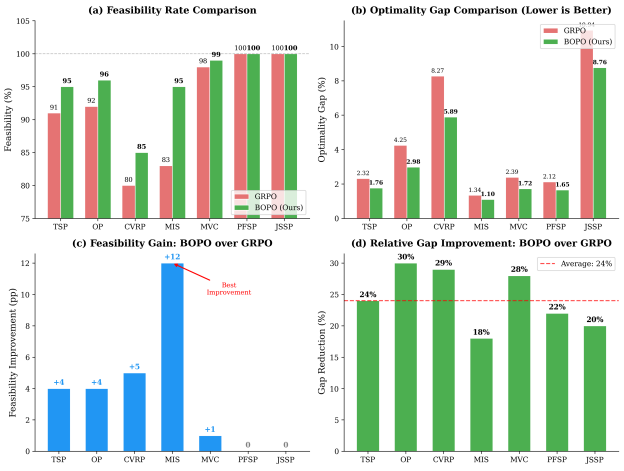


图2。从同一 SFT 检查点出发并采用相同训练预算时 BOPO 与 GRPO 的比较。(a) BOPO 在所有问题上的最优性差距均持续更低。(b,c) BOPO 同时改善了最优性差距和可行率。(d) 相对差距改善。

表 5。固定采样与自适应采样。

问题	固定（$N = 64$）			自适应		
	Gap	样本	Time	Gap	样本	Time
TSP	0.89	64	42.5秒	0.92	18.4	12.8秒
OP	1.35	64	45.2秒	1.38	22.6	16.4秒
CVRP	3.45	64	48.6秒	3.52	31.2	24.5秒
MIS	0.81	64	41.8秒	0.84	25.8	17.9秒
MVC	1.05	64	42.3秒	1.08	21.4	15.2秒
PFSP	0.84	64	46.7秒	0.86	19.2	14.1秒
JSSP	6.85	64	52.4秒	6.98	48.6	41.2秒
Avg.	2.18	64	45.6秒	2.23	26.7	20.3秒

4.3. 消融研究（研究问题3）

表 4展示了 FALCON 各组件的贡献。修复层对于实现 100% 可行性至关重要——移除后，可行性会降至与基线模型相当的水平。文法约束通过确保输出格式有效性来减轻修复负担。BOPO 在可行性与最优性差距两方面均优于 GRPO，而自适应采样在保持解的质量的同时降低了计算成本。

图 2详细比较了 BOPO 与 GRPO。BOPO 始终优于 GRPO，且在约束更多的问题上提升幅度更大（例如，容量约束车辆路径问题的差距降低了 29%；最大独立集的可行性提高了 12 个百分点）。

4.4. 分析

修复层统计。图1 显示，BOPO 训练的模型在修复前能生成很高比例的可行解（81%–97%）

表 6。按实例难度进行样本分配。

难度	TSP			CVRP		
	实例	样本	Gap	实例	样本	Gap
简单（Cons.>0.7）	42%	10.2	0.45	28%	11.8	1.89
中等（0.3–0.7）	38%	24.6	0.98	41%	32.4	3.67
困难（Cons.<0.3）	20%	52.3	1.82	31%	54.7	5.92

。因此，修复层很少被调用，产生的计算开销也极低（<0占总时间的 0.5%）。修复频率与问题约束的复杂度相关：较简单的排列问题（TSP、PFSP）仅有 2.8%–3.2% 的情况需要修复，而具有多种或复杂约束类型的问题（CVRP、JSSP）则更常需要修复（13.3%–18.5%）。重要的是，无论修复频率如何，最终的质量退化都保持在有限范围内，验证了定理 3.7中确立的局部性属性。

自适应采样效率。 表 5 显示，与固定策略相比，自适应采样平均减少了 58% 的样本用量，同时实现了相当的解质量（最优性差距增加 $N = 64$ % <3）。这使推理时间缩短了 55%。效率提升因问题而异：由于模型置信度较高，TSP 和 PFSP 的样本用量降幅最大（减少 71%）；而 JSSP 因其优先约束复杂，所需样本更多（平均 48.6 个）。这表明，自适应分配机制能够有效识别问题难度，而无需手动调参。

难度感知分配。 表 6显示，简单实例会提前终止（平均使用 ∼10 个样本），而困难实例则会使用更多样本（∼50）。解一致性指标能够有效区分实例难度：简单实例（一致性较高， >0.7）以最少的计算量实现了较低的最优性差距（0.45%）；而困难实例（一致性较低， <0.3）则受益于更充分的探索，与提前终止基线相比，其最优性差距降低了 35%。这验证了引理 3.10，证实解一致性是特定模型判断问题难度的可靠指标。

5. 结论

我们提出了 FALCON，这是一种基于大语言模型的组合优化框架，可提供可证明的 100% 可行性保证，解决了现有生成式方法的一项关键弱点。FALCON 将可行性保障拆分为语法（通过语法约束解码）和语义（通过问题专用修复算子）两个层面，从而确保满足硬约束，同时保持有竞争力的解质量。自适应采样机制根据实例难度高效分配计算资源，而 BOPO 训练则支持目标函数感知的偏好学习。针对七个 NP难

问题开展的大量实验表明，FALCON 能够可靠地实现完全可行性，同时达到或超越最先进的神经网络和基于大语言模型的求解器的性能，从而能够稳健地部署于现实场景中。

影响声明

本文介绍了一项旨在推动机器学习领域发展的工作。我们的工作可能带来许多社会影响，但我们认为其中没有任何影响需要在此特别说明。

参考文献

Abouelrous, A.、Bliek, L.、Wu, Y. 和 Zhang, Y. 面向随机多目标优化的端到端深度强化学习，用于带时间窗的容量约束车辆路径问题。*arXiv*预印本 *arXiv:2512.01518*，2025年。Azar, M. G.、Guo, Z. D.、Piot, B.、Munos, R.、Rowland, M.、Valko, M. 和 Calandriello, D. 理解从人类偏好中学习的一般理论范式。载于国际人工智能与统计学会议，第4447–4455页。PMLR，2024年。Bao, L. L. N.、Le, D. H. 和 Nguyen, D. A. 组合优化在物流中的应用。载于*2018*年第四届绿色技术与可持续发展国际会议（*GTSD*），第329–334页。IEEE，2018年。Bejarano, F. P.、Brunke, L. 和 Schoellig, A. 训练过程中的安全过滤：提升强化学习智能体的性能和样本效率。*IEEE*机器人与自动化快报，2024年。Bello, I.、Pham, H.、Le, Q. V.、Norouzi, M. 和 Bengio, S. 基于强化学习的神经组合优化。*arXiv*预印本 *arXiv:1611.09940*，2016年。Dubey, A.、Jauhri, A.、Pandey, A.、Kadian, A.、Al-Dahle, A.、Letman, A.、Mathur, A.、Schelten, A.、Yang, A.、Fan, A. 等。Llama 3 模型家族。*arXiv*预印本 *arXiv:2407.21783*，2024年。Ethayarajh, K.、Xu, W.、Muennighoff, N.、Jurafsky, D. 和 Kiela, D. KTO：基于前景理论的模型对齐优化，2024年。网址 *https://arxiv.org/abs/2402.01306*。Iklassov, Z.、Du, Y.、Akimov, F. 和 Takac, M. 面向组合问题的自引导探索。神经信息处理系统进展，37:130569–130601，2024年。Jain, S. 和 Bharti, K. K. 使用元启发式算法进行灾后紧急资源分配的组合优化模型。软计算，27(18):13595–13611，2023年。

Jiang, X.、Wu, Y.、Li, M.、Cao, Z. 和 Zhang, Y. 大型语言模型作为端到端组合优化求解器。*arXiv*预印本 *arXiv:2509.16865*，2025年。

Kool, W., VanHoof, H., and Welling, M. 注意力机制，学会解决路径规划问题！*arXiv* 预印本 *arXiv:1803.08475*，2018年。

Liu, A., Feng, B., Xue, B., Wang,B., Wu, B., Lu, C.,Zhao, C., Deng, C., Zhang, C., Ruan, C., 等。Deepseek-v3 技术报告。*arXiv* 预印本 *arXiv:2412.19437*，2024a。

Liu, H., Zhang, Q., Marcus, R., and Sabek, I. Sefrqo：一种自我进化的基于微调 RAG 的查询优化器。*ACM* 数据管理会刊, 3(6):1–27, 2025年。Liu, S., Chen, C., Qu, X., Tang, K., and Ong, Y.-S. 大语言模型作为进化优化器。载于*2024*年 *IEEE*进化计算大会（*CEC*）, pp. 1–8. IEEE, 2024b。Liu, Y., Zhou, C., Zhang, P., Zhang, S., Zhang, X., Li, Z., and Chen, H. 面向图学习与优化的决策导向图神经网络。载于*2023*年 *IEEE*国际数据挖掘会议（*ICDM*）, pp. 1151–1156. IEEE, 2023。Liu, Y., Zhou, C., Zhang, P., Li, Z., Zhang, S., Lin, X., and Wu, X. Cl4co：一种用于基于图的神经组合优化的课程训练框架。载于*2024*年 *IEEE*国际数据挖掘会议（*ICDM*）, pp. 779–784. IEEE, 2024c。Liu, Y., Zhou, C., Zhang, P., Pan, S., Li, Z., and Chen, H. 面向组合优化的决策导向图神经网络。*arXiv* 预印本 *arXiv:2406.03647*, 2024d。McGarvey, J., Grabowski, M. R., Custard, B., and Gabelein, S. 用于偏远且基础设施匮乏环境中应急响应物流的自愈数据库。物流, 9(1):23, 2025年。Meng, Y., Xia, M., and Chen, D. Simpo：采用无参考奖励的简易偏好优化。神经信息处理系统进展, 37:124198–124235, 2024年。Meurer, M., Kelliger, T., Gerhard, N., R¨uppel, A. K., Grimmert, A., and Bergs, T. ManuSafeNextGen：面向下一代发动机安全关键部件的基于模型的制造——第一部分：方法论。*Procedia CIRP*, 133:370–375, 2025年。

