UbD 逆向教学设计在编译原理课程教学改革中的应用

王飞** 奚雪峰 吴宏杰 郝永静 姜晓彤

苏州科技大学电子与信息工程学院，苏州 215009

摘 要 针对普通本科院校《编译原理》课程教学中存在的理论抽象难理解、实践环节碎片化、教学评价滞后等问题，引入 UbD 逆向教学设计理论，构建“预期学习目标—评估证据—学习体验”三阶段逆向教学框架。依据 UbD 理论将教学目标分为掌握知能、理解意义、学习迁移三个层次，确立三层次预期学习目标；遵循“评估先行”原则设计表现性任务、阶段性验证和期末考核三类评估体系；将 WHERETO 七要素精简为 W-H-E-O 框架，设计“L1 基础验证—L2 模块实现—L3 综合设计”三阶递进实验体系。在 160 名学生中实施，与 93 名对照组相比，实验组课程平均成绩从 71.4 分提高至 74.7 分（提高了 3.3 分），项目通过率提高了 24.6 个百分点，课堂参与率提高了 25.9 个百分点，81.9%的学生认同思政内容对理解国产编译技术有帮助。实践表明，UbD 逆向教学设计框架能有效降低《编译原理》课程的理论门槛，提升学生综合实践能力和课堂参与度，为普通本科院校计算机核心课程教学改革提供了可复制的实践路径。

关键字 UbD 逆向教学设计，编译原理，教学改革，三阶递进实验，课程思政

Application of UbD Backward Design in the Teaching Reform of Compilation Principles Course*

Wang Fei; Xi Xuefeng; Wu Hongjie; Hao Yongjing; Jiang Xiaotong

School of Electronics & Information Engineering
Suzhou University of Science and Technology,
Suzhou 215009, China;
fwang@usts.edu.cn

Abstract—To address the challenges of abstract theory, fragmented practical training, and delayed evaluation in Compiler Principles courses at ordinary undergraduate institutions, this study introduces Understanding by Design (UbD) backward design and establishes a three-stage framework: defining expected outcomes, determining assessment evidence, and designing learning experiences. Based on UbD's three objective levels, namely acquisition, meaning making, and transfer, we set corresponding outcome tiers. Adhering to the principle of placing assessment prior to instruction, we develop three assessment types: performance tasks, phased verification, and end-of-term examinations. The WHERETO elements are streamlined into four key components, designated as W, H, E, and O, and a three-level progressive experiment system is implemented, consisting of L1 verification, L2 module implementation, and L3 comprehensive design. Applied to 160 students, with a control group of 93, the experimental group achieved a 4.6% increase in average score, a project pass rate increase of 24.6 percentage points, and a class participation rate increase of 25.9 percentage points. Moreover, 81.9% of students affirmed that the integrated ideological and political content deepened their understanding of domestic compiler technologies. The findings demonstrate that the UbD framework effectively lowers the theoretical threshold, improves comprehensive practical skills and engagement, and offers a replicable reform model for core computer science courses at similar institutions.

Keywords—UbD backward design, Compiler Principles, teaching reform, three-level progressive experiment system, curriculum ideological and political education

1 引 言

《编译原理》是计算机类专业核心课程，通过系统讲授词法分析、语法分析、语义分析与中间代码生成、目标代码生成与优化等编译过程，以培养学生系统思维与复杂工程问题求解能力。然而，该课程涉及的形式语言、自动机理论、语法分析算法等高度抽象理论内容，被公认为是计算机专业最具挑战性的课程之一[1]。对于普通本科院校学生而言，抽象思维能力参差不齐，前期编程基础相对薄弱，使得传统教学模式下教学效果普遍不理想。

当前普通本科院校《编译原理》教学主要存在三类突出问题：一是理论教学与工程实践脱节[2]，课程侧重理论推导教学，使得学生难以建立对编译器工作机制的完整认知；二是实践体系碎片化，现有实验多为孤立模块验证任务，缺少递进式训练链路，导致学生对编译过程缺乏整体认知，制约了工程素养的培养；三是教学评价体系滞后，课程考核以期末考试为主，

