

基于容器化技术的人工智能系统实验教学体系构建与实践*

王周恺^{1,2}, 刘小雪^{1,2}, 王一川^{1,2}

(1. 西安理工大学 计算机科学与工程学院, 陕西 西安 710048;

2. 西安理工大学 陕西省网络计算与安全技术重点实验室, 陕西 西安 710048)

摘要 针对人工智能专业本科实验教学中普遍存在的重算法轻系统结构性失衡问题, 分析了学生在环境部署、算力调度与系统优化三方面工程实践能力培养不足的深层原因, 提出并构建了基于 Docker 容器化技术的人工智能高性能计算递进式实验教学体系, 阐述了遵循认知、理解、优化、应用四级递进逻辑的 5 个实验项目设计方法。依托西安理工大学人工智能专业 60 名学生的教学实践验证: 容器化部署将环境准备耗时由 135 分钟压缩至 8 分钟, 前后测知识掌握度提升 41.7%, 实验完成率达 90%, 为新工科背景下培养人工智能系统工程人才提供了可推广的课程改革方案。

关键词 人工智能, 实验教学, 容器化技术, 工程能力培养, 并行计算

Construction and Practice of an Experimental Teaching Framework for Artificial Intelligence and High-Performance Computing Based on Containerization Technology*

WANG Zhoukai^{1,2}, LIU Xiaoxue^{1,2}, WANG Yichuan^{1,2}

(1. School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China;

2. Shaanxi Key Laboratory of Network Computing and Security Technology, Xi'an University of Technology, Xi'an 710048, China)

Abstract—Undergraduate experimental teaching in artificial intelligence programs at Chinese universities is characterized by a structural imbalance, in which algorithm design and model training receive disproportionate emphasis at the expense of system-level engineering practice. Students tend to graduate with insufficient competence in environment deployment, computing resource scheduling, and system performance optimization - a gap that substantially limits their readiness for industrial AI engineering roles. This paper analyzes the underlying causes of this imbalance and presents a containerized experimental teaching framework for artificial intelligence and high-performance computing education. The framework is grounded in a four-stage pedagogical progression spanning cognitive awareness, mechanistic understanding, performance optimization, and engineering application, and is implemented through Docker containerization with direct GPU passthrough via the NVIDIA Container Toolkit. Five progressive laboratory projects are developed within this framework, addressing AI system profiling, data pipeline optimization, mixed-precision training, distributed data-parallel training, and parameter-efficient fine-tuning of large language models. A teaching experiment conducted with 60 undergraduate students at Xi'an University of Technology shows that containerized deployment reduced environment preparation time from 135 minutes to 8 minutes with a deployment success rate of 100%, the laboratory completion rate reached 90.0%, and pre- and post-test scores improved from 54.2 to 76.8 points, representing a 41.7% gain in system-level knowledge mastery. These results suggest that the proposed framework effectively addresses the persistent deficiencies in artificial intelligence experimental teaching and provides a transferable model for curriculum reform oriented toward system engineering competency under the initiative of emerging engineering education.

Keywords—artificial intelligence, experimental teaching, containerization technology, engineering competency cultivation, parallel computing

1 引言

近年来, 生成式人工智能与大规模预训练模型的快速发展, 使得算力系统的高效调度与性能优化成为

人工智能工程实践的核心能力之一。教育部新工科建设相关文件明确指出, 计算机类专业人才培养须从单一的算法设计能力向系统工程综合能力拓展, 以满足产业对复合型 AI 工程人才的需求^[1]。在此背景下, 面向人工智能专业的本科实验教学亟需从以算法实现为中心转向以系统工程能力培养为目标, 使学生不仅能

