基于生成式强化学习的代码缺陷修复生成方法

罗莉琼[1]
LUO Liqiong

摘 要 代码缺陷修复作为软件工程中的关键环节，其自动化智能化程度直接影响系统的健壮性与开发效率。文章研究了一种基于生成式强化学习的代码缺陷修复方法，构建缺陷上下文的状态空间与修复动作空间，引入多维特征组合设计回报函数，建立融合预训练模型与 DDPG 策略优化的修复生成架构，分析不同回报设计对训练收敛性的影响，探讨模型在真实缺陷数据集上的修复性能与泛化能力表现。该方法有助于提升代码修复过程中的智能决策水平，为面向复杂代码环境的自适应缺陷修复提供技术路径。

关键词 生成式强化学习；代码缺陷修复；深度强化学习

doi：10.3969/j.issn.1672-9528.2025.11.044

0 引言

强化学习（reinforcement learning, RL）是一种机器学习方法，强化学习的基础框架是马尔可夫决策过程，允许智能体（Agent）能够在与环境（Environment）的交互中通过试错来学习最优策略。智能体在环境中执行行动（Action），并根据行动的结果接收反馈，即奖励（Reward）。这些奖励信号指导智能体调整其策略，以最大化长期累积奖励。软件开发过程中的代码缺陷频繁出现，传统修复方法依赖人工经验，效率低且难以适应复杂语境。生成式强化学习具备自动探索策略与动态优化的能力，为代码缺陷修复提供了新思路。该方法可将缺陷定位与修复过程建模为马尔可夫决策问题，以状态 - 动作序列驱动修复建议生成，结合回报函数引导模型学习有效策略。构建合理的回报函数对修复效果有显著影响，策略优化机制直接决定修复生成质量，强化学习框架的选择影响训练效率与泛化能力。围绕问题建模、策略设计与实验验证三个层面展开，将为实现高质量的自动代码修复奠

1. 株洲开放大学 湖南株洲 412000

[4] 周志立，曹燚，孙星明 . 基于图像 Bag-of-Words 模型的无载体信息隐藏 [J]. 应用科学学报 ,2016,34(5):527-536.
[5] LIU Q, XIANG X Y, QIN J H,et al.Coverless steganography based in image retrieval of densenet features and DWT sequence mapping[J].Knowledge-based systems, 2020, 192:105375.
[6] ZHOU Z L, CAO Y, WANG M M,et al.Faster-RCNN based robust coverless information hiding system in cloud environment[J].IEEE access,2019,7:179891-179897.
[7] MENG L J, JIANG X H, ZHANG Z Z, et al. A robust coverless image steganography based on an end-to-end hash generation model[J]. IEEE transactions on circuits and systems for video technology,2023,33(7): 3542-3558.
[8] LIU H L, ZHANG C Y, WANG Z J, et al. To deliver more information in coverless information hiding[J]. Multimedia tools and applications,2023,83:7215-7229.
[9] JIANG Q Y, LI W J.Deep cross-modal hashing[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Piscataway: IEEE, 2017: 3270-3278.
[10] WOO S, PARK J, LEE J Y, et al. CBAM: convolutional block attention module[C]//Computer Vision-ECCV 2018. Berlin: Springer, 2018: 3-19.
[11] 陈云攀 . 基于小波变换和遗传算法的混沌图像加密系统研究 [D]. 南昌 : 江西财经大学 , 2023.
[12] 赵欣 . 不同一维混沌映射的优化性能比较研究 [J]. 计算机应用研究 , 2012, 29 (3): 913-915.