*基金资助：全国高等院校计算机基础教育研究会项目（2024-AFCEC-550），苏州科技大学“本科教学工程”教学改革与研究项目（2025JG-45）

过程性考核薄弱，无法实时跟踪学生的学习进度并进而纠正学生的学习漏洞[3]。同时，多数院校课程思政仍停留在表层，未能与专业知识教学逻辑有机融合[4]。

针对上述问题，国内外学者已从 OBE 理念下的实验考核体系构建[5][6]、产教融合背景下的课程综合改革[7][8]、基于 LLVM IR 的实验方案设计[9]等角度进行了探索。这些研究在提升学生实践能力方面取得了一定成效，但大多聚焦在对教学环节的局部调整，缺乏从全局层面对课程教学目标、评估与教学三者逻辑关系进行系统重构，未能从根源解决核心教学痛点。UbD（Understanding by Design）逆向教学设计理论由 Wiggins 和 McTighe 提出[10]，以“确定预期结果—制定评估证据—设计学习体验”为核心流程，秉持“评估先行”理念，并明确将教学目标分为掌握知能（Acquisition，A）、理解意义（Meaning Making，M）、学习迁移（Transfer，T）三个层次，可实现教学目标、过程与评价的深度统一，契合高阶工程人才培养需求。近年来，UbD 理论在高校计算机专业课程中已显示出较好的应用价值[11]，但在《编译原理》这类高难度、强抽象课程中形成系统化落地方案。

基于此，本文将 UbD 逆向教学设计理论系统引入《编译原理》课程教学：针对理论实践脱节，UbD 第一阶段明确以“学习迁移”为顶层目标，引导学生从“会做理论题”转向“能阅读和调试真实编译器”；针对实践碎片化，设计“L1 基础验证—L2 模块实现—L3 综合设计”三阶递进实验体系，将孤立实验串联为从模仿到完整构建的工程实践链条；针对教学评价滞后，UbD 第二阶段“评估先行”，构建表现性任务、阶段性验证与反思报告相结合的全过程评价体系，使评估贯穿教学始终。同时，将课程思政自然融入各教学阶段，以“编译技术自主化”为主线实现价值引领。通过教学实践检验该框架在降低理论门槛、提升学生综合实践能力方面的有效性。

2 UbD 理论驱动的课程教学改革设计

2．1 改革理念与指导思想

UbD 逆向教学设计理论以“三阶段逆向设计”为核心实施路径，依次完成确定预期学习结果、确定评估证据、设计学习体验三项任务，从而达成课程目标、评价方式与教学活动的高度统一[10]。与传统正向教学“确定内容、设计活动、制定评估”的逻辑不同，UbD 遵循“以终为始”的原则，将评估前置作为教学设计的逻辑起点，从根源上避免“教考脱节”的问题。UbD 逆向设计与传统顺向设计的逻辑差异如图 1 所示。UbD 理论将教学目标分为三个相互关联的层次：（1）掌握知能——学生需要掌握的基础知识与基本技能；（2）理解意义——学生能够洞察核心概念的本质，建立知识间的联系；（3）学习迁移——学生能够在真实情境中独立运用所学知识解决新问题。三者形成完整的学习闭环，其中掌握知能是基础，理解意义是中介，学习迁移是最终目标。

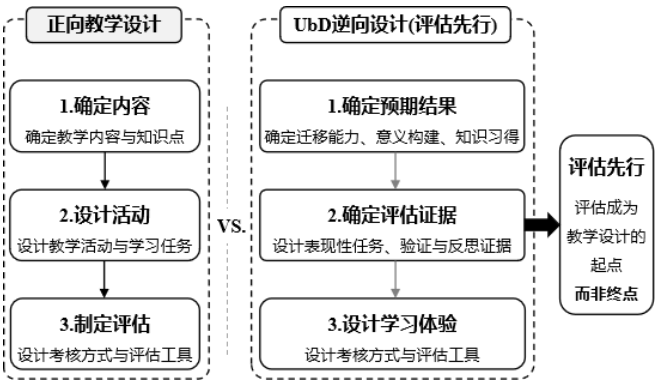


图 1 UbD 逆向设计三阶段与传统正向设计对比示意图

针对当前普通本科院校《编译原理》课程教学中存在的理论实践脱节、实践体系碎片化、教学评价体系滞后的问题，本文将 UbD 三阶段框架与《编译原理》课程的专业特性进行适配：第一阶段解决预期学习目标的设定问题，确立三层次目标体系；第二阶段解决评估证据的确定问题，构建“评估先行”的评价框架；第三阶段解决学习体验的组织问题，设计 W-H-E-O 精简框架和三阶递进实验体系。

