

基于多模型集成的代码语义理解优化方法研究

门汝熹¹
MEN Ruxi

摘要

针对代码语义理解中多维特征难以统一建模的问题,文章提出一种基于多模型集成的优化方法。构建由文本、结构与交互语义编码器组成的多通道特征提取模块,通过注意力引导的动态加权机制实现融合表示。引入联合训练策略以实现异构语义空间的协同优化。在代码智能理解评估基准(CodeXGLUE)数据集上进行代码分类、缺陷检测与克隆检测实验。相较主流单模型与集成方法,在准确率、 F_1 值及平均准确率均值(mean average precision, mAP)等指标上表现更优。消融实验表明,结构建模与语义交互模块贡献显著,融合与优化策略增强模型稳定性,验证方法在复杂语义场景下的有效性与适应性。

关键词

代码语义理解;多模型集成;异构特征建模;动态融合机制;联合训练;抽象语法树

doi: 10.3969/j.issn.1672-9528.2026.07.007

0 引言

近年来,代码语义理解在软件缺陷检测、代码克隆分析与智能审计中发挥关键作用。源代码同时具备语言序列、结构拓扑与上下文依赖等多重语义,单一建模策略难以全面捕捉其深层信息,限制模型的泛化与迁移能力。为提升语义建模能力,多模型融合与异构语义集成成为研究重点。

魏政等^[1]提出结合语义还原机制与大模型知识迁移可提升混淆代码识别精度,验证深层语义建模在安全分析中的有效性。周放等^[2]基于源代码迁移设计编译优化策略,强调结构语义对迁移过程的稳定支撑。王彦昕等^[3]在跨形态相似性识别中指出结构建模对语义保持与对齐的重要性。张璐等^[4]构建多元语义图模型,实现对二进制语义的高精度建模,为

多模态集成提供路径参考。

在此基础上,整合文本、结构与交互三类语义通道,构建多通道编码器架构,设计注意力引导的动态融合机制,提出统一的多模型集成优化框架,增强语义空间的协同建模能力与表达一致性,以期复杂代码语义理解任务提供具备可扩展性与鲁棒性的建模方案。

1 多模型集成概述

多模型集成通过构建多元表征空间,缓解单一模型在编码粒度与语义视角上的局限。在代码语义理解任务中,语言模型建构序列上下文语义,结构模型捕捉抽象语法结构的拓扑依赖,交互模型建模多通道间的语义协同。模型间的异构性带来特征空间的不一致,需通过融合策略进行对齐与调控。集成方式涵盖输入共享、编码耦合、输出对齐,融合机制包

1. 南宁市勘测设计院集团有限公司 广西南宁 530000

[7] 柏栋,于英,宋亮,等.面向军用车辆细粒度检测的遥感图像数据集构建与验证[J].中国图象图形学报,2024,29(12):3564-3577.

[8] 禹文奇,程堪,王美君,等.MAR20:遥感图像军用飞机目标识别数据集[J].遥感学报,2023,27(12):2688-2696.

[9] Sohan M, Sai Ram T, Rami Reddy C V. A review on YOLOv8 and its advancements[C]//International Conference on Data Intelligence and Cognitive Informatics. Berlin:Springer,2024: 529-545.

[10] Ci Wenyan, Hou Hongyi, Lu Shan, et al. Multi-module enhanced low-light object detection: from image enhancement

to adaptive feature fusion[J]. Advanced Engineering Informatics, 2026, 69: 104097.

【作者简介】

王一凡(1997—),男,河南郑州人,硕士,助教,研究方向:深度学习、计算机视觉。

樊登华(1997—),男,河南新乡人,硕士,助教,研究方向:通信工程、计算机。

吴栋(1990—),男,河南商丘人,硕士,助教,研究方向:控制工程。

(收稿日期:2025-12-01 修回日期:2026-07-08)

括注意力分配、门控调节与互信息筛选。为保障表示稳定性与训练收敛性，融合机制须具备可微性与梯度传导一致性。集成效果取决于模型结构化差异性的利用及融合策略的动态适应能力。