Nazari, M., Oroojlooy, A., Snyder, L., and Tak´ac, M. 用于求解车辆路径规划问题的强化学习。神经信息处理系统进展, 31, 2018年。Park, K., Wang, J., Berg-Kirkpatrick, T., Polikarpova, N., and D' Antoni. 语法对齐解码。神经信息处理系统进展, 37:24547–24568, 2024年。Qiu, X., Gan, Y., Hayes, C. F., Liang, Q., Meyerson, E., Hodjat, B., and Miikkulainen, R. 大规模进化策略：超越强化学习的大语言模型微调。*arXiv* 预印本 *arXiv:2509.24372*, 2025年。Rafailov, R., Shar ma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. 直接偏好优化：你的语言模型其实暗中充当着奖励模型。神经信息处理系统进展, 36: 53728–53741, 2023年。Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. DeepSeekMath：推动开放语言模型数学推理能力的极限。*arXiv* 预印本 *arXiv:2402.03300*, 2024年。Vinyals, O., Fortunato, M., and Jaitly, N. 指针网络。神经信息处理系统进展, 28, 2015年。Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*, 2025年。Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. 大语言模型作为优化器。载于第十二届国际学习表征会议, 2023年。Zhang, C., Wu, Y., Ma, Y., Song, W., Le, Z., Cao, Z., and Zhang, J. 制造业中组合优化问题求解学习研究综述。*IET*协作智能制造, 5(1):e12072, 2023年。Zheng, C., Liu, Z., Xie, E., Li, Z., and Li, Y. 渐进式提示增强大语言模型的推理能力。*arXiv* 预印本 *arXiv:2304.09797*, 2023年。Zhu, J., Wu, Y., Lin, Z., Zhang, Z., Yin, H., Cao, Z., Jayavelu, S., and Li, X. 通过大语言模型引导的实例生成与渐进式适应，弥合合成与真实路径规划问题之间的差距。*arXiv* 预印本 *arXiv:2511.10233*, 2025年。

附录

A. 相关工作

神经组合优化 神经组合优化方法从指针网络（Vinyals 等，2015）发展到基于 Transformer 的架构（Kool 等，2018；Abouelrous，2025年），以及强化学习方法（Bello 等，2016；Nazari 等，2018）。近期研究探索了将大语言模型（LLM）用作组合优化求解器，采用的技术包括提示优化（Yang 等，2023）、进化策略（Qiu 等，2025年）以及直接微调（Zhu 等，2025年）。然而，现有方法通常依赖通过奖励塑形或事后拒绝采样实现的软约束，缺乏正式的可行性保证。

受约束文本生成 受约束解码通过词法约束（Park 等，2024年）、语法合规性（Liu 等，2025年）和安全过滤（Bejarano 等，2024年）来限制大语言模型的输出。我们将基于语法的过滤应用于组合优化问题所要求的特定输出格式。我们的方法不仅确保语法有效性，还通过专门的修复层确保语义可行性——这构成了大语言模型用于组合优化时一种新颖的两阶段可行性强制机制。

偏好优化。直接偏好优化（DPO）（Rafailov 等，2023年）为大语言模型对齐提供了一种高效的替代方案，可取代基于人类反馈的强化学习（RLHF）。它将奖励函数重新参数化为策略的隐式函数，从而无需进行显式奖励建模。后续研究扩展了这一框架：IPO（Azar 等，2024年）通过一种通用的 Ψ偏好优化框架来解决过拟合问题；KTO（Ethayarajh 等）受前景理论启发，使模型能够从非配对二元反馈中学习；SimPO（Meng 等，2024年）则完全摆脱了对参考模型的依赖。对于数学推理，GRPO（Shao 等，2024年）引入了组相对优势估计，利用可验证的正确性信号，无需人工标注，展现了基于结果的偏好学习的有效性。

B. 问题定义

本文对评估的全部七个组合优化问题给出了正式的数学定义。每个定义均明确了问题输入、决策变量、约束和优化目标。

B.1. 旅行商问题（TSP）

定义 B.1 （TSP）。 给定一组带有坐标 $\{(x_i, y_i)\}_{i=0}^{n-1}$ 的 n 城市，以及欧几里得距离矩阵 $d_{ij} = \|(x_i, y_i) - (x_j, y_j)\|_2$，求一个排列 π： $\{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, n-1\}$，使巡回路径总长度最小：

$$\begin{aligned} \min_{\pi} \quad & \sum_{i=0}^{n-1} d_{\pi(i),\pi((i+1) \bmod n)} \\ \textit{s.t.} \quad & \pi \textit{ is a permutation of } \{0, 1, \ldots, n-1\}. \end{aligned}$$

约束要求构成一条哈密顿回路，使每个城市恰好访问一次，并最终返回起始城市。目标函数是使行进的欧几里得距离总和最小。

B.2. 定向越野问题（OP）

定义 B.2 （定向越野问题）。 给定 n 个带有坐标 $\{(x_i, y_i)\}_{i=0}^{n-1}$ 的地点、奖励 $\{p_i\}_{i=1}^{n-1}$ （其中 $p_i > 0$）、位于节点 0 的车场、欧几里得距离矩阵 $d_{ij} = \|(x_i, y_i) - (x_j, y_j)\|_2$以及距离预算 $D > 0$，求一个子集 $S \subseteq \{1, \ldots, n-1\}$及其访问顺序 π，使收集到的总奖励最大：

$$\begin{aligned} \max_{S,\pi} \quad & \sum_{i \in S} p_i \\ \textit{s.t.} \quad & \sum_{j=0}^{|S|} d_{\pi(j),\pi(j+1)} \le D \\ & \pi(0) = \pi(|S|+1) = 0 \end{aligned}$$

$$\pi : \{0, 1, \ldots, |S| + 1\} \to \{0\} \cup S \textit{ is a valid path.}$$

距离预算约束将总行驶距离限制为至多 D。路径必须从车场（节点 0）出发并返回车场。目标函数是最大化从访问地点收集的奖励之和。

B.3. 带容量约束的车辆路径问题（CVRP）

定义 B.3 （CVRP）**：** 给定 n 个客户，其坐标为 $\{(x_i, y_i)\}_{i=0}^{n-1}$，需求量为 $\{q_i\}_{i=1}^{n-1}$，其中 $q_i > 0$；节点 0 处设有一个车场，其 $q_0 = 0$；给定欧几里得距离矩阵 $d_{ij} = \|(x_i, y_i) - (x_j, y_j)\|_2$，以及车辆容量 $Q > 0$，求一组路径 $\{R_1, R_2, \ldots, R_K\}$，使总行驶距离最小：

$$\begin{aligned} \min_{\{R_k\}_{k=1}^{K}} \quad & \sum_{k=1}^{K} \textit{length}(R_k) \\ \textit{s.t.} \quad & \bigcup_{k=1}^{K} R_k = \{1, \ldots, n-1\} \\ & R_i \cap R_j = \emptyset \textit{ for all } i \neq j \\ & \sum_{i \in R_k} q_i \leq Q \quad \textit{for all } k \in \{1, \ldots, K\} \\ & \textit{Each route } R_k \textit{ starts and ends at node } 0. \end{aligned}$$

其中，路径 $R_k = (r_k^{(1)}, r_k^{(2)}, \ldots\ldots, r^{(|R_k|)}{}_k)$ 的长度为 R_k）$= d_{0,\ r_k^{(1)}} + \sum_{j=1}^{|R_k|-1} d_{r_k^{(j)},\ r_k^{(j+1)}} + d_{r_k^{(|R_k|)},\ 0}$，其中 $r_k^{(j)}$ 表示路径 k中访问的第 j个客户。

划分约束确保所有客户都恰好在所有路径中被服务一次。容量约束确保分配给每条路径的客户总需求不超过车辆容量 Q。每条路径都必须从车场出发并返回车场。

B.4. 最大独立集（MIS）

定义 B.4 （MIS）。给定一个无向图 $G = (V, E)$，其顶点集为 $V = \{v_1, v_2, \ldots, v_n\}$，边集为 $E \subseteq V \times V$，求一个子集 $S \subseteq V$，使其基数最大：

$$\begin{aligned} \max_{S \subseteq V} \quad & |S| \\ \textit{s.t.} \quad & \forall (u, v) \in E : \neg(u \in S \wedge v \in S). \end{aligned}$$

独立性约束要求 S 中任意两个顶点都不相邻；也就是说， E 中不存在一条边，其两个端点都在 S 中。目标函数是最大化独立集的大小。

B.5. 最小顶点覆盖（MVC）

定义 B.5 （MVC）。给定一个无向图 $G = (V, E)$，其顶点集为 $V = \{v_1, v_2, \ldots, v_n\}$，边集为 $E \subseteq V \times V$，求一个子集 $S \subseteq V$，使其基数最小：

$$\begin{aligned} \min_{S \subseteq V} \quad & |S| \\ \textit{s.t.} \quad & \forall (u, v) \in E : u \in S \vee v \in S. \end{aligned}$$

覆盖约束要求 E 中的每条边至少有一个端点位于 S中，从而确保所有边都被“覆盖”。目标函数是使顶点覆盖的规模最小。

注 B.6 (最大独立集—最小顶点覆盖对偶性**)。** 在任意图 $G = (V, E)$中，最大独立集问题与最小顶点覆盖问题互为补问题。子集 $S \subseteq V$是独立集，当且仅当其补集 $V \setminus S$是顶点覆盖。这种对偶性意味着，解决其中一个问题即可立即得到另一个问题的解。

B.6. 置换流水车间调度（PFSP）

定义B.7 (置换流水车间调度问题)。 给定 n 个作业和 m 台机器，以及加工时间 $p_{i,\ j} > 0$ （作业 $i \in \{1, \ldots, n\}$ 在机器 $j \in \{1, \ldots, m\}$上的加工时间），求一个排列 π： $\{1, \ldots, n\} \to \{1, \ldots, n\}$，使最大完工时间（最后一个作业在最后一台机器上的完工时间）最小:

$$
\begin{aligned}
\min_{\pi} \quad & C_{\pi(n),m} \\
\textit{s.t.} \quad & C_{\pi(k),j} = \max(C_{\pi(k),j-1}, C_{\pi(k-1),j}) + p_{\pi(k),j} \quad \textit{for all } k \in \{1, \ldots, n\}, j \in \{1, \ldots, m\} \\
& C_{\pi(k),0} = 0 \quad \textit{for all } k \in \{1, \ldots, n\} \\
& C_{\pi(0),j} = 0 \quad \textit{for all } j \in \{1, \ldots, m\} \\
& \pi \textit{ is a permutation of } \{1, \ldots, n\},
\end{aligned}
$$

其中$C_{i,\ j}$ 表示作业 i 在机器 j上的完工时间。

流水车间约束要求所有作业在所有机器上按相同顺序 π 进行加工。完工时间递推确保作业 π（k）在机器 j 上仅能在同一作业已于前一台机器$j-1$ 上完工、且前一作业 π（$k-1$）已在当前机器 j上完工之后开始。边界条件将任何加工开始之前的完工时间设为零。

B.7. 作业车间调度问题（JSSP）

定义 B.8 （JSSP）。 给定 n 个作业和 m 台机器，其中每个作业 $i \in \{1, \ldots, n\}$ 由 m 道工序组成的序列（$o_{i,\ 1}$， $o_{i,\ 2}$，.....，$o_{i,\ m}$）构成，每道工序$o_{i,\ k}$需要特定机器 μ（$o_{i,\ k}$） $\in \{1, \ldots, m\}$，加工时间为p（$o_{i,\ k}$） > 0，求使最大完工时间最小的可行调度:

$$
\begin{aligned}
\min \quad & \max_{i \in \{1,\ldots,n\}} C_i \\
\textit{s.t.} \quad & S_{o_{i,k+1}} \geq C_{o_{i,k}} \quad \textit{for all } i \in \{1, \ldots, n\}, k \in \{1, \ldots, m-1\} \\
& C_{o_{i,k}} = S_{o_{i,k}} + p(o_{i,k}) \quad \textit{for all } i, k \\
& \textit{For any two distinct operations } o, o' \textit{ with } \mu(o) = \mu(o') : \\
& \quad S_o \geq C_{o'} \ \textit{ or } \ S_{o'} \geq C_o,
\end{aligned}
$$