2．2 第一阶段：确定预期学习结果

第一阶段核心问题是需要明确学生在完成课程学习后，能够依托真实工程场景独立开展哪些实践任务？本文依据UbD三层次目标结构，确立了如下预期学习结果：

（1）学习迁移：学生能够阅读和调试已有编译器（如TinyC、SysY）的关键模块代码，理解开源编译工具链的工作机制，并将编译原理知识迁移至后续专业课程及综合性实践项目（如毕业设计）中。该能力层次服务于UbD所倡导的“持久理解”，是普通本科院校学生毕业后最可能应用且最需具备的能力。

（2）理解意义：学生能够理解词法分析、语法分析、语义分析、中间代码生成、目标代码生成与优化等核心模块的基本概念与设计理念，能够分析简单编程语言的文法结构，阐明编译过程中各个阶段的主要任务与衔接关系。该层次关注对编译系统运作逻辑的理解，而非对孤立知识点的机械记忆。

（3）掌握知能：学生需要掌握正则表达式、有穷自动机、LL(1)分析、LR分析、中间代码生成及目标代码生成与优化等基础理论与经典算法，能够在给定框架下独立完成词法分析器、语法分析器与中间代码生成模块的实现。

在UbD框架中，学习迁移处于目标体系的顶层，

理解意义服务于迁移能力的达成，掌握知能则为更高层次的能力发展提供基础支撑。三个层次相互关联、逐级递进，共同构成课程预期学习目标的完整体系。

2．3 第二阶段：确定评估证据

UbD第二阶段的核心问题是需要确认哪些证据能够证明学生达成了预期学习结果？为此，本文构建了三类评估证据体系。

（1）表现性任务（占总评50%）。表现性任务是UbD评估体系的核心，重点考察学习迁移和理解意义，要求学生在真实或模拟情境中展示对知识的理解与应用能力。表现性任务由五个子项构成，各子项权重依据教学时长与内容复杂程度差异化设定（详见表1）。其中，词法分析模块演示需要展示词法分析器对源码中标识符、关键字等词法单元的识别能力；语法分析模块演示需要展示根据文法规则判断语句结构正确性并输出语法树或错误提示的能力；语义分析与中间代码生成模块演示需要展示类型检查、符号表管理及三地址码生成能力。此外，各小组还需提交完整编译器代码并进行限时5分钟的技术答辩，接受关于模块接口、算法选择及设计决策的提问；答辩结束后，各小组汇总各模块实验报告中的“反思与小结”，系统回顾编译器开发全过程中的关键困难、解决思路及团队协作情况，形成完整的课程反思总结报告。

表 1 评估证据及其与 UbD 目标层次的对应关系

评估类型	子项	权重	评估时间	对应 UbD 目标层次
表现性任务(占总评50%)	词法分析模块演示	20%	第 6 周	T
	语法分析模块演示	30%	第 12 周	T
	语义分析与中间代码生成演示	10%	第 14 周	T
	技术答辩	25%	第 15 周	T、M
	反思报告	15%	第 16 周	T、M
阶段性验证(占总评30%)	单元验证（在线评测）	30%	每章结束时	A
	实验报告	45%	每次实验后	A、M
	课堂研讨与参与	25%	持续	M
期末考核(占总评 20%)	期末试题	100%	期末考试	A、T