2 多模型集成优化框架设计

2.1 整体架构设计

多模型集成框架采用分层解耦结构，由输入表征、语义建模、融合调控与任务解码四个子系统组成，如图1所示。

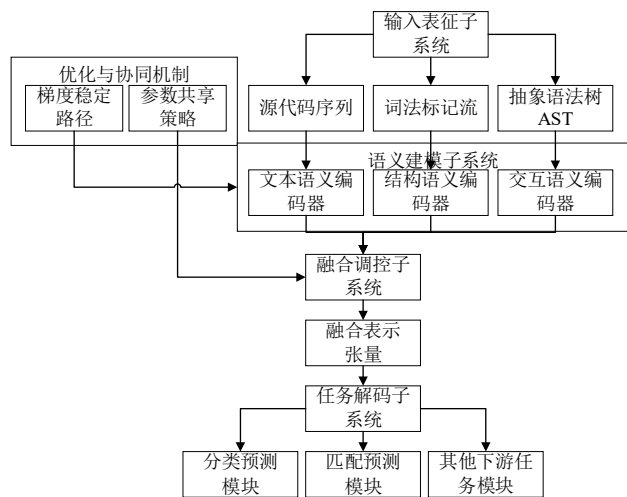


图1 多模型集成架构图

输入表征构建原始序列、词法标记与抽象语法树三视图表示。语义建模并行部署文本、结构与交互编码器，分别建模序列特征、结构拓扑与跨模态语义，输出统一维度张量。融合调控引入注意力机制，依据表示差异与上下文相关性动态生成融合权重，整合异构语义。任务解码构建多任务映射空间，利用共享投影层保持语义一致性^[5-6]。各子系统通过梯度路径与参数共享协同训练，支持多粒度语义融合与异构模型并行优化，确保结构完整与训练稳定性。

2.2 核心特征提取模块

(1) 文本编码器：基于Transformer架构的CodeBERT，参数规模 125×10^6 ，含12层Transformer单元，每层设12个注意力头，隐藏维度768。输入采用BPE分词，最大序列长度512，相对位置编码增强方向感知。嵌入层设置Dropout率0.1，优化器为AdamW，初始学习率 2×10^{-5} ，批大小16。输出为token级上下文向量，构建序列语义基础。

(2) 结构编码器：采用双向门控图神经网络(Bi-GGNN)建模抽象语法树(AST)拓扑，AST由tree-sitter生成，平均包含180个节点，边类型包括语法边、控制流边与数据依赖边。初始节点向量为128维token嵌入与64维标签编码拼接构成。Bi-GGNN设3层图计算单元，单层步长为2，通过门控机制调控信息传播方向，保持多轮更新下的语义稳定性。

(3) 交互编码器：采用4层Cross-Attention模块实现文本与结构语义显式对齐，每层含8个注意力头，维度512。核心计算表达为：

$$\text{CrossAttn}(\mathbf{Q}_s, \mathbf{K}_t, \mathbf{V}_t) = \text{softmax}\left(\frac{\mathbf{Q}_s \mathbf{K}_t^T}{\sqrt{d}}\right) \mathbf{V}_t \quad (1)$$

式中： $\mathbf{Q}_s$ 为文本编码输出映射矩阵， $\mathbf{K}_t$ 、 $\mathbf{V}_t$ 分别为结构编码器的键与值， d 为注意力维度^[7-8]。该结构引导语义关联计算，强化跨模态耦合。交互模块引入残差与层归一化机制，并在训练初期冻结结构侧参数，抑制优化初期的噪声干扰。

2.3 动态融合机制

融合机制基于注意力驱动的可学习加权策略，按通道维度对文本、结构与交互三类编码结果进行动态集成。设各编码器输出分别为 H_t 、 H_s 、 H_i ，经维度对齐后输入融合模块。核心采用门控注意力机制生成融合权重向量 $\alpha=[\alpha_t, \alpha_s, \alpha_i]$ ，满足归一性约束 $\sum \alpha_j=1$ 。融合语义表示计算公式为：

$$H_f = \alpha_t H_t + \alpha_s H_s + \alpha_i H_i \quad (2)$$

式中： H_f 为融合后的全局语义表示，权重 α 由三层感知机生成，输入为各编码器全局平均池化向量，经Softmax归一处理^[9-10]。模块内置Dropout(率0.2)防止融合偏置，训练阶段引入正则项控制权重过拟合。部署于NVIDIA A100(40 GB显存)，模块显存占用约3.8 GB，前向推理延迟11 ms，保证高维语义融合下的训练可控性与推理效率。