***基金资助:** 本文得到西安理工大学实验技术开发基金项目《面向新工科的“AI+高性能计算”实验教学体系构建》(SJK202514)的支持。

***通讯作者:** 王周恺 zkwang@xaut.edu.cn。

够设计和训练模型，更能理解并优化其在真实算力环境中的运行行为。

然而，当前高校人工智能专业本科实验教学在转型过程中面临三方面困难。首先是实验内容缺失，现有人工智能本科专业课程的实验普遍聚焦于神经网络结构与超参数调整，对系统层知识涉及不足，导致学生缺乏底层优化能力^[2]。此外，由于实验环境部署繁琐，GPU 计算环境依赖复杂，软硬件耦合性强，环境冲突引发的配置错误易占据大量课时，导致教学时间被压缩^[3]。最后，受制于并行计算知识的缺乏，学生易将多卡工作站作为单卡设备使用，已有硬件投入未能转化为相应的教学效益^[4]。

针对上述困难，主流研究方法主要聚焦于容器化技术应用、人工智能实验平台建设与实验内容设计三方面。例如，李慧春等将 Docker 技术引入计算机基础实验教学，实现实验环境的标准化分发与快速部署^[5]。但未针对 GPU 设备的容器化透传与异构算力调度，难以直接适用于深度学习实验场景。韩磊等构建了基于服务器集群的大数据实训平台^[6]，但对深度学习框架支持不足，未能覆盖 GPU 性能分析与优化的教学需求。许方亮等提出了学练测一体化方法^[7]，但仍缺乏面向 AI 系统性能优化的系统性实践训练框架。

综上所述，现有工作或聚焦于通用环境的容器化部署，或仅限于平台层面的资源管理，尚未形成将标准化异构算力环境与系统性工程能力培养相结合的完整实验教学体系^[8,9]。针对上述不足，本文工作如下：首先，基于 Docker 容器化技术与 NVIDIA 容器运行时工具包，构建支持 GPU 直通的异构并行计算实验平台，实现深度学习软件栈的标准化封装与快速部署；其次，设计覆盖性能监测、输入输出优化、显存管理、分布式训练与参数高效微调五个层次的递进式实验项目，构建面向人工智能系统工程能力的完整培养路径。

2 实验教学体系设计

针对现存教学困难，本实验体系由两个相互支撑的部分组成。首先是基于容器化技术构建的实验平台，该平台从技术层面解决环境部署与算力调度问题，为人工智能相关实验提供标准化、可复现的运行基础；在此基础上，设计递进式实验项目体系，并围绕人工智能系统工程能力的培养目标，构建覆盖性能监测、资源优化、分布式训练与参数微调的实验序列。

2.1 基于容器化的实验平台架构设计

实验平台采用三层架构设计，如图 1 所示，由物理硬件层、容器运行层与应用交互层依次构成，各层分别承担算力供给、环境隔离与交互接入的功能。

物理硬件层依托陕西省网络计算与安全技术重点实验室的 GPU 工作站群构建，每台节点配备 RTX 3090

显卡与 NVMe 固态硬盘，分别满足算力供给与高速数据读取的需求。容器运行层负责解决异构算力的隔离与软件依赖繁杂的问题，基于官方 PyTorch 镜像构建，集成统一计算设备架构 CUDA，并通过 NVIDIA 容器运行时工具包实现对宿主机 GPU 的直接访问。实现对宿主机 GPU 的直接访问，在保持容器隔离性的同时尽可能地降低硬件性能损耗。应用交互层以 JupyterLab 作为统一的交互式开发环境，学生通过 Web 浏览器完成代码编写、终端操作与文件管理，无需在本地安装额外软件。交互层中预装的 nvdashboard 插件提供 GPU 利用率与显存占用的实时可视化监控，使学生在实验过程中能够直观观察计算资源的动态变化。