注：*目标代码生成与优化以理论讲授为主，不设独立演示模块，相关理解与迁移能力通过期末开放性试题进行综合考查。*

（2）阶段性验证（占总评30%）。阶段性验证贯穿教学全过程，涵盖掌握知能和理解意义，重点评估过程性学习成效。每章教学内容结束后通过在线平台实施阶段性单元验证（占阶段性验证的30%），采用自动评测方式（含理论选择题与代码填空题），即时生成反馈报告；实验报告（占阶段性验证的45%）要求学生记录实验过程、分析问题并提出改进方案；课堂研讨与参与（占阶段性验证的25%）贯穿整个学期，持续评估学生的课堂互动质量。

（3）期末考核（占总评20%）。期末考试以笔试的形式开展，侧重考查学生对编译原理核心概念的理解与应用能力，减少对机械记忆的依赖，对应掌握知能与学习迁移目标达成情况。

全部评估标准会在开课第一周向学生公布，以保障评估的透明性与过程性。全部评估证据及其与UbD目标层次的对应关系如表1所示。

2．4 第三阶段：设计学习体验

UbD第三阶段的核心问题是需要明确哪些学习活动与教学资源最能达成预期目标？为此，本文将WHERETO七要素框架精简改造为W-H-E-O四要素框架，以适应《编译原理》课程的教学实际。

（1）W（方向明确）：在课程第一周展示完整微型编译器的运行效果。具体操作：演示TinyC编译器处理一段包含语法错误的C代码时输出精准错误行号的问题，引导学生思考“编译器是如何实现该功能的？”帮助学生明确课程学习的整体目标与方向。

（2）H（兴趣激发）：分段设置探究式悬念。例如，在讲授语法分析内容之前，展示“a=1；1=a；”两条语句编译结果迥异的演示效果，引导学生思考“计算机如何区分这两种结构差异？”，以激发学生持续的探究兴趣。

（3）E（体验设计）：构建如图2所示的三阶递进实验体系。L1为基础验证层，引导学生使用或修改Lex/YACC等开源工具解析简单语言，消除“从零编写”的畏难情绪；L2为模块实现层，在半成品编译器框架下，要求学生独立实现词法分析器核心函数及部分语法分析器代码；L3为综合设计层，以3-4人为一组，协作完成一个简化子语言（如类Pascal或SysY子集）的完整编译器设计，并进行代码评审与功能演示。

（4）O（系统组织）：教学流程的时间序列安排。课程总学时为52学时（理论40学时，实验12学时）。第1周完成W导入，演示完整微型编译器运行效果；第2-6周讲授词法分析并安排L1基础实验，第6周末进行第一次单元验证并完成词法分析模块演示；第7-12周讲授语法分析并安排L2模块实现实验，第12周末进行第二次单元验证并完成语法分析模块演示；第13-14周讲授语义分析与中间代码生成，结合L3综合设计小组项目，第14周末进行第三次单元验证并完成语义分析与中间代码生成演示；第15-16周讲授目标代码生成与优化，并同步在第15周进行项目答辩与代码评审，在第16周提交反思报告并进行课程总结。

