CCF大模型能力认证大纲

CCF Large Model Competence Certification

中國計算機學會

CHINA COMPUTER FEDERATION

CCF大模型能力认证大纲

CCF Large Model Competence Certification

中國計算機學會

CHINA COMPUTER FEDERATION

大模型能力大纲

近年来，大模型[1]（Large Models）以其强大的生成、理解和推理能力，成为人工智能领域的核心突破之一。从 GPT 系列到 DeepSeek 系列等，大模型不仅在自然语言处理任务中表现出色，还在代码生成、科学推理、多模态交互等领域展现了巨大的潜力。大模型的核心在于其基于 Transformer 架构的设计，通过海量数据的预训练和精细的微调，实现了对人类语言的高度理解和生成能力。然而，大模型的复杂性和多样性也带来了技术挑战，包括模型架构设计、训练优化、对齐与安全等问题。因此，深入理解大模型的原理、技术及应用，成为人工智能从业者和研究者的必备技能。

随着大模型技术的快速发展，行业对相关人才的需求急剧增加。然而，大模型技术的复杂性和跨学科性使得其学习和掌握门槛较高。为了帮助从业者系统性地掌握大模型的核心知识，并验证其能力水平，能力认证成为一种重要的评估手段。通过能力认证，不仅可以证明个人在大模型领域的专业能力，还能为企业和研究机构提供人才选拔的参考依据。此外，能力认证还能推动大模型技术的标准化和规范化，促进技术的健康发展。

本大纲以“大模型能力认证”为目标，系统性地组织了大模型的核心知识点，涵盖从基础概念到高级技术的全方位内容。大纲分为多个模块，包括大模型基础概念、模型架构、预训练技术、指令微调、人类对齐、解码与部署、提示学习、复杂推理、智能体、模型评测等。每个模块按照知识点的难度分级（用【】标注），并明确了考核方式（如概念理解、公式推导、代码实现等）。通过这种结构化的组织方式，学习者可以循序渐进地掌握大模型的核心技术，并为能力认证做好充分准备。

本大纲适用于大模型技术的初学者、从业者以及研究者。对于初学者，建议从基础概念和模型架构入手，逐步深入预训练、微调等高级技术；对于从业者，可以根据自身需求选择特定模块进行深入学习，如指令微调、人类对齐等；对于研究者，可以重点关注前沿技术模块，如智能体、复杂推理等。在学习过程中，建议结合实践项目或开源代码，通过动手实践加深对知识点的理解。此外，学习者可以通过考核方式自测学习效果，查漏补缺，确保全面掌握大模型的核心能力。