【作者简介】

贾颜泽（2000—），女，山东枣庄人，硕士研究生，研究方向：信息隐藏和数字图像处理。

张春玉（1979—），女，陕西汉中人，硕士，硕士生导师、副教授，研究方向：信息隐藏与数字水印、数字图像处理。

（收稿日期：2025-04-11 修回日期：2025-10-30）

定技术基础。

1 理论基础

1.1 代码缺陷修复的建模方法

代码缺陷修复过程可建模为马尔可夫决策过程（markov decision process, MDP），记作五元组 (S, A, P, R, γ)[1]。其中 S 表示状态空间，定义为待修复代码片段在上下文嵌套结构下的语义表示；A 表示动作空间，定义为可能的修复建议序列，包括插入、替换、删除等操作；P 表示状态转移函数，描述当前状态 $\boldsymbol{s} \in S$ 执行动作 $a \in A$ 后转移至下一个状态 $\boldsymbol{s}'$ 的概率分布；R 表示回报函数，用于评价状态转移过程的有效性；$\gamma \in [0,1]$ 表示折扣因子，调节未来回报的重要程度。状态定义采用编码器生成向量表示 $\boldsymbol{s} = f(c)$，其中 c 为缺陷代码上下文；f 为编码函数，通常由预训练模型 CodeBERT 等参数化实现。动作以修 token 序列表示，形式为 $a = \{t_1, t_2, \cdots, t_n\}$。在状态 s 下执行动作 a 后接收即时回报 $r = R(\boldsymbol{s}, a)$，用于驱动策略网络优化目标函数。强化学习目标为最大化期望累计回报：

$$J(\pi) = E_{\pi}[\sum_{t=0}^{T} \gamma^{t} R(s_t, a_t)] \tag{1}$$

该过程形成一个从缺陷输入到修复建议输出的交互式序列生成框架。状态 - 动作对的连续性与可微性保证了策略梯度方法在此任务中的可行性，并为下层模型训练提供统一优化目标。

1.2 强化学习策略结构设计

对于代码缺陷修复任务的高维动作空间与连续生成需求，策略网络采用 Actor-Critic 架构。Actor 网络 $\pi\theta(\boldsymbol{s})$ 以编码状态 $\boldsymbol{s}$ 为输入，输出动作 a 的概率分布或具体 token 序列，用于直接生成修复建议。Critic 网络 $Q_{\phi,i}$ 对当前状态 - 动作对进行价值评估，为策略提供回报反馈信号。在高维连续空间下，采用深度确定性策略梯度（deep deterministic policy gradient, DDPG）方法，增强生成序列的稳定性。Actor 网络基于 Transformer 解码器结构构建，结合残差连接和多头注意力机制以保持语义一致性。Critic 网络使用多层感知器对状态编码向量与生成动作进行联合建模，输出 Q 值作为当前策略优劣的量化指标。经验回放机制引入历史状态 - 动作对，使模型在训练过程中保持稳定分布。Target 网络用于延迟更新参数，以缓解学习目标的振荡问题。梯度更新采用目标值最小化策略：

$$L(\phi) = \frac{1}{N} \sum_{i=1}^{N} (y_i - Q_{\phi,i})^2 \tag{2}$$

$$y_i = r_i + \gamma Q_{\phi}(s_{i+1}, \pi\theta(s_{i+1}))$$

式中：Q 表示目标 Critic 网络；$Q_{\phi,i}$ 表示当前 Critic 网络对第 i 条轨迹的价值估计；$\pi\theta$ 表示当前 Actor 策略。该结构可在复杂语义上下文下持续优化策略，支持多步生成场景中的修复稳定性需求。

1.3 生成模型与修复任务的结合

生成式模型基于 Transformer 结构构建，利用其强大的上下文建模能力对缺陷代码序列进行编码 [2]。输入代码通过分词、位置编码与语法结构增强处理后送入编码器，输出表示用于生成阶段的策略输入状态向量。修复建议的生成过程由策略网络控制，Actor 网络输出隐向量作为解码器的初始状态，解码器按 token 级顺序生成候选补丁，采用自回归机制。该解码过程可与 Beam Search 策略结合，用以提高修复候选的多样性与保真度。Critic 网络则基于 Transformer 的中间表示评估当前生成结果的策略价值，形成状态 - 动作对的回报估计输入。为保持修复建议的语法约束与语义关联，解码器可引入 AST 先验或语法引导掩码，使模型更易生成结构合法的代码。模型输出可直接用于语义补丁匹配、编译验证或静态分析，从而与环境交互形成强化学习闭环。