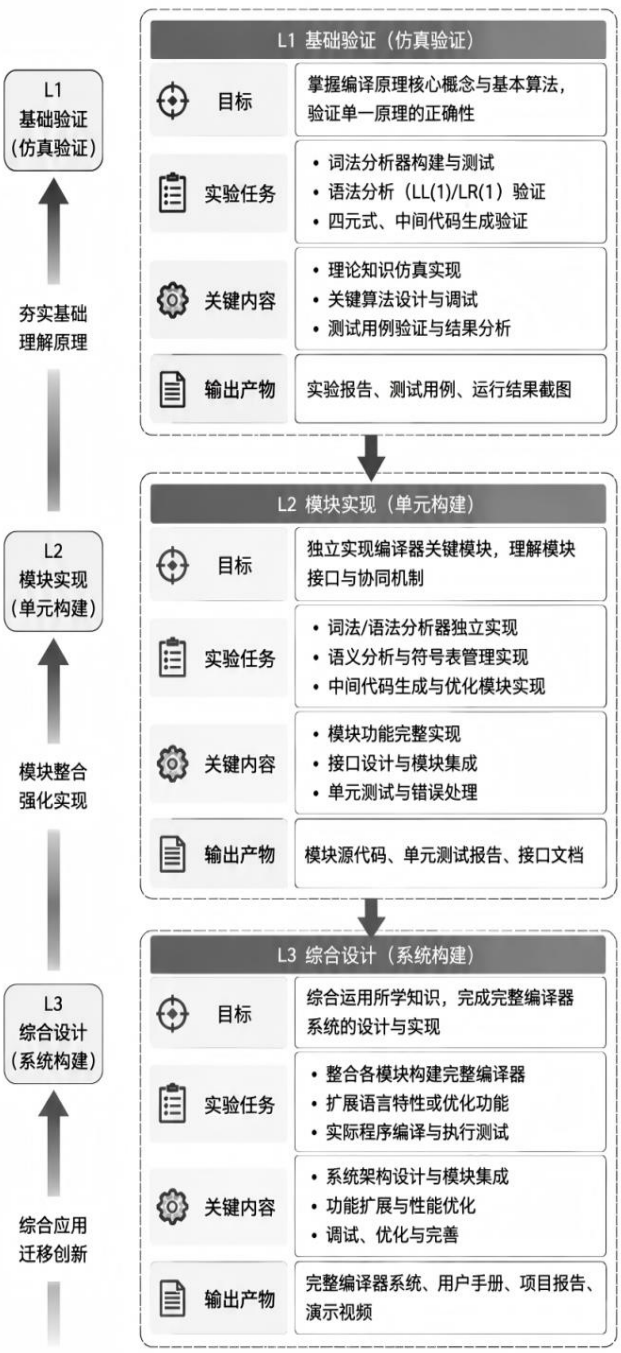


图 2　L1-L2-L3 三阶递进实验体系

3　课程思政融合设计

3．1　融合理念与主线

课程思政是落实立德树人根本任务的重要途径，其核心在于将价值引领有机融入专业教学全过程，而非脱离知识内容进行生硬灌输。针对引言所述课程思政流于表面的现象，本文以UbD理论所倡导的“大概念”和“持久理解”为导向，确立“编译技术自主化”为课程思政主线，将思政元素与编译各阶段的技术内涵深度融合，实现知识传授、能力培养与价值引领的有机统一，具体融合主线框架如图3所示。

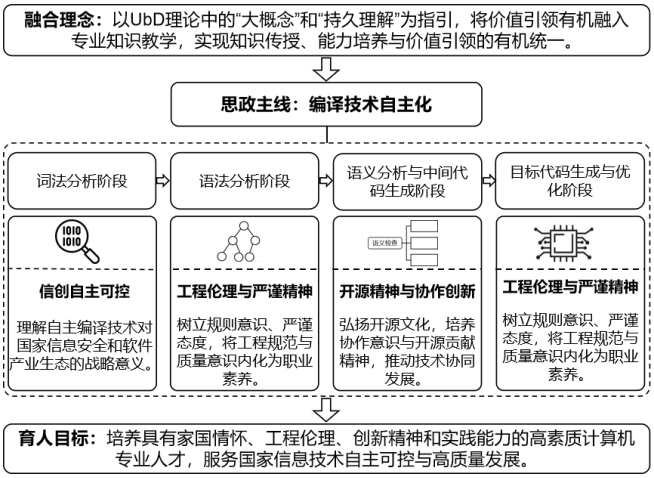


图 3　《编译原理》课程思政融合主线框架图

3．2　分阶段思政融合设计

本文根据《编译原理》课程教学进程，以“编译技术自主化”为主线，依托各知识模块实施分段式思政元素融入。各教学阶段严格遵循UbD逆向教学设计目标框架，按照掌握知能、理解意义、学习迁移的层级逻辑逐层递进开展教学，实现知识传授、能力培养与价值引领的深度融合、同向同行。各阶段思政元素与知识点的对应关系如表2所示。