其中， S_o 和 C_o 分别表示工序 o的开始时间和完工时间，$C_i = C_{o_{i,\ m}}$ 表示作业 i的完工时间。

优先约束确保每个作业内的工序按照指定顺序处理；每道工序只能在其所属作业中的前一道工序完成后开始。机器资源约束（析取约束）确保每台机器在任意时刻至多处理一道工序（不允许抢占）。目标函数是最小化所有作业的完工时间，即最小化各作业完工时间中的最大值。

C. 完整证明

假设 C.1（光滑性）。 损失函数 $\mathcal{L}_{BOPO}$（θ）是 L-光滑的:

$$\|\nabla \mathcal{L}_{BOPO}(\theta_1) - \nabla \mathcal{L}_{BOPO}(\theta_2)\| \leq L \|\theta_1 - \theta_2\|.$$

假设 C.2 （有界方差）。 随机梯度具有有界方差:

$$\mathbb{E}[\|\nabla \mathcal{L}_{BOPO}(\theta; \mathcal{B}) - \nabla \mathcal{L}_{BOPO}(\theta)\|^2] \leq \sigma^2,$$

其中， $\mathcal{B}$ 是随机抽取的小批量。

假设 C.3 (有界目标引导缩放)。 *BOPO* 中的目标引导缩放因子满足 $w(\Delta) \in [w_{\min}, w_{\max}]$，其中某个 $0 < w_{\min} \leq w_{\max} < \infty$。

具体而言，对于 $w(\Delta) = \Delta / \bar{\Delta}$，其中 $\bar{\Delta} = \frac{1}{K'-1} \sum_{y_i \in \mathcal{F}_p \setminus \{y^*\}} \Delta_i$，我们要求:

- $\Delta_i = f_p$（y_i） $- f_p$（y^*）$\geq \delta_{\text{最小值}} > 0$ 对于所有劣解均成立（由组合优化问题中的离散目标值保证）。

• 比值最大值$_i\,\Delta_i$/最小值$_j\,\Delta_j \le C$ 由常数 C界定。

在这些条件下， w（Δ） $\in [1/C, C]$, 从而确保缩放有界。

C.1. 定理3.15 (BOPO 收敛)

证明。 为简洁起见，我们记 $\mathcal{L} := \mathcal{L}_{\text{BOPO}}$。证明分四步进行。

步骤 1：下降引理。 根据假设 C.1 （L-光滑性），对于任意 θ_1， $\theta_2 \in \mathbb{R}^d$:

$$\mathcal{L}(\theta_2) \le \mathcal{L}(\theta_1) + \langle \nabla\mathcal{L}(\theta_1), \theta_2 - \theta_1\rangle + \frac{L}{2}\|\theta_2 - \theta_1\|^2.$$

将其应用于 SGD 更新 $\theta_{t+1} = \theta_t - \eta g_t$，其中 g_t 是第 t 次迭代时的随机梯度，可得:

$$\begin{aligned}\mathcal{L}(\theta_{t+1}) &\le \mathcal{L}(\theta_t) + \langle \nabla\mathcal{L}(\theta_t), -\eta g_t\rangle + \frac{L}{2}\| - \eta g_t\|^2 \\ &= \mathcal{L}(\theta_t) - \eta\langle \nabla\mathcal{L}(\theta_t), g_t\rangle + \frac{L\eta^2}{2}\|g_t\|^2.\end{aligned}$$

步骤 2：取期望。在给定 θ_t 的条件下取期望，并利用随机梯度的无偏性$\mathbb{E}[g_t|\theta_t] = \nabla\mathcal{L}(\theta_t)$，得到:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta_{t+1})|\theta_t] &\le \mathcal{L}(\theta_t) - \eta\langle \nabla\mathcal{L}(\theta_t), \mathbb{E}[g_t|\theta_t]\rangle + \frac{L\eta^2}{2}\mathbb{E}[\|g_t\|^2|\theta_t] \\ &= \mathcal{L}(\theta_t) - \eta\|\nabla\mathcal{L}(\theta_t)\|^2 + \frac{L\eta^2}{2}\mathbb{E}[\|g_t\|^2|\theta_t].\end{aligned}$$

步骤 3：界定二阶矩。我们将随机梯度的二阶矩分解为:

$$\begin{aligned}\mathbb{E}[\|g_t\|^2|\theta_t] &= \mathbb{E}[\|g_t - \nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\theta_t)\|^2|\theta_t] \\ &= \mathbb{E}[\|g_t - \nabla\mathcal{L}(\theta_t)\|^2|\theta_t] + \|\nabla\mathcal{L}(\theta_t)\|^2 \\ &\quad + 2\mathbb{E}[\langle g_t - \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\theta_t)\rangle|\theta_t].\end{aligned}$$

由于 $\mathbb{E}[g_t - \nabla\mathcal{L}(\theta_t)|\theta_t] = 0$，交叉项为零:

$$\mathbb{E}[\|g_t\|^2|\theta_t] = \mathbb{E}[\|g_t - \nabla\mathcal{L}(\theta_t)\|^2|\theta_t] + \|\nabla\mathcal{L}(\theta_t)\|^2.$$

根据假设 C.2和 C.3，方差有界:

$$\mathbb{E}[\|g_t - \nabla\mathcal{L}(\theta_t)\|^2|\theta_t] \le w_{\max}^2\sigma^2.$$

因子 $w_{\max}^2$ 源于 BOPO 梯度中包含缩放项 $w(\Delta_i)$，根据假设 $w_{\max}$，该项有界于C.3。因此:

$$\mathbb{E}[\|g_t\|^2|\theta_t] \le \|\nabla\mathcal{L}(\theta_t)\|^2 + w_{\max}^2\sigma^2.$$

步骤 4：望远镜求和与最终界。将此界代回期望中:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta_{t+1})|\theta_t] &\le \mathcal{L}(\theta_t) - \eta\|\nabla\mathcal{L}(\theta_t)\|^2 + \frac{L\eta^2}{2}\left(\|\nabla\mathcal{L}(\theta_t)\|^2 + w_{\max}^2\sigma^2\right) \\ &= \mathcal{L}(\theta_t) - \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla\mathcal{L}(\theta_t)\|^2 + \frac{L\eta^2 w_{\max}^2\sigma^2}{2}.\end{aligned}$$

取完全期望并重新排列各项:

$$\eta\left(1 - \frac{L\eta}{2}\right)\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \le \mathbb{E}[\mathcal{L}(\theta_t)] - \mathbb{E}[\mathcal{L}(\theta_{t+1})] + \frac{L\eta^2 w_{\max}^2\sigma^2}{2}.$$

从 $t=0$ 求和到 $T-1$:

$$\eta\left(1-\frac{L\eta}{2}\right)\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]\le\mathcal{L}(\theta_0)-\mathbb{E}[\mathcal{L}(\theta_T)]+\frac{L\eta^2w_{\max}^2\sigma^2T}{2}.$$

由于 $\mathcal{L}(\theta_T)\ge\mathcal{L}^*$（可能的最小损失）:

$$\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]\le\frac{\mathcal{L}(\theta_0)-\mathcal{L}^*}{\eta(1-\frac{L\eta}{2})}+\frac{L\eta w_{\max}^2\sigma^2T}{2(1-\frac{L\eta}{2})}.$$

设学习率为 $\eta=\frac{1}{L\sqrt{T}}$，则对于 $T\ge1$，有 $\frac{L\eta}{2}=\frac{1}{2\sqrt{T}}\le\frac{1}{2}$，这意味着 $1-\frac{L\eta}{2}\ge\frac{1}{2}$。因此:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]&\le\frac{\mathcal{L}(\theta_0)-\mathcal{L}^*}{T\cdot\frac{1}{L\sqrt{T}}\cdot\frac{1}{2}}+\frac{L\cdot\frac{1}{L\sqrt{T}}\cdot w_{\max}^2\sigma^2}{2\cdot\frac{1}{2}}\\&=\frac{2L(\mathcal{L}(\theta_0)-\mathcal{L}^*)}{\sqrt{T}}+\frac{w_{\max}^2\sigma^2}{\sqrt{T}}.\end{aligned}$$

证明至此完成。 □

C.2. 推论3.16（迭代复杂度）

证明。根据定理3.15，我们有:

$$\min_{t\in\{0,\dots,T-1\}}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]\le\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]\le\frac{C_1}{\sqrt{T}},$$

其中 $C_1=2L(\mathcal{L}(\theta_0)-\mathcal{L}^*)+w_{\max}^2\sigma^2$。
为了实现 $\min_t\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2]\le\epsilon$，我们需要:

$$\frac{C_1}{\sqrt{T}}\le\epsilon\quad\Longrightarrow\quad T\ge\frac{C_1^2}{\epsilon^2}.$$

因此，迭代复杂度为 $O(1/\epsilon^2)$。 □

C.3. 定理3.3（格式有效性保证）

证明。 我们对解码步骤的数量进行强归纳法，以证明该定理。

设置。设 $G=(V,\Sigma,R,S)$ 为上下文无关文法，设 $\mathcal{A}=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ 为识别 $\mathcal{L}(G)$ 的对应下推自动机（PDA）。在每个步骤 t 中，算法 1 维护下推自动机状态（q_t,γ_t），其中 $q_t\in Q$ 是当前状态，$\gamma_t\in\Gamma^*$ 是栈内容。

不变式。 我们证明以下不变式：经过 t 个解码步骤并生成部分输出 $y^{(t)}=v_1v_2\cdots v_t$ 后，在 G 中存在一个推导，使得 $y^{(t)}$ 是 $\mathcal{L}(G)$ 中某个字符串的前缀，并且下推自动机状态 $(q_t,\ \gamma_t)$ 可通过读取 $y^{(t)}$ 从初始配置 $(q_0,\ Z_0)$ 到达。

基本情况（$t=0$）。 初始时，$y^{(0)}=\epsilon$（空串），且下推自动机处于状态 $(q_0,\ Z_0)$。空串显然是 $\mathcal{L}(G)$ 中任意字符串的前缀，而 $(q_0,\ Z_0)$ 是初始配置。
归纳步骤。 假设经过第 t 步后不变式成立。在第 $t+1$ 步:

1. 算法1计算有效下一词元集合:

$$\mathcal{V}_{\text{valid}}=\{v\in\Sigma\mid\delta((q_t,\gamma_t),v)\text{ is defined}\}.$$

2. 算法会屏蔽所有词元 $v \notin \mathcal{V}_{\text{valid}}$，方法是将它们的逻辑值设为 $-\infty$。

3. 从剩余的有效词元中采样一个词元 v_{t+1}。

4. 根据构造，$v_{t+1} \in \mathcal{V}_{\text{valid}}$，因此 $\delta((q_t,\ \gamma_t)\ v_{t+1})$ 有定义。