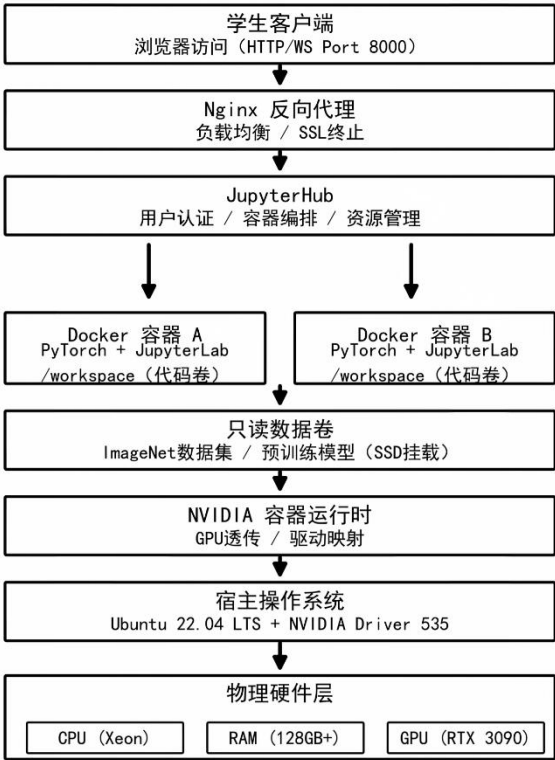


图 1 基于容器化的实验平台架构

在部署流程方面，平台首先通过 Dockerfile 定义完整的环境规格，将 JupyterLab、Nsight Systems 等工具集成于基础镜像之中，构建完成后经由本地 Registry 统一分发至各实验节点。学生使用时只需执行启动脚本，系统即自动完成容器创建、GPU 设备挂载、数据集挂载及端口转发配置，无需人工干预。

2.2 进阶式实验教学内容设计

传统人工智能专业实验以模型训练收敛、获得预期准确率为目标，对训练过程中计算资源的实际利用情况关注较少。学生因而容易将 GPU 视为只需调用即可得到结果的黑盒，对算力在何处被消耗、又有多少未被有效利用缺乏认识^[10]。本课程的实验设计即以此为切入点，围绕一次真实训练任务在系统中的执行路径，将数据读取、显存占用与多卡通信等典型的性

能受限环节逐一剖析,引导学生先建立观测能力,再实施针对性优化。

基于上述思路,五个实验被组织为一条由浅入深的能力阶梯,如图2所示。实验一着力解决可观测性问题,借助性能分析工具使学生了解训练时间在各执行阶段的分布,建立系统层面的观测视角;实验二至实验四分别针对数据 IO、显存与通信三类典型瓶颈,每个实验均围绕一个具体的性能制约因素设计对比方案,使学生在量化对比中理解相关技术的作用机理及其代价;实验五为综合应用环节,将前序实验形成的“观测—权衡—优化”方法迁移至大模型微调这一前沿场景。其整体遵循“认知—优化—应用”的递进逻辑,即由识别瓶颈到消除瓶颈,再到在资源受限条件下进行合理取舍。

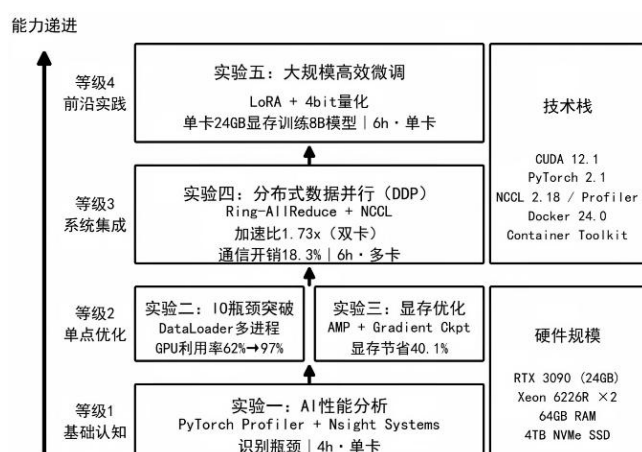


图2 基于能力递进的 AI 系统实验教学内容设计

五个实验并非相互独立的技术演示,而是共享同一容器化环境与同一份基准代码。学生每完成一项优化,均可在一致的度量口径下与前一版本结果进行对照。该设计在减少无关变量干扰的同时,也使学习效果具备可度量、可复盘的特征。