表 2 编译原理知识点与思政元素融合映射

知识模块	知识点	思政主题	融入方式	对应 UbD 目标层次
词法分析	正则表达式、有穷自动机	信创自主可控	介绍华为毕昇方舟编译器及国产编程语言案例	A
语法分析	LL(1)分析、LR分析	工程伦理与严谨精神	分析编译器错误处理机制与代码质量保障的关系	M
语义分析与中间代码生成	属性文法、中间语言	开源精神与协作精神	分析 LLVM/GCC 开源社区案例及国内企业贡献	M、T
目标代码生成与优化	指令选择与寄存器分配、局部优化与全局优化	科技自立自强	介绍 LoongArch 配套编译工具链研发历程	T

（1）词法分析阶段：以“编译原理赋能信创产业发展”为切入点，结合课程内容，介绍华为毕昇编译器、方舟编译器在鸿蒙生态体系中的核心应用价值，梳理仓颉、凹语言等国产编程语言在编译器研发层面面临的技术瓶颈与攻关难点。帮助学生在了解国产编译技术发展现状的同时，引导其认识编译技术自主可控对国家信息安全与产业生态的战略意义，增强科技报国的责任感与使命感。

（2）语法分析阶段：以“规范严谨：筑牢代码质量与安全底线”主题，融入工程伦理与职业素养教育。语法分析在编译过程中具有承上启下的作用，承担着

语法正确性校验的任务，其核心准则是严格依据文法规则完成代码解析与校验，对所有格式错误、语法违规的代码精准识别并输出规范报错信息。这一过程中所体现的规则思维、严谨态度与零容错理念，与软件工程领域的代码规范、质量保障、安全审计等工程实践要求高度契合。通过将编译阶段的错误处理机制与工程实践中的质量意识相对照，引导学生将严谨规范的精神迁移到日常编程与项目开发中，培育精益求精、审慎负责的职业素养。

（3）语义分析与中间代码生成阶段：以“开源共享与协作创新”为育人主线，着力培育学生的团队协作能力与开源奉献精神。结合LLVM、GCC等主流开源编译器社区的发展案例，剖析开源协作模式对基础软件技术迭代与产业创新的推动作用；同时以华为深度布局LLVM、RISC-V开源生态为例，阐述国内科技企业在全球开源共建中的担当与贡献。引导学生将开源协作、共建共享的理念内化于心，未来主动参与开源实践、回馈技术社区。

（4）目标代码生成与优化阶段：以“科技自立自强”为育人落脚点，将编译优化技术教学与国家核心技术战略需求进行对接。通过介绍龙芯LoongArch等国产自主指令集架构配套编译工具链的研发攻坚历程，阐明编译技术在构建自主可控计算生态中的基础性、战略性地位。引导学生正视我国基础软件领域的短板与“卡脖子”难题，鼓励学生扎根基础软件方向、投身核心技术攻关，增强为自主可控计算生态建设贡献力量的责任意识。

4 实施效果与分析

4．1 教学实施情况概述

本教学改革方案在苏州科技大学计算机科学与技术专业2022级、2023级共计两届共4个教学班中实施，累计参与学生160人。对照班级为使用传统教学模式的2021级2个教学班，共计93人。两批学生同属普通本科批次招生，培养方案与课程体系基本一致，前置课程（程序设计基础、数据结构）成绩分布基本相当，具备可比性。教学改革实验涵盖两个完整教学周期，以保证数据的稳定性和可靠性。

4．2 学习效果对比分析

实验组与对照组在各考核维度上的对比结果如表3所示。数据表明，采用UbD三阶段框架后，学生在课程成绩、实验完成度、项目通过率、课堂参与度和评教满意度五个维度上均有明显提升。课程平均成绩从对照班的71.4分提升至实验班的74.7分，提高了3.3分；实验任务完成率从66.7%提升至87.5%，提高了20.8个百分点；项目通过率（指完成完整编译器设计并通过基本功能测试的学生比例）从57.3%提升至81.9%，提高了24.6个百分点；课堂参与率（统计课堂提问作答次数≥3次的学生比例）从48.7%提升至74.6%，提高了25.9个百分点；学生评教满意度（评价结果为“优”或“良”的比例）从73.6%提升至92.3%，提高了18.7个百分点。