5. 下推自动机转移到 $(q_{t+1},\ \gamma_{t+1}) = \delta((q_t,\ \gamma_t)\ v_{t+1})$。

由于 δ 仅对对应于 G 中有效推导步骤的转移有定义，因此新的部分输出 $y^{(t+1)} = y^{(t)}v_{t+1}$ 仍然是 $\mathcal{L}(G)$ 中某个字符串的前缀。

终止。 算法1 当下推自动机到达接受状态时终止 $q_T \in F$ 且栈为空。根据上下文无关文法与下推自动机之间的标准对应关系，当且仅当生成的字符串 $y = y^{(T)}$ 属于 $\mathcal{L}(G)$。

因此，输出 y 满足 $y \in \mathcal{L}(G)$。 □

C.4. 定理的证明3.6（100% 可行性保证）

证明。设 y_1、y_2、……、y_Nb e由大语言模型策略 π_θ生成的 N 样本。在应用修复算子后，我们obtai n:

$$\tilde{y}_i = \mathcal{R}(y_i), \quad i = 1, \ldots, N.$$

根据性质(1)，对于定义3.5 （可行性），每个修复后的解满足：

$$\tilde{y}_i \in X_{C_p}.$$

因此，所有 N解 $\{\tilde{y}_1, \ldots, \tilde{y}_N\}$ 均为可行。特别地：

$$P(\exists i : \tilde{y}_i \in X_{C_p}) = P(\forall i : \tilde{y}_i \in X_{C_p}) = 1.$$

□

C.5. 定理3.7（修复质量界）

证明。 假设目标函数 f_p 关于距离度量 d 是 L_f-利普希茨连续的；即，对于任意 $x,\ y \in X_p$：

$$|f_p(x) - f_p(y)| \leq L_f \cdot d(x, y).$$

将此不等式应用于 $x = \mathcal{R}(\hat{}x_p)$ 和 $y = \hat{}x_p$，得到：

$$|f_p(\mathcal{R}(\hat{x}_p)) - f_p(\hat{x}_p)| \leq L_f \cdot d(\mathcal{R}(\hat{x}_p), \hat{x}_p).$$

根据修复算子的局部性假设 $\mathcal{R}$ （定义 3.5中的性质 (2)）：

$$d(\mathcal{R}(\hat{x}_p), \hat{x}_p) \leq \alpha \cdot v(\hat{x}_p),$$

其中 $v(\hat{}x_p)$ 衡量原始解的约束违反程度。

结合这些不等式可得：

$$|f_p(\mathcal{R}(\hat{x}_p)) - f_p(\hat{x}_p)| \leq L_f \cdot \alpha \cdot v(\hat{x}_p).$$

这意味着单侧界：

$$f_p(\mathcal{R}(\hat{x}_p)) \leq f_p(\hat{x}_p) + L_f \cdot \alpha \cdot v(\hat{x}_p).$$

请注意，修复操作也可能改善目标函数值，此时 $f_p(\mathcal{R}(\hat{}x_p)) < f_p(\hat{}x_p)$。该界给出了潜在退化的最坏情况保证。

□

C.6. 推论3.8 的证明 (与拒绝采样的比较)

证明。令 $Y_i \in \{0,1\}$ 表示样本 y_i 可行的指示变量。根据假设，$P(Y_i = 1) = p_f$，且这些样本相互独立。

N 个样本中没有任何一个可行的概率为：

$$P(\forall i : Y_i = 0) = \prod_{i=1}^{N} P(Y_i = 0) = (1 - p_f)^N.$$

因此，至少获得一个可行解的概率为：

$$P(\exists i : Y_i = 1) = 1 - (1 - p_f)^N.$$

为了确保该概率至少为 $1 - \delta$，我们要求：

$$\begin{aligned} 1 - (1 - p_f)^N &\geq 1 - \delta \\ (1 - p_f)^N &\leq \delta \\ N \log(1 - p_f) &\leq \log \delta \\ N &\geq \frac{\log(1/\delta)}{\log(1/(1 - p_f))}. \end{aligned}$$

对于较小的 p_f，使用近似 $\log(1 - p_f) \approx -p_f$，可得简化后的界：

$$N \gtrsim \frac{\log(1/\delta)}{p_f}.$$

□

C.7. 定理证明3.11 (采样带来的质量提升)

证明。令 $G_i = f_p(y_i) - f_p(x^*)$ 表示样本 y_i 的最优性差距。那么 $g_N = \min_i G_i$ 是最佳样本的差距。我们有：

$$\begin{aligned} P(g_N > \epsilon) = P(\min_i G_i > \epsilon) &= \prod_{i=1}^{N} P(G_i > \epsilon) \\ &= (1 - F(\epsilon))^N, \end{aligned}$$

其中，$F(\cdot)$ 是单个样本的差距的累积分布函数。

根据非负随机变量期望的标准尾积分公式：

$$\mathbb{E}[g_N] = \int_0^\infty P(g_N > \epsilon)\, d\epsilon = \int_0^\infty (1 - F(\epsilon))^N\, d\epsilon.$$

由此得到所需结果。 □

C.8. 推论证明3.12(指数分布)

证明。对于速率为 λ 的指数分布，尾概率为：

$$1 - F(\epsilon) = e^{-\lambda\epsilon}.$$

将其代入定理3.11可得：

$$\mathbb{E}[g_N] = \int_0^\infty \left(e^{-\lambda\epsilon}\right)^N\, d\epsilon$$

$$
\begin{aligned}
&= \int_0^\infty e^{-N\lambda\epsilon}\, d\epsilon \\
&= \left[-\frac{1}{N\lambda} e^{-N\lambda\epsilon}\right]_0^\infty \\
&= \frac{1}{N\lambda}.
\end{aligned}
$$

由于 $\mathbb{E}[g_1] = \frac{1}{\lambda}$对于指数分布成立，因此可得：

$$
\mathbb{E}[g_N] = \frac{\mathbb{E}[g_1]}{N}.
$$

□

C.9. 定理3.13 的证明（自适应采样复杂度）

证明。 令 T 表示 算法 2的随机停止时间。我们将期望样本数分解为 ：

$$
\begin{aligned}
\mathbb{E}[T] &= \sum_{n=1}^{N_{\max}} \Pr(T \geq n) \\
&= N_{\min} + \sum_{n=N_{\min}+1}^{N_{\max}} \Pr(T \geq n),
\end{aligned}
$$

其中，等式成立是因为算法在考虑提前终止之前总会至少收集 $N_{\min}$ 个样本。

收集 n 个样本后，令 n_{y^*} 表示与当前最佳解 y^*匹配的样本数。采用均匀先验（$\alpha_0 = \beta_0 = 1$）的贝塔-二项模型计算的贝叶斯置信度估计为：

$$
\mathrm{Conf}(y^*; n) = \frac{\alpha_0 + n_{y^*}}{\alpha_0 + \beta_0 + n} = \frac{1 + n_{y^*}}{2 + n}.
$$

若该置信度仍低于阈值 τ，算法会在步骤 n 继续采样；当 $n_{y^*} < \tau(2 + n) - 1$时，便会出现这种情况。

考虑一个问题实例，其中模型以概率 q采样到一个近优解（其最优性差距至多为 δ）。令 S_n 表示在前 n个样本中至少已采样到一个此类近优解的事件。补事件的概率为 $\Pr(S_n) = (1 - q)^n$。当置信度估计机制校准良好且已采样到近优解时，算法最终会通过对同一高质量解的重复采样识别出这一点，从而导致提前终止。

为得到一个简化但有信息量的上界，我们假设算法一旦以足够频率采样到近优解并达到置信度阈值便会停止。在此假设下，继续超过步骤 n的概率受限于尚未出现近优解的概率，从而对 $n \geq N_{\min}$给出 $\Pr(T \geq n) \leq (1 - q)^n$ 。

将此界应用于期望分解：

$$
\begin{aligned}
\mathbb{E}[T] &\leq N_{\min} + \sum_{n=N_{\min}+1}^{N_{\max}} (1 - q)^n \\
&= N_{\min} + (1 - q)^{N_{\min}+1} \cdot \frac{1 - (1 - q)^{N_{\max} - N_{\min}}}{1 - (1 - q)} \\
&= N_{\min} + \frac{(1 - q)^{N_{\min}+1}}{q} \cdot \left(1 - (1 - q)^{N_{\max} - N_{\min}}\right).
\end{aligned}
$$

由于 $(1 - q)^{N_{\max} - N_{\min}} \geq 0$，我们有：

$$
\mathbb{E}[T] \leq N_{\min} + \frac{(1 - q)^{N_{\min}} \cdot (1 - q)}{q}.
$$

采用一个较宽松但更易解释的界（通过在感兴趣的范围内将几何级数的和近似为线性函数），我们得到：

$$\mathbb{E}[N_{\text{adaptive}}] \leq N_{\min} + \frac{(N_{\max} - N_{\min})(1-q)^{N_{\min}}}{q}.$$

该界揭示了自适应行为。对于简单实例（其特征是 q 较大，近似最优解经常被采样到），指数衰减项 $(1-q)^{N_{\min}}$ 会变得非常小，从而得到 $\mathbb{E}[N_{\text{adaptive}}] \approx N_{\min}$。算法在收集到最低所需样本数后迅速终止。对于 q 较小的困难实例，该界会增大，算法将继续采样，直至达到最大预算 $N_{\max}$，以确保充分探索。这表明，自适应采样无需手动调参或预先了解问题特征，就能根据实例特定难度自动分配计算资源。

□

C.10. 引理证明3.10（一致性-难度关系）

证明。 我们对期望一致性及其与 2 阶 R´ényi 熵之间的关系给出严格推导。

对 K 个独立采样的解 $\{y_1, \ldots, y_K\}$，其一致性度量定义为 s:

$$\text{Cons} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i} \mathbb{I}[y_i = y_j].$$

取期望 独立样本上的tation，每个样本均抽取自分布 p（于解空间上） $\mathcal{Y}$:

$$\begin{aligned}
\mathbb{E}[\text{Cons}] &= \mathbb{E}\left[\frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i} \mathbb{I}[y_i = y_j]\right] \\
&= \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i} \mathbb{E}[\mathbb{I}[y_i = y_j]] \\
&= \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i} \Pr(y_i = y_j).
\end{aligned}$$

由样本的独立性，对于任意 $i \neq j$:

$$\begin{aligned}
\Pr(y_i = y_j) &= \sum_{y \in \mathcal{Y}} \Pr(y_i = y) \cdot \Pr(y_j = y) \\
&= \sum_{y \in \mathcal{Y}} p(y) \cdot p(y) = \sum_{y \in \mathcal{Y}} p(y)^2.
\end{aligned}$$

由于共有 $K(K-1)$个满足 $i \neq j$的有序对(i, j)，我们得到：

$$\mathbb{E}[\text{Cons}] = \frac{1}{K(K-1)} \cdot K(K-1) \cdot \sum_{y \in \mathcal{Y}} p(y)^2 = \sum_{y \in \mathcal{Y}} p(y)^2.$$