3 实验教学验证与评估

为了验证实验教学体系能否在真实教学场景中达到预期效果,设计评估维度如下:在技术性能维度,通过考察容器化平台的部署效率及各实验项目的性能优化效果,确认平台是否存在制约教学开展的技术瓶颈;在教学应用维度,通过考察学生的知识掌握变化与主观体验,评估体系对人才培养目标的实际贡献。两个维度相辅相成,前者为后者提供可靠的技术前提,后者为前者赋予教育学意义。

教学对象选取西安理工大学计算机科学与工程学院2023级人工智能专业两个教学班60名学生,依托专业课《人工智能系统实践》,从平台性能与教学效果两个维度展开对实验教学体系的评测。每个实验项目安排2学时讲解技术原理与实验要求,2至4学时上机课由学生独立完成实验任务,最后组织1学时讨论

课对实验结果与问题进行总结。实验教学在 Dell PowerEdge 工作站上进行,单节点配置为2路 Intel Xeon Gold 6226R 处理器、64GB 内存、2块 RTX 3090 显卡及4TB NVMe 固态硬盘,运行 Ubuntu 22.04 操作系统及 Docker 24.0.5 容器引擎。

3.1 平台部署效率验证

开展实验前需要首先测试环境准备效率和 GPU 性能开销,教学团队设计对比实验测试传统手动安装与容器部署两种方案的用时。其中传统方案要求学生依次安装 CUDA Toolkit、cuDNN、Miniconda、PyTorch 及依赖库。容器化方案采用预构建镜像,学生执行约10行 Shell 命令完成服务启动,从本地 Registry 拉取镜像,待镜像缓存后,再初始化容器并启动 JupyterLab 服务。两种方式的耗时对比结果如表1所示。

表1 不同部署模式下的环境准备时间对比

部署模式	基础环境配置耗时	依赖包安装耗时	排错耗时	总耗时	部署成功率
传统裸机模式	45min	30min	60min	135min	88.3%
容器化模式(首次)	8min	0min	0min	8min	100%
容器化模式(热启动)	<1min	0min	0min	<1min	100%

由表1可知,容器化方案将部署耗时从135分钟压缩至8分钟,部署成功率从88.3%提升至100%,证明预编译镜像分发能够有效消除依赖冲突风险。

3.2 实验项目性能优化效果验证

实验一考察容器化环境对 GPU 计算性能的影响。学生在裸机与容器环境中分别运行 ResNet-50 训练任务,使用 ImageNet-1k 数据集的10%子集, batch size 设为128,训练10个 epoch,每组测试重复5次取均值,最终实验结果如表2所示。容器环境的单 epoch 训练耗时为318秒,较裸机的312秒仅高出1.9%, GPU 平均利用率为94.8%,与裸机利用率相近,均满足高性能计算教学的精度要求。

实验二聚焦于数据输入输出瓶颈的量化与优化。学生在机械硬盘、网络文件系统与 NVMe 固态硬盘三种存储方案下,测量1000个 batch 的数据加载耗时与 GPU 等待时间,结果如图3所示。三种方案中,机械硬盘的 GPU 平均利用率为62.3%,输入输出瓶颈显著;网络文件系统受网络延迟制约,利用率为75.8%,改善有限; NVMe 固态硬盘方案将加载时间从186ms/batch 压缩至23ms/batch, GPU 利用率稳定在94.7%,瓶颈基本消除。

表 2 容器化GPU性能开销测试

环境类型	Epoch 时间(s)	GPU 利用率(%)	性能损耗(%)	核心算子耗时(ms)
裸机	312±3.2	96.2	-	Conv2d: 128.4
容器	318±3.5	94.8	1.9	Conv2d: 130.6