总之，本大纲旨在为大模型学习者提供一条清晰的学习路径，帮助其系统性地掌握大模型技术，并为能力认证奠定坚实基础。

[1] 目前阶段，大模型能力认证主要聚焦于大语言模型能力的认证。

CONTENTS 目录

一 . 人工智能基础概念 1

二 . 大模型基础概念 2

三 . 模型架构 4

四 . 预训练技术 6

五 . 指令微调 10

六 . 人类对齐 12

七 . 解码与部署 13

八 . 提示学习 15

九 . 复杂推理 17

十 . 智能体 18

十一 . 模型评测 20

十二 . 模型伦理与安全 23

中小学组知识点表 24

成人组知识点表 27

LMCC 第一轮样题 37

一．人工智能基础概念

●【1】人工智能相关概念定义

○ 知识点：智能、人工智能、机器学习、模型、训练等重要概念的定义

○ 扩展知识点：有监督（如分类、回归）、无监督（如聚类）等概念

○ 考核方式：概念

●【1】机器学习流程及经典模型

○ 知识点：数据预处理、模型训练、验证、测试、神经网络

○ 扩展知识点：数据清洗、数据变换、数据工程等重要环节的概念

○ 考核方式：概念

●【1】验证及评测

○ 知识点：交叉验证，精确率（Precision）、召回率（Recall）、受试者工作特征曲线（ROC）等重要评测指标

○ 扩展知识点：过拟合等概念

○ 考核方式：概念

●【1】人工智能常见应用领域

○ 知识点：自然语言理解、计算机视觉等

○ 扩展知识点：智能检索、推荐系统等

○ 考核方式：概念

二．大模型基础概念

●【1】自然语言的基础概念

○ 知识点：语言的定义、语言的特点、语言的应用

○ 扩展知识点：与其他信息（如图像，声音，代码等）的区别

○ 考核方式：概念、简答题

●【1】基本定义

○ 知识点: 大语言模型的定义、核心范式（生成、理解）

○ 扩展知识点：与传统NLP模型的区别、大模型的优势和局限性

○ 考核方式：概念、简答题

●【1】发展历程与现状

○ 知识点：四代语言模型的发展历程（统计语言模型、神经网络语言模型、预训练语言模型、大语言模型）

○ 扩展知识点：关键里程碑（如GPT系列、DeepSeek系列等）

○ 考核方式：概念、简答题

●【2】扩展法则

○ 知识点：KM扩展法则、Chinchilla扩展法则

○ 扩展知识点：计算最优模型规模、数据规模与计算资源的权衡

○ 考核方式：概念、计算题

●【2】大模型代表能力

○ 知识点：上下文学习、指令微调、逐步推理等代表性涌现能力

○ 扩展知识点：涌现能力的合理解释

○ 考核方式：概念、案例分析、API调用

三. 模型架构

●【1】注意力机制

○ 知识点：查询（query）-键（key）-值（value），softmax函数

○ 扩展知识点：自注意力机制、交叉注意力机制

○ 考核方式：概念、计算

●【1】主流架构

○ 知识点：编码器-解码器架构、因果解码器架构、前缀解码器架构

○ 扩展知识点：架构选择与任务类型的关系（如生成任务 vs. 理解任务）

○ 考核方式：概念、架构对比分析

●【1】Transformer模型的基本结构组成

○ 知识点：输入编码、位置编码、多头自注意力机制、前馈网络层、编码器、解码器

○ 扩展知识点：位置编码的变体（如相对位置编码、旋转位置编码）

○ 考核方式：概念、公式、代码实现

●【2】Transformer模型的详细配置

○ 知识点：归一化方法（LayerNorm、RMSNorm）、归一化模块位置、激活函数（GELU、Swish）、注意力机制

○ 扩展知识点：不同配置对模型性能的影响、大模型可解释性分析方法（探针技术、注意力可视化、特征重要性分析）

○ 考核方式：概念、公式、实现与分析

●【3】混合专家模型（MoE）

○ 知识点：稀疏激活、路由机制、负载均衡

○ 扩展知识点：MoE的显存优化与通信开销

○ 考核方式：概念、公式、实现

●【3】长上下文模型

○ 知识点：位置编码拓展（如ALiBi）、长上下文窗口拓展、长上下文数据构建

○ 扩展知识点：长上下文模型的显存与计算效率优化、长上下文模型的训练方法

○ 考核方式：概念、方法、实现

●【3】高效注意力架构

○ 知识点：稀疏注意力、线性注意力、键值缓存优化、多查询注意力（MQA）、分组查询注意力（GQA）、多头潜在注意力（MLA）

○ 扩展知识点：高效注意力机制的理论基础（如低秩近似）

○ 考核方式：概念、公式、实现

四．预训练技术

●【1】回归分析

○ 知识点：极大似然学习

○ 扩展知识点：多目标分类学习任务

○ 考核方式：概念、公式

●【1】自监督学习

○ 知识点：对比学习、掩码预测、自回归

○ 扩展知识点：自然语言与图像模态自监督学习的区别

○ 考核方式：概念、公式、简答

●【1】监督学习

○ 知识点：监督学习、无监督学习

○ 扩展知识点：监督学习与自监督学习的区别与联系

○ 考核方式：概念

●【1】预训练任务

○ 知识点：下一个词元预测（语言建模）、去噪自编码（如BERT的MLM）

○ 扩展知识点：下N个词元预测、预训练与多任务学习的关系

○ 考核方式：概念、公式、实现

●【1】优化设置：基于批次数据的训练方法

○ 知识点：Batch及Batch Size概念、Batch Size设置对模型训练的影响

○ 扩展知识点：动态批次大小调整策略