二阶R´enyi熵定义为 $H_2(p) = -\log \sum_y p(y)^2$。因此：

$$\mathbb{E}[\text{Cons}] = \sum_y p(y)^2 = e^{-H_2(p)}.$$

这建立了期望一致性与R´enyi熵之间的反指数关系。

当模型的输出分布高度集中于单个解（即大部分概率质量集中在某个特定的 y^* 上）时，对于 $y \neq y^*$，我们有 $p(y^*) \approx 1$ 且 $p(y) \approx 0$。这会产生 $\sum_y p(y)^2 \approx 1$，对应于低熵

$H_2(p) \approx 0$ 且具有较高的一致性 $\mathbb{E}[\text{Cons}] \approx 1$。这种集中表明模型对特定解具有较高的置信度，这通常出现在最优解明确且训练策略易于识别的实例中。

相反，当模型的分布分散在许多解上时，有效支撑集中的每个解 y 都会$\mathcal{S}$ 获得中等概率 $p(y) \approx 1/|\mathcal{S}|$。这会导致 $\sum_y p(y)^2 \approx 1/|\mathcal{S}|$，对应于高熵和低一致性。这种分散反映了模型对于哪个解是最优解的不确定性，通常出现在困难实例中：多个解看起来几乎等价，或问题结构复杂且模糊。

对于一个训练良好的模型，当较高的预测概率与实际解的质量相关时，早期样本之间的高一致性可以作为实例简单且不需要大量额外采样的可靠指标。低一致性则表明实例较为困难，需要更多探索以识别高质量的解。

□

D. 文法规范

本节为七个组合优化问题提供完整的上下文无关文法（CFG）规范。这些文法定义了有效解表示的句法结构，大语言模型必须生成这些表示。如第 3.1 节所述，每个文法都通过语法约束解码确保格式有效性。

D.1. 文法形式体系

我们采用标准的巴科斯-诺尔范式（BNF）表示上下文无关文法。文法 $G = (V、\ \Sigma、\ R、\ S)$ 由一组非终结符 V、一组终结符 Σ（对应于大语言模型词汇表中的词元）、一组产生式规则 R以及一个开始符号 $S \in V$组成。产生式规则表示为 $A \to \alpha$，其中 $A \in V$和 $\alpha \in (V \cup \Sigma)^*$。记号 X^+ 表示符号 X的一次或多次出现，而 ϵ表示空串。

对于每个问题，我们定义一个输入相关文法 G_p，通过根据问题实例 p将节点和作业索引限制在有效范围内，使其专门化于基础文法。具体而言，对于包含 n 个节点或作业的实例，索引的产生式规则将被专门化为仅生成 $\{0, 1, \ldots, n-1\}$ 或 $\{1, \ldots, n\}$中的值，具体取决于问题。

D.2. 通用定义

以下非终结符在所有问题特定语法中通用。数字 非终结符表示浮点目标值，而数位 表示单个十进制数位。

$$
\begin{aligned}
\textit{Number} &\to \textit{Digit}^+ \texttt{"."} \textit{Digit}^+ \\
\textit{Digit} &\to \texttt{"0"} \mid \texttt{"1"} \mid \texttt{"2"} \mid \texttt{"3"} \mid \texttt{"4"} \mid \texttt{"5"} \mid \texttt{"6"} \mid \texttt{"7"} \mid \texttt{"8"} \mid \texttt{"9"}
\end{aligned}
$$

D.3. TSP文法

TSP文法规定了单条路径的格式，该路径表示为节点索引的有序列表，后接计算得到的巡回路径长度。该文法确保括号匹配和逗号分隔正确，但不强制执行所有节点必须恰好出现一次这一语义约束；这一约束由修复算子处理。

$$
\begin{aligned}
S &\to \texttt{"Route: ["}\ \textit{NodeList}\ \texttt{"], Objective: "}\ \textit{Number} \\
\textit{NodeList} &\to \textit{Node} \mid \textit{Node}\ \texttt{", "}\ \textit{NodeList} \\
\textit{Node} &\to \textit{Digit}^+
\end{aligned}
$$

对于包含 n 个节点且其索引为 $\{0, 1, \ldots, n-1\}$的实例，输入相关文法通过将产生式规则替换为显式的备选项，限制 节点 仅生成有效索引：节点 $\to$"0" |"1" $|\cdots|$"$n-1$"。

D.4. CVRP 文法

CVRP 文法规定了一组路径，其中每条路径都是由客户节点组成的括号列表。文法允许多个路径以逗号分隔，以体现 CVRP 解的多路径结构。

$$
S \to \texttt{"Routes: ["}\ \textit{RouteList}\ \texttt{"], Objective: "}\ \textit{Number}
$$

$$
\begin{aligned}
\mathit{RouteList} &\to \mathit{Route} \mid \mathit{Route}\ \texttt{", "}\ \mathit{RouteList} \\
\mathit{Route} &\to \texttt{"["}\ \mathit{NodeList}\ \texttt{"]"} \\
\mathit{NodeList} &\to \mathit{Node} \mid \mathit{Node}\ \texttt{", "}\ \mathit{NodeList} \\
\mathit{Node} &\to \mathit{Digit}^{+}
\end{aligned}
$$

文法允许任意的路线结构，而修复算子负责强制满足语义约束：所有客户都恰好访问一次，且每条路径都满足车辆容量约束。

D.5. 定向越野问题语法

定向越野问题文法允许路径为空，这反映出某些实例可能在距离预算内没有可行解，或者最优解可能除了车场之外不访问任何客户。

$$
\begin{aligned}
S &\to \texttt{"Route: ["}\ \mathit{NodeListOpt}\ \texttt{"], Objective: "}\ \mathit{Number} \\
\mathit{NodeListOpt} &\to \epsilon \mid \mathit{NodeList} \\
\mathit{NodeList} &\to \mathit{Node} \mid \mathit{Node}\ \texttt{", "}\ \mathit{NodeList} \\
\mathit{Node} &\to \mathit{Digit}^{+}
\end{aligned}
$$

产生式 节点列表可选项 $\to \epsilon$ 允许采用空路径表示法；这种写法在语法上有效，表示除了车场之外不访问任何地点。

D.6. 最大独立集/最小顶点覆盖文法

最大独立集和最小顶点覆盖问题使用相同的输出语法，因为两者都需要指定一个顶点子集。其文法结构与定向越野问题语法相同，允许集合为空。

$$
\begin{aligned}
S &\to \texttt{"Set: ["}\ \mathit{NodeListOpt}\ \texttt{"], Objective: "}\ \mathit{Number} \\
\mathit{NodeListOpt} &\to \epsilon \mid \mathit{NodeList} \\
\mathit{NodeList} &\to \mathit{Node} \mid \mathit{Node}\ \texttt{", "}\ \mathit{NodeList} \\
\mathit{Node} &\to \mathit{Digit}^{+}
\end{aligned}
$$

最大独立集（MIS）的独立性语义约束和最小顶点覆盖（MVC）的覆盖性语义约束，由相应的修复算子而非文法本身来强制保证。

D.7. 置换流水车间调度语法

置换流水车间调度文法规定了适用于所有机器的单一作业加工顺序。其结构与旅行商问题（TSP）类似，但使用关键字”Order”以体现调度语境。

$$
\begin{aligned}
S &\to \texttt{"Order: ["}\ \mathit{JobList}\ \texttt{"], Objective: "}\ \mathit{Number} \\
\mathit{JobList} &\to \mathit{Job} \mid \mathit{Job}\ \texttt{", "}\ \mathit{JobList} \\
\mathit{Job} &\to \mathit{Digit}^{+}
\end{aligned}
$$

对于包含 n 个作业且其索引为 $\{1,\ldots,n\}$ 的实例，输入相关文法会限制 Job，使其只能生成有效的作业标识符。

D.8. 作业车间调度问题语法

作业车间调度问题文法通过为每台机器指定作业加工顺序，来表示工序到机器的分配。嵌套括号结构反映了机器与作业的两级层次结构。

$$
\begin{aligned}
S &\to \texttt{"Schedule: ["}\ \mathit{MachineList}\ \texttt{"], Objective: "}\ \mathit{Number} \\
\mathit{MachineList} &\to \mathit{Machine} \mid \mathit{Machine}\ \texttt{", "}\ \mathit{MachineList} \\
\mathit{Machine} &\to \texttt{"["}\ \mathit{JobList}\ \texttt{"]"}
\end{aligned}
$$

$$JobList \rightarrow Job \mid Job \texttt{", "} JobList$$
$$Job \rightarrow Digit^{+}$$

该文法允许灵活的机器调度顺序，而修复算子则强制执行优先约束，并确保每个工序在调度中恰好出现一次。

D.9. 语法性质与实现

所有指定的文法都是无歧义上下文无关文法，可由确定性下推自动机识别。每种文法都具有以下与受约束解码实现相关的计算属性。这些文法的递归深度有界，且与问题规模成正比，从而确保生成过程中下推自动机栈深度保持在可控范围内。所需的下推自动机状态数量为 $O(|V|)$，其中 $|V|$ 是非终结符的数量；对于每个问题类别，该数量都是常数。每个解码步骤的有效词元计算需要 $O(|\Sigma|)$ 时间，其中 $|\Sigma|$ 是词汇表大小，主要耗时在于检查终结符兼容性。

文法规范有意只确保语法有效性，而将语义约束检查交由修复层处理。这种职责分离使大语言模型能够在解码过程中专注于生成结构良好的输出，同时由修复算子处理特定问题的可行性要求；若将这些要求直接编码到上下文无关文法中，实际操作将十分困难。

E. 修复算法详情

本节完整地给出了第 3.2 节中介绍的修复算子的算法规范。每个修复算子都旨在满足定义 3.5 中提出的三项性质：可行性保证、对可行解的幂等性以及有界局部性。我们将提供详细的伪代码、复杂度分析和形式化验证，以证明每个算子都满足这些要求。

E.1. TSP修复算法

TSP修复算子用于处理两类约束违反：重复节点和缺失节点。该算法分两个阶段执行：首先，删除所有重复出现的节点，同时保留每个节点的首次出现；然后，将缺失节点贪心地插入到能使巡回路径长度增量最小的位置。

该算法在计算插入成本时通过模运算正确处理边界条件。对于包含 $|r'|$ 个节点的巡回路径，位置 i 与位置 $i \bmod |r'|$ 相连以形成一个环。贪心插入策略是旅行商问题中一种广为人知的 2-近似启发式算法，能够限制解的质量退化。

复杂度分析。 重复项移除阶段会处理每个 k 输入节点一次。使用适当的数据结构维护已访问集合，所需时间为 $O(k)$。缺失节点插入阶段会处理 $m = |V \setminus$已访问$|$ 缺失节点，并针对每个缺失节点评估 $O(n)$ 个插入位置，其中 $n = |V|$。评估每个位置时，计算插入成本需要常数时间。因此，总复杂度为 $O(k + mn)$。在最坏情况下，当 $k = O(n)$ 且$m = O(n)$ 时，复杂度可简化为 $O(n^2)$。