2 基于生成式强化学习的代码缺陷修复方法

2.1 回报函数的构建方法

回报函数采用三维特征组合形式构建，分别为语法正确性、编译成功状态与测试执行覆盖比例，每一维特征取值范围归一化至 (0, 1) 区间，确保回报尺度一致性 [3]。每个修复建议执行后会调用语法分析器生成抽象语法树并校验是否可解析，作为语法特征的评估基础。编译特征通过构建后的工程是否成功产出可执行中间码进行判断，状态成功则为 1，失败为 0。测试覆盖特征在单位测试集中对修复后的代码执行情况进行统计，记录实际触发路径与预期路径的重合比例作为定量指标。最终回报函数形式为：

$$R(s,a) = \boldsymbol{\theta}_1 f_{\text{synatx}} + \boldsymbol{\theta}_2 f_{\text{compile}} + \boldsymbol{\theta}_3 f_{\text{test}} \tag{3}$$

式中：$\boldsymbol{\theta}_1$、$\boldsymbol{\theta}_2$、$\boldsymbol{\theta}_3$ 为待调整参数，默认设定为均值权重。

为验证回报函数在不同缺陷类型下的适用性与分辨能力，选取具有代表性的修复样例进行特征值提取与回报计算。表 1 展示各修复场景中 3 项特征指标与最终回报值的组合情况。数据中存在多种典型缺陷类型，覆盖空指针处理、边界判断、资源控制、异常处理及接口调用等维度。语法错误在部分逻辑修复中频繁出现，特别是“异常未捕获”与“重复代码段未提取”样例，其语法校验失败导致语法得分偏低。编译状态多数可达成成功输出，说明结构级错误较少，修复方案对构建过程影响较弱。测试覆盖差异体现修复行为的行

为层次有效性，如“索引越界”与“API 参数顺序错误”虽可成功编译但覆盖率偏低，说明修复未能完全恢复功能路径。回报值受三维指标共同作用影响，不完全由任一指标主导，具备可用于强化训练的评估信号特性。

表 1 不同缺陷修复样例的回报特征与结果

缺陷位置	修复方式	语法正确性 fsyntax	编译成功状态 fcompile	测试覆盖率 ftest	回报值 $R(s, a)$
NullPointer 异常处理缺失	添加空值判断逻辑	1	1	0.67	0.89
索引越界	调整循环上限	1	0.5	0.33	0.61
资源未关闭	增加 try-with 资源块	0.67	1	1	0.89
类名拼写错误	修复类名引用	0.33	1	0.67	0.67
条件判断逻辑反向	修复判断语句	1	1	1	1
异常未捕获	添加 catch 块	0	1	0.67	0.55
重复代码段未提取	提炼函数抽取重复逻辑	0.33	0.5	0	0.28
API 调用参数顺序错误	调整参数顺序	0.67	0.67	0.33	0.56

2.2 强化学习模型架构设计

整体架构包含输入编码模块、策略网络模块与价值评估模块，编码模块采用 CodeBERT 模型作为状态表示提取器，将缺陷代码上下文映射为定长向量 $\boldsymbol{s} \in \mathbf{R}^d$ [4]。策略网络采用 DDPG 结构，Actor 部分构建在 Transformer 解码器之上，输入当前状态向量，输出修复 token 的概率分布序列；Critic 部分为多层感知器结构，输入状态向量与动作 token 序列经嵌入后的拼接表示，输出当前动作序列在该状态下的 Q 值评估结果。

为增强模型收敛效果，采用双网络结构构建目标网络，用于生成稳定的动作价值估计。经验回放机制缓解数据相关性，缓冲区中保存已执行的状态 - 动作 - 回报 - 下状态元组，用于打乱训练数据顺序，提高样本利用率。损失函数由均方误差构成，优化目标为使 Critic 网络输出值逼近目标值 $y = r + \gamma Q_{\phi}(s', \pi_{\theta'}(s'))$。Actor 网络使用策略梯度更新，采用确定性策略输出避免高维空间下的采样效率下降问题。训练过程分阶段进行，先预训练编码器与 Actor 部分，再引入 Critic 训练以稳定收敛曲线。