2.4 联合训练策略

联合训练通过参数联动机制实现多通道梯度同步更新，结合任务驱动与表示对齐目标。各编码器在统一调度图中并行构建，融合后连接任务解码器，形成完整反向路径。主损失采用交叉熵或对比损失，辅助损失以 L_2 范数约束各编码器输出的全局表示，抑制表征偏移。训练调度采用静态图构建与动态图执行混合策略，冻结Transformer底层三层，仅更新上层以提取高阶特征。优化器选用AdamW，初始学习率 2×10^{-5} ，调度方式为线性预热与余弦退火结合，总轮数40，Warm-up比例10%，梯度裁剪阈值设为1.0。训练平台为NVIDIA A100GPU，单卡显存40 GB，并行度设为4，批次规模32，训练时长控制在12 h内。显存通过通道划分方式分配，各编码器独立缓存，避免张量冲突。多模态参数同步通过NCCL完成，主干融合模块收敛波动小于1.7%。

3 关键实现技术与优化

3.1 代码预处理与对齐

源代码采用UTF-8编码标准，统一换行符与缩进方式，基于tree-sitter v0.20.3进行词法解析与语法建树，构建抽象语法树(AST)以支持多语言结构建模。AST平均节点数为180，边类型规约为语法依赖边、控制流边、调用边与属性

边共 4 类，分别嵌入为边标签特征；每节点记录起止字符位置与类型索引，用于结构编码输入。代码按函数级划分，经 RoBERTa BPE 分词器编码，词表大小 50 265，最大序列长度 512，超限片段采用窗口大小 384 滑动切分，片段间保持 token 级重叠。分词后每个 token 回映至原始字符区间生成索引序列，特殊符号与非字母字符经正则归一化处理以维持语义一致性。文本与结构表示之间构建稀疏对齐矩阵 $M \in \{0,1\}_{n \times m}$ ，其中 n 为 token 数， m 为 AST 节点数，对齐规则为：

$$M_{i,j} = \begin{cases} 1, & \text{if token}_i \in \text{Span}(\text{AST}_j) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

该矩阵桥接两类编码器表示空间，支持跨模态语义对齐；对未对齐 token 应用位置掩码并设零值以避免对齐误差传播。结构部分输入图神经网络前编码为稀疏邻接矩阵 $A \in \{0,1\}_{m \times m}$ 与对应边类型向量，结合边标签嵌入构成图输入张量。最终数据统一保存为 JSONL 格式，单样本平均大小 11.2 kB，在 Intel Xeon Gold 6348 (2.60 GHz, 64 核) 平台下加载速率为 420 样本/s，I/O 延迟控制在 12 ms 以内。

3.2 模型交互机制

该机制采用双向语义对齐结构，通过跨模态注意力桥接文本与结构表示，增强语义对称性与耦合能力。输入张量维度统一为 512，交互模块构建对称式双路注意力结构，分别以一方为查询、另一方为键值，实现上下文对齐。引入耦合注意力算子捕捉异构分布间的联合激活区域，并以卷积增强相似性度量替代常规内积计算，提升细粒度对齐效果。相似度计算为：

$$S_{ij} = \tanh((H_i^{(t)} \odot H_j^{(s)})W + b) \quad (4)$$

式中： $H_i^{(t)}$ 、 $H_j^{(s)}$ 分别表示第 i 、 j 个位置的向量， $\odot$ 表示 Hadamard 积， $W \in \mathbb{R}^d$ 、 $b \in \mathbb{R}$ 表示可学习参数^[11]。该相似度矩阵用于构建注意力掩码图，过滤无效映射。交互模块在 NVIDIA A100 上并行加速，采用混合精度调度，推理延迟 3.1 ms，显存占用 2.1 GB。训练阶段采用参数冻结式逐层解锁策略，初期冻结结构编码器，后期联合调优，优化器为 AdamW，梯度累计步长设为 2，以控制显存峰值。