性质验证。我们验证 TSP修复算子满足定义 3.5 中的三项必需性质。对于可行性，算法通过构造显式确保输出 r' 恰好包含节点集 V：重复项移除阶段保证没有重复项，而缺失节点插入阶段保证所有节点都出现。对于幂等性，如果输入路径已经恰好访问 V中的每个节点一次，那么第一阶段结束后，已访问集合等于 V，缺失集合为空，第二阶段不会进行任何插入，并原样返回输入。对于有界局部性，令汉明距离 d_H 衡量位置变化的数量；算法最多修改 $|V \setminus$已访问$|$ 个位置以插入缺失节点。利用违反程度 $v(r) = |V \setminus$ 已访问$| + |$个重复项$(r)|$，可得 $d_H(r, r') \leq v(r)$，从而以常数 $\alpha = 1$建立该界。

E.2. CVRP 修复算法

CVRP 修复算子处理三种违规情况：无效的单条路线、缺失客户和容量约束违规。该算法对每条路径应用 TSP 修复，确保客户覆盖完整，然后拆分超载路径。

算法 3 TSP修复 输入：路径 $r = [v_1, v_2, \ldots, v_k]$，节点集 $V = \{0, 1, \ldots, n-1\}$，距离矩阵 d输出：有效的TSP回路 r'，恰好访问 V 中的所有节点一次 seen $\leftarrow \emptyset$ $r' \leftarrow [\,]$ 对于 $i \leftarrow 1$到 k 中的每个值，执行 如果 $v_i \in V$且 $v_i \notin$ 已在 seen 中，则 将 v_i 追加到 r' seen $\leftarrow$ seen $\cup \{v_i\}$ missing $\leftarrow V \setminus$seen 对于 missing 中的每个节点 $v \in$ ，执行 best pos $\leftarrow 0$ best cost $\leftarrow d(v, r'[0])$ // 在开头插入的成本 对于 $i \leftarrow 1$到 $|r'|$中的每个值，执行 cost $\leftarrow d(r'[i-1], v) + d(v, r'[i \bmod |r'|]) - d(r'[i-1], r'[i \bmod |r'|])$ 如果 cost $<$ best cost，则 best cost $\leftarrow$cost best pos $\leftarrow i$ 在 r'中的 best pos 位置插入 v 再t次 r'

算法 4CVRP修复

输入：路径 $R = \{r_1, r_2, \ldots, r_m\}$、客户集 $V = \{1, \ldots, n-1\}$、需求量 $\{q_i\}_{i=1}^{n-1}$、容量 Q、距离矩阵 d输出：覆盖所有客户且满足容量约束的有效 CVRP 解 对于 $k \leftarrow 1$ 到 m 执行 $r_k \leftarrow$ TSP修复$(r_k, V \cup \{0\}, d)$ // 确保路径中的每个客户最多访问一次 若车场节点 0 位于 r_k 内部，则将其移除 已覆盖 $\leftarrow \bigcup_{k=1}^{m} (r_k \setminus \{0\})$ 缺失 $\leftarrow V \setminus$ 已覆盖 对于每个客户 $v \in \in$缺失，执行 $k^* \leftarrow \arg\min_k \min_i \{d(r_k[i-1], v) + d(v, r_k[i]) - d(r_k[i-1], r_k[i])\}$ 将 v 插入路径 r_{k^*} 中，使其位于代价最小的位置 对于 $k \leftarrow 1$到 $|R|$ 执行 当 $\sum_{v \in r_k} q_v > Q$ 时执行 找到分割点 $i^* = \arg\min_{i=1}^{|r_k|-1}$ 分割代价(r_k, i)，其中 $\sum_{j=1}^{i-1} q_{r_k[j]} \le Q$ $r_{\text{new}} \leftarrow [r_k[i^*], r_k[i^*+1], \ldots, r_k[|r_k|]]$ $r_k \leftarrow [r_k[1], r_k[2], \ldots, r_k[i^*-1]]$ $R \leftarrow R \cup \{r_{\text{new}}\}$ **返回** R

拆分成本函数衡量在位置 i 处分割路径，并让两个路段都返回车场所增加的距离：

$$\text{SplitCost}(r, i) = d(r[i-1], 0) + d(0, r[i]) - d(r[i-1], r[i]).$$

分割点搜索仅限于第一个路段满足容量约束的位置，以确保通过反复拆分，最终使所有路线载荷都降至容量以下。

复杂度分析。对于 m 条路径，路线修复阶段需要 $O(m \cdot n^2)$ 的时间。对于 p 个缺失客户，缺失客户插入阶段需要 $O(p \cdot m \cdot n)$ 的时间。路线拆分阶段需要 $O(m' \cdot n)$ 的时间，其中 m' 是最终

路径数量。由于最坏情况下 $m' = O(n)$，总体复杂度为 $O(n^2 + n \cdot m)$。当 $m = O(n)$ 时，可简化为 $O(n^2)$。

性质验证。在可行性方面，该算法通过显式插入缺失客户，确保覆盖 V 中的所有客户，并通过反复拆分路径直至收敛，确保所有路径均满足容量约束。在幂等性方面，如果所有路径都是有效的 TSP 回路、覆盖了所有客户且满足所有容量约束，则不会进行任何修改。在有界局部性方面，修改次数不超过各路径中 TSP 违反约束的总数，加上缺失客户的数量以及容量约束违反次数，从而确立了所需的上界。

E.3. 定向越野问题修复算法

定向越野问题修复算子会迭代移除节点，直至满足距离预算约束，并优先移除奖赏与距离贡献比最差的节点。

算法5定向越野问题修复

输入：路径 $r = [v_0, v_1, \ldots, v_k, v_{k+1}]$，包含 $v_0 = v_{k+1} = 0$（车场）；距离预算 D；奖励 $\{p_i\}$；距离矩阵 d
输出：满足距离预算的有效定向越野问题路径 **1** $r \leftarrow$去除重复项(r) // 保留每个节点的首次出现 **2** 当总距离(r) $> D$ 且 $|r| > 2$时执行**3** $v^* \leftarrow \arg\min_{v \in r \setminus \{0\}} p_v$ 距离贡献(r, v) **4** 从 r中移除 v^* **5** 如果总距离(r) $> D$ 则 **6** 返回 $[0, 0]$ // 如果仍然违反预算，则返回空路径

7 **返回** r

路径 r 中位置 i的节点 v 的距离贡献定义为：

$$\text{DistContrib}(r, v) = d(r[i-1], v) + d(v, r[i+1]) - d(r[i-1], r[i+1]),$$

其中，i 是 v 在 r中的位置。这衡量了将节点 v 纳入路径所增加的距离。路径总距离为：

$$\text{TotalDistance}(r) = \sum_{i=0}^{|r|-1} d(r[i], r[i+1]).$$

复杂度分析。 while 循环的每次迭代都会评估 $O(n)$ 个候选节点并移除一个节点。在最坏情况下，必须移除 $O(n)$ 个节点，因此总复杂度为 $O(n^2)$。使用优先队列数据结构按奖赏与距离之比对节点排序后，复杂度可降至 $O(n \log n)$。

性质验证。 关于可行性，算法仅在满足距离约束或只剩车场时才会终止（此时显然满足预算）。关于幂等性，如果输入路径已满足距离预算，则 while 循环条件为假，输入将原样返回。关于有界局部性，移除节点的数量受初始预算违反程度的限制，从而满足所需的界限，违反程度为$v(r) = \max(0, \text{总距离}(r) - D)$。

E.4. 最大独立集修复算法

最大独立集修复算子会迭代移除涉及冲突的顶点（即两个相邻顶点都在集合中的情况），直至不再存在冲突。在冲突的顶点对中，算法会移除当前集合中度较高的顶点，因为该顶点涉及更多冲突。

算法 6最大独立集修复

输入： 集合 $S \subseteq V$，图 $G = (V, E)$
输出： 独立集 S'，其中没有相邻顶点

$S' \leftarrow S$
冲突 $\leftarrow \{(u, v) \in E : u, v \in S'\}$
当存在冲突 $\neq \emptyset$ **时执行**
  选择任意 (u, v) 个 $\in$ 冲突
  $\deg_u \leftarrow |\{w \in S' : (u, w) \in E\}|$
  $\deg_v \leftarrow |\{w \in S' : (v, w) \in E\}|$
  如果 度$_u$ >度$_v$ **则**
    $S' \leftarrow S' \setminus \{u\}$
  else
    如果 度$_v$ >度$_u$ **然后**
      $S' \leftarrow S' \setminus \{v\}$
    else
      $S' \leftarrow S' \setminus \{u\}$ // 任意打破平局
  冲突 $\leftarrow \{(u, v) \in E : u, v \in S'\}$
返回 S'

复杂度分析。 该算法最多执行 $|S|$ 次顶点移除。每次迭代都需要重新计算冲突集，通过检查所有边，这需要 $O(|E|)$ 时间。因此，最坏情况下的复杂度为 $O(|V| \cdot |E|)$。借助增量冲突跟踪，可将其改进为 $O(|E| + |V| \log |V|)$。

性质验证。对于可行性，只有在冲突集为空时算法才会终止，从而确保 S' 中任意两个顶点均不相邻。对于幂等性，若输入集合 S 已经是独立集，则初始冲突集为空，算法会立即原样返回 S。对于有界局部性，移除的顶点数等于已解决的冲突数，其上限由违反程度 $v(S) = |\{(u, v) \in E : u, v \in S\}|$ 确定。

E.5. 最小顶点覆盖修复算法

最小顶点覆盖修复算子通过添加顶点来覆盖所有未覆盖边。对于每条未覆盖边，算法会添加在未覆盖边中度较高的端点，因为该顶点可以覆盖更多边。

算法 7最小顶点覆盖修复

输入：集合 $S \subseteq V$，图 $G = (V, E)$ 输出：顶点覆盖 S'，其中每条边在 S' 中至少有一个端点1 $S' \leftarrow S$2 未覆盖边 $\leftarrow \{(u, v) \in E : u \notin S' \wedge v \notin S'\}$3 当存在未覆盖边时 $\neq \emptyset$ 4 任取一条 (u, v) 未覆盖边 $\in$5 $\deg_u \leftarrow |\{w : (u, w)$ 未覆盖边 $\in \}|$6 $\deg_v \leftarrow |\{w : (v, w)$ 未覆盖边 $\in \}|$7 如果 $\deg_u \geq \deg_v$ 则8 $S' \leftarrow S' \cup \{u\}$9 否则 $S' \leftarrow S' \cup \{v\}$11 未覆盖边 $\leftarrow \{(u, v) \in E : u \notin S' \wedge v \notin S'\}$12 返回 S'

复杂度分析。 该算法最多添加 $|V|$ 个顶点。每次迭代都需要更新未覆盖边集，这需要 $O(|E|)$ 时间。因此，最坏情况下的复杂度为 $O(|V| \cdot |E|)$。然而，借助高效的数据结构

正在追踪 u 个未覆盖的边，复杂度可以通过对每条边最多处理两次而降至 $O(|E|)$ e.