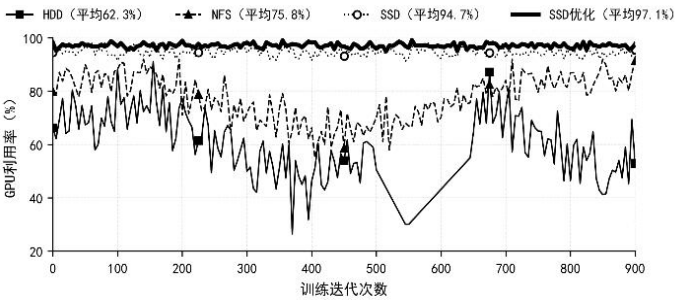


图 3 不同存储方案下的 GPU 利用率变化曲线

实验三对三种显存优化方案进行横向比较，分别为 FP32 全精度训练、自动混合精度训练以及自动混合精度结合梯度检查点训练，结果如表 3 所示。FP32 模式下，batch size 为 128 时峰值显存占用 17.2GB，扩大至 256 时显存溢出；切换至自动混合精度后，显存占用降至 10.3GB，优化率达 40.1%，batch size 256 正常运行，且训练速度加快 4.5%。梯度检查点的引入将显存进一步压缩至 7.8GB，优化率达 54.7%，但是训练速度下降 12.5%。

表 3 显存优化技术效果对比

优化方案	Batch Size	峰值显存 (GB)	优化率 (%)	Epoch 时间(s)	速度变化 (%)
FP32 基准	128	17.2	-	312	-
FP32 基准	256	OOM	-	-	-
AMP	128	10.3	40.1	298	-4.5
AMP	256	18.6	-	285	-8.7
AMP+检查点	128	7.8	54.7	351	+12.5
AMP+检查点	256	13.2	-	334	+7.1

实验四考察分布式数据并行训练的加速效果与通信开销。以 ResNet-50 在 ImageNet-1k 上的训练为基准，统一配置 batch size 为 256，训练 5 个 epoch，对比单卡、双卡与四卡三种配置下的训练时间，结果如图 4 所示。双卡训练实现了 1.73 倍加速，四卡为 3.12 倍，均低于理论线性加速比，差距主要来自 Ring-AllReduce 梯度同步算法在多卡间传输数据所产生的通信延迟。将梯度桶容量参数从 25MB 调整至 50MB 后，四卡通信开销降至 21.2%，加速比相应提升至 3.28 倍。

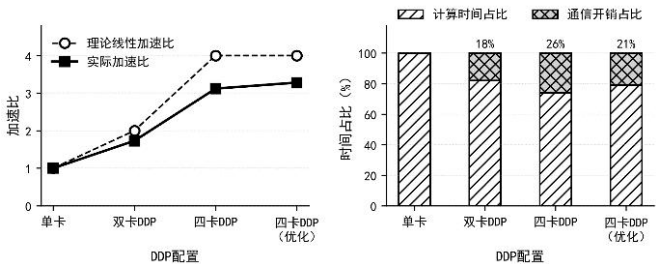


图 4 多卡并行加速比对比与通信开销占比分析

实验五在单张 RTX 3090 显卡上，对比全参数微调与低秩适配微调在 Llama-3-8B 量化模型上的显存占用与模型效果，微调任务为基于 Alpaca 数据集 5 万条样本的指令跟随训练，结果如图 5 所示。全参数微调在 batch size 为 1 时显存占用已达 23.8GB，接近显卡硬件上限，batch size 扩大至 2 时显存溢出；低秩适配微调在 batch size 为 8 时显存为 14.6GB，降幅达 38.7%，硬件门槛降低。模型效果方面，两种方案的 GPT-4 评分相差 0.08 分，表明低秩适配在降低硬件需求的同时，能够保持与全参数微调相近的模型质量。

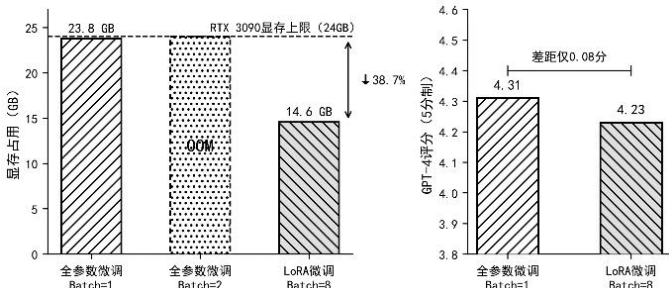


图 5 LoRA 微调与全参数微调的显存占用与模型效果对比

3.3 教学效果评估

实验完成后的教学效果评估面向全体学生开展，从 60 名学生中抽取 30 人进行前后测与问卷调查。27 名学生 (90.0%) 按时完成全部实验，成绩平均分 82.3 分。前测涵盖 GPU 工作原理、性能分析、并行计算、系统优化四个模块，平均得分 54.2 分，标准差 18.6 分，如图 6 所示。