3.3 长代码处理策略

为解决长代码输入超出编码器最大序列限制的问题，采用固定窗口滑动策略将源代码按长度 384 的子词窗口进行切片，滑动步长为 256，最大切片数控制在 8 段以内，切片间引入重叠区域缓解上下文边界信息断裂，平均样本产生 4.2 段，最长段 token 数为 498。结构信息处理方面，引入跨段 AST 融合机制，使用 tree-sitter 抽取各段语法结构

后，基于全局唯一标识进行节点重映射，保持调用边、控制边与变量定义边等跨段依赖的拓扑完整性，新增虚拟桥接节点统一对接段间入口与出口。融合后语法图平均节点数增长 12%，邻接矩阵稠密度控制在 0.017，保证结构紧凑性与计算资源平衡。编码输出经线性映射归一化至统一空间后，采用全局加权池化进行语义聚合，段权重由注意力机制生成并动态更新。聚合张量作为融合模块输入，以统一视图参与后续建模。该策略部署于 NVIDIA A100 平台，训练阶段单批显存占用峰值 2.8 GB，平均推理耗时 5.2 ms，满足长序列处理过程中的结构连续性与表示对齐要求。

4 实验设计与评估

4.1 实验设置

实验基于 CodeXGLUE 任务集，涵盖代码分类、缺陷检测与克隆检测三类任务，分别对应 Java 与 C 语言，覆盖属性预测与结构相似建模两类范式。数据经统一清洗处理，文本采用 BPE 分词，序列上限 512；结构信息通过 tree-sitter 解析生成 AST 并构建异构图。各任务样本数分别为 53 000、21 854 与 2 000 对，平均 token 长度为 75、120 与 320。分类与缺陷检测任务评估指标包括准确率、精确率、召回率与 F_1 值，克隆检测采用 MAP 与 MRR。实验部署于 NVIDIA A100 GPU (40 GB 显存) 平台，主控服务器为 Intel Xeon Gold 6348 (2.60 GHz, 64 核)。训练使用 AdamW 优化器，初始学习率 1×10^{-4} ，批大小 32，总轮数 10，引入 Early Stopping 防止过拟合。

4.2 结果与分析

4.2.1 主实验

各模型在三个典型任务上进行了性能对比实验。评估指标采用准确率 (Accuracy)、 F_1 值、平均准确率均值 (mAP) 与平均倒数排名 (mRR)，反映分类、检测与匹配任务在样本判定与排序上的建模能力。 F_1 值用于度量精确率与召回率的调和性，其定义为：

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad (5)$$

式中： P 为精确率 (Precision)， R 为召回率 (Recall)，二者均由预测标签与真实标签之间的匹配关系计算得出。主实验对比了七种主流模型与集成方法，所有模型在相同硬件环境与训练轮次下运行，输入格式统一，评估指标基于全量测试集计算。表 1 展示各模型在三项任务上的主要性能。

由表 1 可见，多模型集成方法在三类任务上均取得最优结果，尤其在克隆检测任务中 mAP 和 mRR 分别较 GraphCodeBERT 提升 1.8% 和 1.8%。 F_1 值方面，相较于单模型提升稳定，在缺陷检测任务中达成 3.5% 的增长，验证多

表 1 主实验性能结果

模型	分类 ACC	分类 F_1	缺陷 检测 F_1	缺陷检测 Recall	单位: %	
					克隆检测 mAP	克隆检测 mRR
CodeBERT	84.20	83.50	72.60	71.90	89.70	90.40
GraphCodeBERT	85.30	84.10	74.20	73.10	91.20	91.90
ASTNN	80.70	79.30	68.40	66.80	86.50	87.60
GIN	78.90	77.60	65.30	64.10	84.20	85.70
HGT	83.10	82.40	71.10	70.50	88.90	89.60
Bagging	85.80	84.60	73.50	72.40	91.40	92.00
Late Fusion	85.60	84.30	73.80	72.90	91.50	92.20
多模型集成方法	87.20	86.00	76.10	75.30	93.00	93.70