表 3 实验组与对照组学习效果对比数据

考核维度	对照组（n=93）	实验组（n=160）	提升幅度
课程平均成绩(分)	71.4	74.7	+3.3分
实验任务完成率(%)	66.7	87.5	+20.8%
项目通过率（完整编译器）(%)	57.3	81.9	+24.6%
课堂参与率（提问≥3次）(%)	48.7	74.6	+25.9%
学生评教满意度（优/良率）(%)	73.6	92.3	+18.7%

***注：**“实验任务完成率”指学生按时提交各模块实验报告的比例，而“项目通过率”指完成完整编译器并通过基本功能测试的比例，二者分别反映过程性学习态度与综合性工程能力。*

4．3 学生反馈与质性分析

为进一步了解学生对UbD教学改革方案的接受程度和学习体验变化，课程结束后开展问卷调查，共发放问卷160份，回收有效问卷147份，回收率为91.9%。调查聚焦UbD教学改革的关键设计要素展开，包括目标导向设计、递进式实验体系、形成性评价机制以及项目化实践等内容。调查结果如图4所示。

从图4可以看出，学生对UbD教学改革各项措施均表现出了较高认可度，各项指标的认可率均超过 80%。其中，“三阶递进实验让我一步步建立了完整编译器概念”的认可率最高（86.4%），其次为“单元验证及时反馈帮助我发现知识盲点”（83.0%）。上述结果表明，学生普遍认可UbD框架下目标引领、过程评价和项目实践相结合的教学模式。

结合问卷反馈和课程反思报告开展定性归纳分析发现，学生反馈主要集中在学习目标认知、知识体系建构和工程实践能力提升三个方面。首先，完整编译器成果演示能够引导学生从全局视角把握课程目标与学习脉络，增强学习动机，降低理论知识学习难度。其次，三阶递进实验体系促进了词法分析、语法分析和语义分析等知识模块的逐步整合，助力学生建立编译系统的完整认知框架。最后，综合项目设计、技术答辩以及阶段性单元验证等环节强化了学习反馈与实践训练，使学生能够在真实任务情境中提升系统开发与工程实践能力。