2.3 生成式强化学习算法流程

修复建议生成过程以状态行为对的采样为核心，策略网络针对每个缺陷样本输入生成多个修复候选序列，每个序列构成轨迹集合，存入经验缓冲池，用于后续策略优化训练。每条轨迹中的回报计算采用加权累积的方式，定义为：

$$R_{\varsigma} = \sum_{t=1}^{T} \gamma^{t-1} r_t \tag{4}$$

式中：γ 为折扣因子；r_t 为第 t 步对应的即时回报。该累计回报值作为策略评估依据，用于更新 Actor 与 Critic 网络参数。

候选修复行为在静态分析与语法验证阶段生成回报信号，并作为策略学习的反馈引导，强化网络生成语义与结构均合法的修复输出。轨迹采样中引入噪声扰动提升策略探索性，避免陷入策略局部最优区域。优化策略基于 DDPG 结构，每轮从经验缓冲池中抽取固定批量轨迹数据，分别对 Critic 网络与 Actor 网络进行参数更新。Critic 部分利用目标值计算误差最小化，提升估值精度；Actor 部分最大化当前状态下所获回报的预期值，强化最优行为输出倾向。

目标网络采用软更新机制控制主从网络权重同步速度，防止策略振荡过大。训练过程设定最大步数与收敛阈值，策略稳定后终止更新。训练完成后所得策略可直接用于实际缺陷样本的修复建议生成。整体生成式强化学习修复流程如图 1 所示。

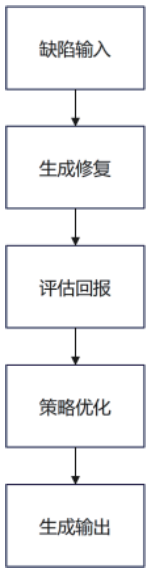

图 1 生成式强化学习修复流程图

3 实验及结果分析

3.1 实验环境

实验数据采用 Defects4J 和 CodeXGLUE 中的 Java 修复任务子集，选取包含语法错误、空指针异常、逻辑判断失误和 API 误用等典型缺陷类型的开源项目 [5]。每个数据集中从官方标注的修复前代码中提取缺陷位置及修复片段作为监督信号，对照真实修复内容构建训练轨迹。数据集中的修复样例被分割为训练集、验证集和测试集，保证数据分布一致。模型输入为缺陷位置前后固定窗口的代码上下文，输出为修复建议 token 序列，所有代码均转换为 AST 编码序列后进行 BPE 分词处理，便于模型处理复杂结构。

实验在 Ubuntu 20.04 环境下运行，配备 Intel Xeon Silver 4310 CPU、NVIDIA RTX A6000 GPU 与 256 GB 内存。软

件环境基于 Python 3.9，PyTorch 1.13，Transformers 4.24 与 TensorFlow 2.10 版本。预训练编码器初始化选用 CodeBERT 权重参数，冻结低层 Transformer block，仅微调高层表示。DDPG 参数设置为学习率 1e-4，折扣因子 0.99，目标网络软更新系数 0.005，经验缓冲区大小为 100 000 条轨迹。每次训练迭代中从缓冲区中抽取 64 条轨迹样本作为 batch 执行优化，每轮最大训练步数设定为 300 000 步，验证指标每 5 000 步记录一次，选取最优权重用于测试集推理评估。

3.2 实验对比与结果分析

实验对比采用三类方法作为参照组，分别为监督学习方法 SupTrans、基于策略梯度优化的 PPOFix 以及依赖静态规则和补丁模板的 HeuristicPatch。SupTrans 使用标准 Transformer 编码 - 解码结构，训练过程中以真实修复片段作为目标序列，无强化反馈机制；PPOFix 基于 Proximal Policy Optimization 算法进行策略更新，策略网络使用 RNN 结构表示生成轨迹，价值函数设计与修复准确性绑定；HeuristicPatch 利用人工设计的缺陷 - 修复模式对进行枚举替换，基于语法树结构匹配位置并填充固定修复模板。