通道语义集成对多维特征建模的增强能力。

4.2.2 消融实验

为验证各模块在整体性能中的贡献,设计四种消融设置,分别去除结构语义编码器、交互编码器,替换融合方式,以及取消联合训练策略。在缺陷检测任务上对 F_1 值和 Recall 进行测试。以完整模型为参考,其他配置保持不变,仅调整目标模块,测试集保持一致,训练参数固定。消融实验性能结果如表 2 所示。

表 2 消融实验性能结果

实验配置	F_1 值	单位: %	
		召回率	
完整模型 (默认配置)	76.10	75.30	
去除结构语义编码器	72.80	71.40	
去除交互语义编码器	73.50	72.20	
动态融合替换为平均融合	74.00	72.90	
关闭联合训练 (单独预训练各模块)	73.20	71.60	

由表 2 可见,结构语义编码器的缺失对性能影响最显著, F_1 值下降 3.3 个百分点,说明 AST 结构信息对语义建模的重要性。动态融合机制与交互机制的去除亦带来性能退化,体现语义通道间的信息交互与权重调度在提升整体判别能力中的关键作用。

5 结论

本文围绕代码语义理解任务中的异构语义建模难题,提出多模型协同优化框架,构建包括文本、结构与交互通道在内的多源特征编码体系,设计可学习的动态融合机制实现语义表示统一。实验结果表明,该方法在多个典型任务上均实现性能提升,具备较强的适应性与扩展性。消融分析验证各

模块在整体架构中的独立价值,结构建模与交互机制对复杂语义关系建构具有关键作用。结论印证多模型集成策略对提升代码语义建模深度与表达完整性具有积极意义。未来将聚焦跨语言建模与低资源场景下的迁移鲁棒性问题。

参考文献:

[1] 魏政,何树果,程度,等. 基于语义复原和大模型的无文件混淆攻击识别[J]. 信息安全研究,2025,11(12):1125-1133.
[2] 周放,刘茂福,李珊枝. 基于源代码迁移的编译器优化方法研究[J]. 计算机工程与科学,2025,47(10):1890-1900.

[3] 王彦昕,贾鹏,范希明,等. C/C++ 代码跨形态相似性检测技术研究[J]. 信息安全学报,2025,25(10):1627-1638.
[4] 张璐,贾鹏,刘嘉勇. 基于多元语义图的二进制代码相似性检测方法[J]. 信息安全学报,2025,25(10):1589-1603.
[5] 杨宏宇,王云龙,胡泽,等. 基于跨模态协同表示学习的二进制代码相似性检测方法[J]. 电子学报,2025,53(4):1279-1292.
[6] 郭曦,王盼. 面向代码重用检测的程序语义分析模型[J]. 通信学报,2024,45(12):179-196.
[7] 董以越,金大海,宫云战. BscCS: 结合协同注意力的双向语义扩充代码搜索模型[J]. 小型微型计算机系统,2025,46(10):2531-2540.
[8] 陈邳睿,陆雪松. 基于开源代码大语言模型提示的学生代码修复[J]. 华东师范大学学报(自然科学版),2024(5):93-103.
[9] 屈薇,周栋,赵文玉,等. 融合多种语义特征的代码摘要生成方法[J]. 中文信息学报,2023,37(11):81-90.
[10] 杨泽洲,陈思榕,高翠芸,等. 基于深度学习的代码生成方法研究进展[J]. 软件学报,2024,35(2):604-628.
[11] 王敏,潘兴禄,邹艳珍,等. 面向代码审查的细粒度代码变更溯源方法[J]. 软件学报,2023,34(10):4705-4723.

【作者简介】

门汝熹(1985—),男,广西南宁人,本科,工程师,研究方向:系统开发及项目管理等。

(收稿日期:2025-12-05 修回日期:2026-07-09)