综合来看，UbD框架通过“目标引领—过程评价—项目实践”的闭环设计，有效促进了学生从知识掌握向系统理解与综合应用的转变。该结果与课程成绩、

项目完成率和课堂参与度等量化数据表现出一致趋势，进一步验证了UbD教学改革在提升学习投入度、知识整合能力和工程实践能力方面的积极成效。

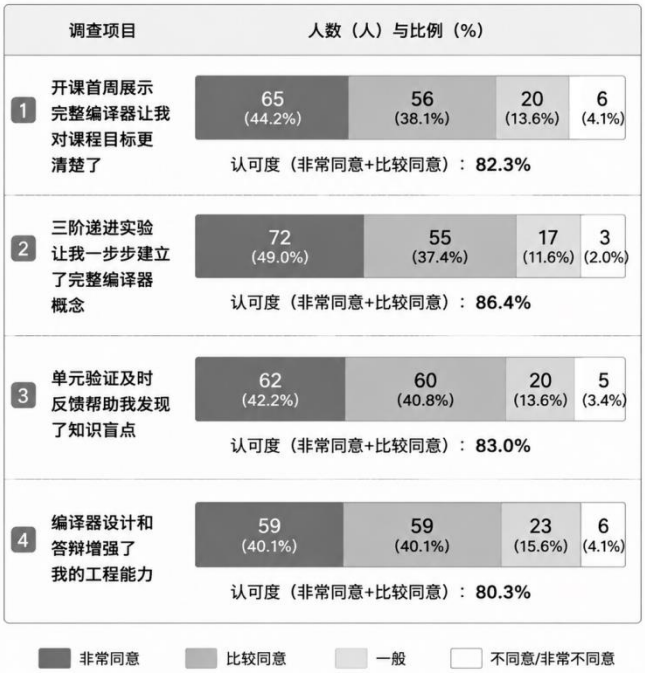


图 4 学生对 UbD 教学改革关键环节的认可度调查结果（N=147）

4．4 课程思政效果评价

为评估以“编译技术自主化”为主线的思政融合设计效果，本文从项目报告、期末考核和问卷调查三个维度进行评价。

在项目报告评价中，设置了“自主可控意识”评价维度（占技术报告成绩的10%），实验班平均得分为85.8分，表明学生能够较好地将专业知识与国产编译技术发展需求相结合。在期末考试试题中设置与信创产业和编译技术自主化相关的论述题（占期末成绩的20%），实验班平均得分率达到80.7%，多数学生能够结合课程内容进行较为深入的分析与阐述。学期末问卷调查结果显示（N=147，回收率91.9%），81.9%的学生认为“课程思政内容让我对国产编译技术有了更深入的了解”，15.4%的学生表示“基本认同”，仅有2.7%的学生表示“不认同”。

综合评价结果表明，以“编译技术自主化”为主线的课程思政设计有效促进了学生对国产基础软件和自主创新战略价值的理解（项目报告“自主可控意识”维度均分85.8分，期末论述题得分率80.7%），增强了科技自立自强意识（81.9%学生认同），实现了专业知识学习、工程能力培养与价值引领的有机统一。

5 结束语

针对普通本科院校《编译原理》课程教学中存在的理论抽象、实践碎片、评价滞后等核心问题，本文引入UbD逆向教学设计理论，构建了“预期学习目标—评估证据—学习体验”三阶段逆向教学框架，设计了三阶递进实验体系，并以“编译技术自主化”为主线实现课程思政的有机融合。教学实践表明，实验班课程平均成绩提高了3.3分，项目通过率提高了24.6个百分点，课堂参与率提高了25.9个百分点，81.9%的学生认为思政内容能够有助于加强对国产编译技术的理解。UbD“评估先行”的逻辑有效解决了教、学、评脱节问题，为普通本科院校《编译原理》课程提供了可复制的教学方案。本研究在实施过程中也反映出若干局限，如三阶递进实验体系要求学生具有较高的编程基础，导致部分学生在L3阶段面临较大挑战；同时，该教学框架对教师的课程设计能力和过程性评价执行能力也提出了更高要求。下一步将探索AI辅助教学工具融合、拓展UbD框架至其它计算机专业核心课程，以持续验证其跨课程适用性。

参 考 文 献

[1] 裴华艳,王焕民,兰丽.面向创新能力培养的“编译原理”课程教学改革研究[J].甘肃教育研究,2025,(16):8-11.

[2] 汪花梅,杨扬,李旭涛.基于框架驱动及阶梯迭代的编译原理实验教学[J].计算机教育,2025,(2):112-117.

[3] 许智宏,董永峰,王利琴,等.基于知识图谱的精细化成果导向编译原理课程[J].计算机教育,2025,(4):75-79.

[4] 王吴凡,王雅文,李文生.编译原理课程思政体系构建初探[J].计算机教育,2026,(2):58-61.

[5] 李建义,陈亭宇,孙红艳,等.OBE背景下新建应用型本科院校编译原理课程“四化”教学改革实践[J].北华航天工业学院学报,2024,34(2):21-23.

[6] 郭春丽,林伟伟,谢明玲.基于OBE理念的课堂教学形成性评价研究与实践[J].计算机技术与教育学报,2024,12(4):8-14.

[7] 李婷,刘顺财.基于产教深度融合的AIoT应用型人才培养模式研究[J].计算机教育,2025,(3):160-165.

[8] 修佳鹏,王莉,陈虹雨,等.产教融合的特色化示范性软件学院实践创新体系探索与实践[J].计算机技术与教育学报,2023,11(2):75-80.

[9] 徐伟,刘硕,张昱.基于LLVM驱动程序的编译原理实践教学改革[J].软件导刊,2023,22(6):191-195.

[10] Wiggins G, McTighe J. Understanding by design (2e) [M]. Alexandria, VA: Association for Supervision and Curriculum Development (ASCD), 2005.

[11]魏妮妮,李丹.基于UbD模式的“C语言程序设计”课程教学改革实践——面向物联网工程应用能力培养[J].物联网技术,2026,16(1):150-155.