本文方法以 GRLFix 表示，采用生成式强化学习架构，基于 DDPG 策略优化与 Transformer 生成模块联合建模，不使用教师强制策略，采用回报函数反馈驱动生成路径偏移，网络参数全程由轨迹回报估值控制。对比指标涵盖修复建议的准确条数、平均训练收敛所需步数与跨项目样本上的有效建议条数。准确条数统计生成建议与真实修复片段在抽象语法树（AST）结构上完全匹配的个数，排除语义相近但结构偏差样例。训练收敛步数记录损失函数稳定进入波动范围内所需累计步数。跨项目测试采用与训练集完全无重合的项目代码，对模型输出建议与真实补丁的结构匹配情况进行评估。表 2 为四种方法在三类指标下的对比数据。

表 2 各方法修复效果与训练表现对比

方法	修复建议准确条数	平均收敛步数	跨项目匹配样本数
SupTrans	328	105 000	137
PPOFix	302	141 000	121
HeuristicPatch	217		94
本文方法（GRLFix）	356	89 000	149

GRLFix 在修复建议准确条数上显著高于其他方法，SupTrans 虽在结构学习能力上具有优势，但缺乏行为反馈导致其误修率偏高。PPOFix 在策略训练中收敛速度较慢，收敛步数超过 14 万，受制于策略梯度估计波动。HeuristicPatch 无训练过程，其修复准确条数最低，缺乏上下文理解能力是主要原因。本文方法在平均收敛步数上优于其他强化学习方法，得益于回报函数结构对轨迹收敛性的正向筛选作用，使高价值轨迹得以快速集中于经验缓冲池内。

在跨项目迁移测试中，GRLFix 表现出最强的泛化能力，在缺失训练样本的目标项目上生成 149 条有效修复建议，相较于 SupTrans 多出 12 条，较 PPOFix 多出 28 条。基于生成式强化学习的策略可在缺陷模式变化中形成自适应的 token 生成路径，而不依赖单一训练域结构，提升在目标领域之外的预测能力。HeuristicPatch 缺乏任何训练机制，其迁移性能受限于人工规则覆盖率，跨域扩展能力最弱。

4 结论

文章构建了基于生成式强化学习的代码缺陷修复方法，将缺陷修复任务建模为马尔可夫决策过程，设计语法合法性、编译可通过性与测试覆盖率构成的多维回报函数，采用 DDPG 策略优化与 Transformer 生成模型融合架构，构建状态驱动的端到端修复生成系统。实验在真实缺陷数据集上进行，结果显示该方法在修复建议准确性、策略收敛速度及跨项目泛化能力上均优于对比方法，强化反馈在引导修复序列收敛方面具备有效性。生成式强化学习框架在可控修复生成任务中具备结构优势，适用于多类语义缺陷场景，未来可结合多轮语义验证与增量更新机制构建更具交互性的修复系统。

参考文献：

[1] 靳恒清 . 基于生成式人工智能强化学习的发展及应用 [J]. 数字技术与应用 ,2024,42(9):7-9.

[2] 尹义鹏 , 施水才 , 黄自力 . 基于强化学习的生成式人工智能综述 [J]. 软件导刊 ,2025,24(1):183-192.

[3] 罗睿锋 . 基于强化学习的桁架结构生成式设计算法研究 [D]. 上海 : 同济大学 ,2022.

[4] 温炜亮 . 基于深度强化学习的恶意代码生成对抗研究 [D]. 广州 : 广东工业大学 ,2022.

[5] 李新童 . 基于逆向强化学习的恶意代码对抗生成技术的研究与设计 [D]. 北京 : 北京邮电大学 ,2022.

【作者简介】

罗莉琼（1981—），女，湖南株洲人，硕士，副教授，研究方向：计算机应用、人工智能、数据科学、高等教育。

（收稿日期：2025-06-13 修回日期：2025-11-11）