性质验证。 对于可行性，只有在所有边都被覆盖后，算法才会终止，从而确保每条边至少有一个端点属于 S'。对于幂等性，如果输入集合 S 已覆盖所有边，则未覆盖边集合为空，算法会立即原样返回 S。对于有界局部性，添加的顶点数量不超过未覆盖边的数量，因此满足所需的界限，违反程度为 $v(S) = |\{(u, v) \in E : u \notin S \wedge v \notin S\}|$。

E.6. PFSP修复算法

置换流水车间调度修复算子通过移除重复作业，并将缺失作业贪心地插入到能使最大完工时间最小的位置，确保所有作业恰好出现一次。

算法 8 PFSP修复 输入：作业序列 $\sigma = [j_1, \ldots, j_k]$，作业集 $J = \{1, \ldots, n\}$，加工时间 $\{p_{i,j}\}$，机器数量 m 输出：所有作业的有效排列 σ' 1 seen $\leftarrow \emptyset$ 2 $\sigma' \leftarrow [\,]$ 3 对于 $i \leftarrow 1$到 k 执行 4 如果 $j_i \in J$ 且 $j_i \notin$ 已见则 5 将 j_i 追加到 σ' 6 seen $\leftarrow$ seen $\cup \{j_i\}$ 7 missing $\leftarrow J \setminus$seen 8 对每个缺失作业 $j \in$ 执行 9 best pos $\leftarrow 0$ 10 best makespan $\leftarrow \infty$ 11 对于 $i \leftarrow 0$ 到 $|\sigma'|$执行 12 $\sigma_{\text{temp}} \leftarrow \sigma'[0:i] + [j] + \sigma'[i:]$ 13 $M \leftarrow$ 计算最大完工时间(σ_{temp}, $\{p_{i,j}\}$, m) 14 如果 $M <$ 小于最佳最大完工时间则 15 best makespan $\leftarrow M$ 16 best pos $\leftarrow i$ 17 将 j 插入 σ'中的最佳位置

18 **返回** σ'

最大完工时间的计算遵循定义中的递归公式 B.7：

$$C_{\sigma(k),j} = \max(C_{\sigma(k),j-1}, C_{\sigma(k-1),j}) + p_{\sigma(k),j},$$
$$\text{ComputeMakespan}(\sigma, \{p_{i,j}\}, m) = C_{\sigma(n),m}.$$

复杂度分析。 重复项移除阶段需要 $O(k)$ 时间。对于每个 $O(n)$ 个缺失作业，算法会评估 $O(n)$ 个插入位置，而每次最大完工时间计算需要 $O(nm)$ 时间。因此，总复杂度为 $O(n^2m)$。

性质验证。 为保证可行性，算法通过移除重复项并插入缺失作业，确保输出恰好包含作业集 J。为保证幂等性，若输入已经是 J 的有效排列，则缺失集为空，不会进行插入。为保证有界局部性，修改次数不超过重复项数量与缺失作业数量之和，从而满足所需的界限。

E.7. JSSP修复算法

作业车间调度修复算子确保每个作业在每台机器上恰好出现一次，然后通过对优先关系图进行拓扑排序，解决优先约束违规问题。

算法 9 JSSP修复 输入：机器调度 $S = [s_1, \ldots, s_m]$，其中 s_i 是机器 i 的作业序列，作业集 $J = \{1, \ldots, n\}$，工序机器 $\{\mu\ (o_i,\ k)\}$输出：满足优先约束的有效 JSSP 调度 1 对于 $i \leftarrow$ 1至 m 执行2 $s_i \leftarrow$ 排列修复(s_i, J) // 删除重复项，插入缺失作业 3 构建工序优先图 $G_{\text{prec}} = (O, E_{\text{prec}})$，其中：4 $O = \{o_{i,k} :\ i \in J, k \in \{1, \ldots, m\}\}$ 5 $E_{\text{prec}} = \{(o_{i,k}, o_{i,k+1}) :\ i \in J, k \in \{1, \ldots, m-1\}\}$6 计算拓扑序 τ（基于 G_{prec}）7 对于 $i \leftarrow$ 1至 m 执行 8 重新排列 s_i 以满足 τ，同时保留已符合 τ的作业的相对顺序

9 **返回** S

PermutationRepair 子程序与 PFSP修复 相似，但不进行最大完工时间优化，只需将缺失的作业追加到每台机器序列的末尾。

复杂度分析。 排列修复阶段需要 $O(m \cdot n^2)$ 时间。构建优先关系图需要$O(nm)$ 时间。拓扑排序需要 $O(nm)$ 时间。使用稳定排序对每台机器的调度进行重新排序需要 $O(n\log\ n)$ 时间。因此，总复杂度为 $O(nm\log(nm))$。

性质验证。 对于可行性，该算法通过排列修复确保每个操作恰好出现一次，并通过拓扑排序满足优先约束。对于幂等性，如果输入已经是满足优先约束的有效排列，则不会进行任何修改。对于有界局部性，修改数量受排列违反与优先约束违反的总数限制，从而满足所需的界限。

F. 扩展理论分析

F.1. 不同训练方法的梯度比较

我们对不同偏好优化方法中的梯度结构进行了正式比较。

命题 F.1 (梯度比较)。 考虑一个偏好对(y_w, y_l)，其中 $y_w \succ y_l$ 表示 y_w 优于y_l。直接偏好优化（*DPO*）、组相对策略优化（*GRPO*）和最佳锚定目标引导偏好优化（*BOPO*）的梯度更新形式如下。

对于偏好强度参数为 β的 *DPO*，其梯度为：

$$\nabla_\theta \mathcal{L}_{DPO} = -\beta \cdot \sigma(-\beta \cdot \Delta s_\theta) \cdot \nabla_\theta \Delta s_\theta,$$

其中，$\Delta s_\theta = \log \pi_\theta(y_w|\phi(p)) - \log \pi_\theta(y_l|\phi(p))$ 表示对数概率比，$\sigma(\cdot)$ 表示 *Sigmoid* 函数。

对于具有归一化优势 $\{A_i\}_{i=1}^K$ 的 *GRPO*，其梯度为：

$$\nabla_\theta \mathcal{L}_{GRPO} = -\frac{1}{K} \sum_{i=1}^{K} A_i \cdot \nabla_\theta \log \pi_\theta(y_i|\phi(p)),$$

其中，优势 A_i 是相对于组平均奖励计算的。

对于采用目标引导权重 $w(\Delta_i) = \Delta_i/\Delta$的 *BOPO*，其梯度为：

$$\nabla_\theta \mathcal{L}_{BOPO} = -\frac{1}{K-1} \sum_{i=1}^{K-1} w(\Delta_i) \cdot \sigma(-\beta \cdot \Delta s_\theta^{(i)}) \cdot \nabla_\theta \Delta s_\theta^{(i)},$$

其中，每个 $\Delta s_\theta^{(i)} = \log \pi_\theta(y^*|\phi(p)) - \log \pi_\theta(y_i|\phi(p))$ 表示最佳解 y^*与劣解 y_i之间的对数概率比。

关键区别在于不同解对梯度更新的贡献方式。在 DPO 中，每个偏好对的贡献都相同，不受质量差异大小的影响。在 GRPO 中，贡献会根据归一化优势进行调节，但归一化可能会削弱高质量解所提供的信号。在 BOPO 中，目标引导加权确保最优性差距较大的解按比例对学习作出更大贡献，从而为明显较差的解提供更强的监督信号，同时通过缩放归一化保持更新幅度有界。

命题 F.2 （方差性质）。 假设梯度方差 $Var[\nabla_\theta \log \pi_\theta（y）]$ 在所有解之间均有界，则当目标差距方差与均值平方之比 $Var[\Delta_i]/\mathbb{E}[\Delta_i]^2$小于归一化优势的方差 $Var[A_i]$时，*BOPO* 的梯度方差低于 *GRPO*。对于目标值具有自然规模且最佳解能够为计算差距提供稳定锚点的组合优化问题，该条件通常成立。

方差降低源于锚定策略。BOPO 始终将当前批次中的最佳解作为比较对象，从而建立一致的参考点，使梯度估计保持稳定。相比之下，GRPO 相对于批次均值计算优势，而批次均值可能在不同批次之间发生显著变化，进而为训练信号引入额外方差。

F.2. 端到端质量保证

定理 F.3（FALCON 端到端保证）。 设 π_θ 是一个经过 *BOPO* 训练并收敛至满足 ϵ-驻点的策略，且该驻点满足 $\mathbb{E}[\|\nabla \mathcal{L}_{BOPO}（\theta）\|^2] \le \epsilon$。当采用语法约束解码、满足定义 $\mathcal{R}$ *3.5* 的修复算子，以及使用 N 个样本进行自适应 *Best-of-* N 采样时，*FALCON* 生成的解 $\hat{y}$ 具有以下保证：

1. 格式有效性： 该解以概率 *1* 在语法上有效，从而确保评估函数 $\psi_p(\hat{y})$ 有明确定义。*2.* 语义可行性： 该解以概率 *1* 在语义上可行，从而确保 $\hat{y} \in \mathcal{X}_{C_p}$。*3.*质量界： 期望最优性差距满足：

$$\mathbb{E}[f_p(\hat{y}) - f_p(y^*)] \le \mathbb{E}[g_N] + L_f \cdot \alpha \cdot \mathbb{E}[v(\hat{y})],$$

其中 $\mathbb{E}[g_N] = O(1/N)$ 是从采样 N个解中得到的期望最优性差距（见定理 *3.11*）， L_f是目标函数相对于解距离度量的利普希茨常数， α 是修复算子的局部性常数（定义*3.5*），而 $v(\hat{y})$是修复前的违反程度。

证明。 格式有效性保证直接由定理3.3得出，该定理证明了语法约束解码确保所有生成文本都属于由问题特定语法定义的语言。
可行性保证由定理 3.6得出，该定理表明，无论输入为何，修复算子都会将任意解映射到可行域。

对于质量界，令 $\tilde{y}_N$ 表示修复前 N 个样本中的最佳解，令 $\hat{y} = \mathcal{R}(\tilde{y}_N)$ 表示修复后的最终输出。根据三角不等式：

$$f_p(\hat{y}) - f_p(y^*) = [f_p(\hat{y}) - f_p(\tilde{y}_N)] + [f_p(\tilde{y}_N) - f_p(y^*)].$$

取期望，并将定理 3.7 应用于第一项，将定理 3.11应用于第二项，即可得到所述质量界：

$$\begin{aligned}\mathbb{E}[f_p(\hat{y}) - f_p(y^*)] &\le \mathbb{E}[f_p(\hat{y}) - f_p(\tilde{y}_N)] + \mathbb{E}[f_p(\tilde{y}_N) - f_p(y^*)] \\ &\le L_f \cdot \alpha \cdot \mathbb{E}[v(\hat{y})] + \mathbb{E}[g_N].\end{aligned}$$