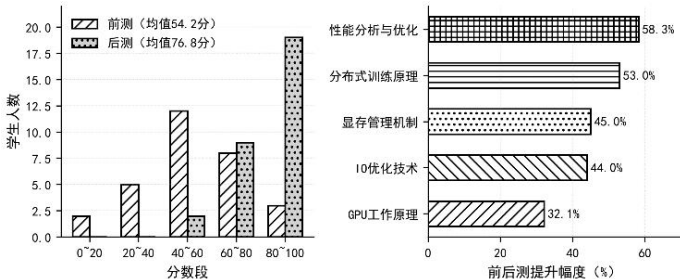


图 6 前后测成绩分布对比与分知识点提升幅度

后测在前测基础上增加应用型题目，平均得分 76.8 分，标准差 12.3 分，相较前测提升 41.7%，标准差下降 33.9%，表明体系在整体提升知识水平的同时缩小了学生间差距。性能分析与优化、分布式训练原

理两模块提升幅度最大, GPU 工作原理模块提升相对较小。

4 结束语

实验教学改革的难点从来不只是技术选型, 而在于如何在有限的课时与硬件条件下, 让学生真正建立起对底层系统的体系化认识。本文的实践表明, 将环境标准化与系统工程导向的实验内容相结合, 是解决这一问题的有效途径之一, 且这一途径对硬件投入的要求不高, 具备在同类高校推广的现实基础。目前的工作仍有若干未竟之处。验证样本局限于单一高校, 结论的普适性有待多校实践的检验。随着云端算力的普及, 如何将本体系迁移至云端环境、进一步降低硬件门槛, 同样是一个值得探索的开放性问题。

参考文献

- [1] 陈君, 肖媛, 王峰萍. 面向新工科的计算机专业创新人才培养的探索与研究[J]. 教育教学论坛, 2025(17): 153-156
- [2] 刘雪峰, 陈玮, 韩旭, 等. 大模型时代计算机实验教学的挑战与机遇[J]. 实验技术与管理, 2025, 42(8): 36-44
- [3] 娄久, 刘宏伟, 袁永峰, 等. 面向未来技术人才培养的计算机实验教学体系建设与探索[J]. 计算机教育, 2025(7): 86-90
- [4] 冯健昭, 肖克辉. AI 大模型时代的物联网课程教学改革探索[J]. 物联网技术, 2025, 15(21): 155-158
- [5] 李慧春, 梁楠, 黄伟, 等. 基于 Docker 的计算机基础实验在线环境建设[J]. 吉林大学学报(信息科学版), 2024, 42(4): 754-759
- [6] 韩磊, 施展, 施炎峰. 基于服务器集群的大数据实训平台建设[J]. 实验科学与技术, 2023, 21(4): 124-131
- [7] 许方亮, 解培岱, 王勇军, 等. "学、练、测"一体的软件安全课程实验动态生成方法[J]. 实验技术与管理, 2025, 42(11): 217-222
- [8] 韦南, 殷丽华, 宁洪, 等. 本科"机器学习"课程教学改革初探[J]. 网络与信息安全学报, 2022, 8(4): 182-189
- [9] 耿海军, 王威, 孟卓. 面向新工科的网络安全实践教学体系与实践平台构建研究[J]. 创新教育研究, 2023, 11(1): 65-71
- [10] 何扬帆, 高建华, 黄文斌. 面向计算思维培养的《人工智能与认知方法论》课程设计[J]. 计算机技术与教育学报, 2025, 8(4): 110-114