□

该定理表明，FALCON 的解的质量取决于两个因素：采样质量会随着样本数量的增加而提高（$\mathbb{E}[g_N] = O(1/N)$），而修复成本则会随着模型通过 BOPO 训练产生更少且更小的违规而降低（$\mathbb{E}[v(\hat{y})]$降低）。这两个因素之间的乘法交互作用解释了实证观察结果：尽管施加了硬可行性约束，FALCON 仍能保持具有竞争力的最优性差距。

F.3. 泛化界

定理 F.4 (从训练分布到测试分布的泛化)。 设 $\mathcal{D}_{train}$ 和 $\mathcal{D}_{test}$ 是定义在问题实例上的训练分布和测试分布。假设:

1. $W_1(\mathcal{D}_{train}, \mathcal{D}_{test}) \leq \delta$，其中 W_1 表示 *Wasserstein-1*距离。
2. 策略 π_θ 关于问题参数 L_π是利普希茨连续的。
3. 目标函数 f_p 关于解表示 L_f是利普希茨连续的。

那么，对于在 $\mathcal{D}_{train}$上训练的策略:

$$\mathbb{E}_{p\sim\mathcal{D}_{test}}[Gap(p)] \leq \mathbb{E}_{p\sim\mathcal{D}_{train}}[Gap(p)] + L_f \cdot L_\pi \cdot \delta,$$

其中 $Gap(p) = f_p(\hat{y}_p) - f_p(y_p^*)$ 是实例 p 的最优性差距，而$\hat{y}_p$ 是训练策略生成的解。

证明。 该证明基于利普希茨假设和 Wasserstein距离的定义。对于任意边缘分布分别为 $\mathcal{D}_{\text{train}}$ 和 $\mathcal{D}_{\text{test}}$、连接 $\mathcal{D}_{\text{train}}$ 和 $\mathcal{D}_{\text{test}}$ 的耦合 γ:

$$\begin{aligned}
\mathbb{E}_{p\sim\mathcal{D}_{\text{test}}}[\text{Gap}(p)] - \mathbb{E}_{p\sim\mathcal{D}_{\text{train}}}[\text{Gap}(p)] &= \mathbb{E}_{(p_{\text{train}},p_{\text{test}})\sim\gamma}[\text{Gap}(p_{\text{test}}) - \text{Gap}(p_{\text{train}})] \\
&\leq L_f \cdot \mathbb{E}_{(p_{\text{train}},p_{\text{test}})\sim\gamma}[\|\hat{y}_{p_{\text{test}}} - \hat{y}_{p_{\text{train}}}\|] \\
&\leq L_f \cdot L_\pi \cdot \mathbb{E}_{(p_{\text{train}},p_{\text{test}})\sim\gamma}[\|p_{\text{test}} - p_{\text{train}}\|] \\
&= L_f \cdot L_\pi \cdot W_1(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) \\
&\leq L_f \cdot L_\pi \cdot \delta.
\end{aligned}$$

对所有耦合 γ 取下确界，即可得到所需的界。 □

F.4. 修复近似比

命题 F.5 (TSP修复近似)。 用于 *TSP* 的贪心插入修复算法（算法 *3*）生成的解，其长度增量不超过插入节点所导致的最优回路长度增量的 *2* 倍。

命题 F.6 (TSP特定修复界)。 对于单位正方形 $[0,1]^2$，内有 n 个城市的 *TSP*，贪心插入修复满足:

$$f(\mathcal{R}(x)) - f(x) \leq 2 \cdot |missing| \cdot \max_{i,j} d_{ij} \leq 2\sqrt{2} \cdot |missing|,$$

其中 $|missing|$ i s 表示缺失节点的数量，而 d_{ij} 表示节点 i 与 j之间的欧几里得距离。

命题 F.7 (MVC 修复近似)。 *MVC* 的贪心修复（算法 *7*）可实现最小顶点覆盖的 *2*-近似。

G. 实现细节

G.1. 完整超参数表

完整流水线的超参数设置汇总于表7。

G.2. 下推自动机构造细节

对于上下文无关文法 $G = (V, \Sigma, R, S)$，我们使用标准的 CFG 到 PDA 的转换技术，构造一个等价的下推自动机（PDA） $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$，以进行语法约束解码。

G.2.1. PDA 组成部分

状态集定义为 $Q = \{q_0, q_{\text{acc}}\} \cup \{q_A : A \in V\}$，其中 q_0 是唯一的控制状态， q_A 是用于跟踪非终结符的辅助状态。实际上，单状态下推自动机就足够了，因为所有推导信息都编码在栈上。输入字母表 Σ 是大语言模型的词汇表（词元集），现代语言模型通常约有 $|\Sigma| \approx 32,000$ 个词元。栈

表7. 完整流水线的超参数设置。

参数	符号	值
模型配置		
基础模型	–	Qwen2.5-7B
LoRA秩	r	64
LoRA alpha	α	64
目标模块	–	q、k、v、o、gate、up、down
SFT 训练阶段		
批次大小	B	4
梯度累积步数	–	4
学习率	η	2×10^{-4}
训练轮数	–	1
预热步数	–	20
权重衰减	–	0.01
BOPO 训练阶段		
批次大小	B	8
梯度累积步数	–	8
学习率	η	1×10^{-6}
温度	β	0.1
每个实例的样本数	K	8
训练轮数	–	1
推理配置		
最小样本数	$N_{\min}$	8
最大样本数	$N_{\max}$	64
置信度阈值	τ	0.85
采样温度	T	0.7

字母表定义为 $\Gamma = V \cup \{\$\}$，其中 V 是文法 G 中的非终结符集合，$ 是栈底标记。我们使用配置 $(q_0, \$S)$ 初始化下推自动机，其中栈顶为开始符号 S，其下是栈底标记 $。最后，接受状态为 $F = \{q_0\}$；当栈中仅剩栈底标记 $ 时，我们便接受。

G.2.2. 转移规则

转移函数 δ 通过两种基本机制运作。首先，对于**ε 转移（非终结符展开）**，每个产生式规则 $A \to \alpha$ 在 R 中都会导出以下转移：

$$\delta(q_0, \epsilon, A) = (q_0, \alpha),$$

这使下推自动机能够用其右部替换栈中的非终结符。其次，对于**终结符消耗**，每个终结符号 $a \in \Sigma$都会产生以下转移：

$$\delta(q_0, a, a) = (q_0, \epsilon),$$

当终结符与输入词元匹配时，该转移会将其从栈中弹出。

G.2.3. 有效词元计算

在每个解码步骤中，生成部分输出 $y^{(t)} = v_1 v_2 \cdots v_t$ 后，PDA 会维护一个配置（q，γ），其中 q 是当前状态，$\gamma \in \Gamma^*$ 是栈内容。有效下一词元的集合通过三个步骤计算得出。

首先，我们**模拟 ε-闭包**：从当前栈配置（q，γ）出发，计算通过 ε 转移（非终结符展开）可到达的所有可能栈配置集合。这构成了下推自动机配置的 ε-闭包。

其次，我们**收集可消耗的终结符**：通过 ε 转移完全展开所有非终结符后，栈顶要么是终结符号，要么是栈底标记。有效词元恰好是与栈顶匹配的那些：

$$V_{\text{valid}} = \{a \in \Sigma : \text{stack top} = a \text{ in epsilon-closure}\}.$$

第三，我们**处理栈底**：如果 ε-闭包产生空栈（栈中只包含 $），则无法再消耗更多词元，这意味着生成结束。当栈为空且到达接受状态时，算法终止。此过程确保只允许那些能够导出语法有效推导的词元，从而在采样前有效屏蔽无效续写。

G.2.4. 对应于算法 1

有效词元　　计算过程如上所述，直接实现了主论文中算法 1 的第 5–8 行　　r:

逻辑值 ← π_θ(x, y) # 第 4 行 V_valid ← {v ∈ Σ | δ(q, v, top(γ)) ≠ ∅} # 第 5 行 for v ∉ V_validdo logits[v] ← -∞ # 第 6–8 行

基于下推自动机的方法为任何上下文无关文法提供了一种高效、声明式的计算方式 V_{valid}，无需手动枚举。

G.2.5. 计算复杂度

每个解码步骤的有效词元计算开销为 $O(|V| \cdot |Q|)$，其中 $|V|$ 是非终结符的数量，$|Q|$ 是下推自动机状态的数量。对于组合优化问题，文法通常较为简单，并具有常数规模的状态空间（例如 $|Q| = O(1)$），因此开销为 $O(|V|) = O(1)$。即使在显式进行状态跟踪的最坏情况下，与大语言模型注意力计算的 $O(d^2)$复杂度相比，该开销也可以忽略不计，其中 d 是隐藏维度。

G.2.6. 实现说明

建议在实际实现中采用多项优化。**预计算下推自动机：** 在初始化期间构建一次下推自动机，并在所有推理步骤中重复使用，以避免重复构建的开销。**缓存ε-闭包：** 缓存每个下推自动机状态的ε-闭包计算，以避免解码期间的重复展开。**位掩码表示：** 为提高效率，将有效词元集表示为覆盖词汇表的位掩码，从而通过按位操作识别用于 logit 掩码的禁用词元。**提前终止：** 监控栈深度；如果在生成预期长度之前栈已到达栈底标记，则强制终止，以防止格式错误的文法导致无限循环。

G.3. 修复算子的计算复杂度

各修复算子的时间和空间复杂度分析汇总于表8，其中 n 表示节点或作业的数量，m 表示机器或路径的数量，$|V|$、$|E|$ 表示图问题中顶点和边的数量。

表*8*。 各修复算子的时间与空间复杂度分析。

问题	时间复杂度	空间复杂度
TSP	$O(n^2)$	$O(n)$
CVRP	$O(n^2 + n \cdot m)$	$O(n + m)$
OP	$O(n^2)$	$O(n)$
MIS	$O(\|V\| \cdot \|E\|)$	$O(\|V\|)$
MVC	$O(\|E\|)$	$O(\|V\|)$
PFSP	$O(n^2 \cdot m)$	$O(n)$
JSSP	$O(n \cdot m \cdot \log(n \cdot m))$	$O(n \cdot m)$

G.4. 数据集

表9展示了数据集 七个组合优化问题的t统计量。GM、Erdős–Rényi和无标度网络表示高斯 n 混合、Erd˝os-R´enyi 和 Barab´asi-Albert 分布，分别。

表9。 所有七个组合优化问题的数据集统计信息。

问题	训练集	测试集	规模	分布	参考求解器
TSP	500K	100	10–100个节点	均匀 , GM	LKH-3
OP	500K	100	10–100个节点	均匀 , GM	COMPASS
CVRP	500K	100	10–100个节点	均匀 , GM	LKH-3
MIS	500K	100	50–500个节点	ER, BA	Gurobi
MVC	500K	100	50–500个节点	ER, BA	Gurobi
PFSP	500K	100	10–100个作业	Taillard	QIG
JSSP	500K	100	6–30个作业	Taillard	OR-Tools