○ 考核方式：概念、实现

●【2】优化设置：学习率

○ 知识点：学习率概念、学习率衰减（退火）、学习率预热（Warm-up）

○ 扩展知识点：自适应学习率方法（如Cosine衰减）

○ 考核方式：概念、实现

● **【2】优化设置：优化器**

○ 知识点：Adam、SGD等

○ 扩展知识点：优化器的改进版本（如AdamW、LAMB）

○ 考核方式：概念、代码实现

● **【2】参数量计算**

○ 知识点：计算模型参数量的方法

○ 扩展知识点：MoE模型的参数量计算

○ 考核方式：计算题

● **【3】训练运算量及时间计算**

○ 知识点：训练运算量及时间复杂度的计算方法

○ 扩展知识点：FLOPs与训练时间的实际估算

○ 考核方式：计算题

● **【3】训练显存计算**

○ 知识点：预训练显存计算方法

○ 扩展知识点：显存优化技术（如梯度检查点）

○ 考核方式：计算题

● **【3】稳定优化技术**

○ 知识点：梯度裁剪、训练恢复、权重衰减等

○ 扩展知识点：混合精度训练中的稳定性问题

○ 考核方式：概念

● **【3】并行训练**

○ 知识点：3D并行（数据并行、流水线并行、张量并行）

○ 扩展知识点：ZeRO优化器与混合并行策略

○ 考核方式：概念

● **【3】训练精度**

○ 知识点：单精度、半精度、混合精度概念，及其对模型训练的影响

○ 扩展知识点：BF16与FP16的对比

○ 考核方式：概念

● **【3】高效训练技术**

○ 知识点：完全分片数据并行、完全和选择性激活重

计算、混合精度训练的流程、融合算子的原理

○ 扩展知识点：分布式训练中的通信优化

○ 考核方式：概念、计算

五．指令微调

●【1】模型解码

○ 知识点：概率分布、采样

○ 扩展知识点：给定分布下的概率采样

○ 考核方式：概念

●【1】指令微调

○ 知识点：指令数据、指令遵循、指令微调的概念

○ 扩展知识点：指令微调与多任务学习的关系

○ 考核方式：概念

●【1】指令数据集的构建

○ 知识点：指令数据合成方法

○ 扩展知识点：高质量指令数据的筛选与清洗

○ 考核方式：概念、实现

●【2】指令数据构建提升方法

○ 知识点：指令进化算法、自引导指令增强（Bootstrapping）、长上下文指令构建

○ 扩展知识点：指令数据质量与泛化能力的关系

○ 考核方式：概念、实现

●【2】微调优化设置

○ 知识点：目标函数、批次大小、多指令合并高效训练、多阶段混合训练（长短指令、数据课程）

○ 扩展知识点：指令微调中的资源消耗估算方法

○ 考核方式：概念、实现

●【2】数据组织策略

○ 知识点：领域专家模型的指令过滤、代理模型引导的指令配比、基于导数的指令数据选择

○ 扩展知识点：垂直领域的指令数据构造与使用

○ 考核方式：概念、实现

●【3】参数高效微调方法

○ 知识点：低秩适配（LoRA）、适配器微调、前缀微调、提示微调

○ 扩展知识点：参数高效微调的理论基础

○ 考核方式：概念、公式、实现

六．人类对齐

●【1】人类对齐的背景与标准

○ 知识点：人类对齐背景、人类对齐标准（如无害性、有用性、诚实性）

○ 扩展知识点: 其他对齐标准（语言表达、道德标准等）

○ 考核方式：概念

●【2】人类偏好与反馈数据收集

○ 知识点：人类反馈收集方法、基于评分的人类反馈、基于排序的人类反馈

○ 扩展知识点：反馈数据的偏差与修正

○ 考核方式：概念

●【2】非强化学习训练的对齐方法

○ 知识点：DPO的公式及原理

○ 扩展知识点: DPO的算法推导、DPO与RLHF的对比、DPO模型的变种（token-level DPO和reference-free DPO算法）

○ 考核方式：概念、公式、实现

●【2】奖励模型训练

○ 知识点：打分式、对比式、排序式奖励模型训练损失

○ 扩展知识点：奖励模型的泛化能力

○ 考核方式：概念、公式、实现

●【3】幻象

○ 知识点：幻象问题的概念与分类、幻象的起因

○ 扩展知识点：幻象的常见缓解方法

○ 考核方式：概念、实现

七．解码与部署

●【1】解码方法

○ 知识点：贪心搜索、束搜索解码的概念

○ 扩展知识点：束搜索的超参数调优

○ 考核方式：概念、公式

● **【2】随机采样及改进策略**

○ 知识点：温度采样、top-k采样、top-p采样

○ 扩展知识点：采样策略对生成多样性的影响

○ 考核方式：概念、公式、实现

● **【3】解码加速算法与实践**

○ 知识点：全量解码与增量解码、解码效率定量评估指标、常见推理工具使用（vLLM）

○ 扩展知识点：解码加速优化算法（推测解码、非自回归解码、早退机制、级联解码）、解码加速的系统级优化（FlashAttention、PagedAttention、批次管理优化）

○ 考核方式：概念

● **【3】低资源部署策略**

○ 知识点：量化基本概念、对称量化、非对称量化、量化粒度、常见量化方法、量化对模型性能的影响

○ 扩展知识点：量化工具的使用

○ 考核方式：概念、实现

● **【3】模型压缩方法：蒸馏、剪枝、量化**

○ 知识点：模型蒸馏的基本概念与基础方法、剪枝基本概念与基础方法、模型量化的基本概念与基础方法

○ 扩展知识点：蒸馏、剪枝和量化方法的使用

○ 考核方式：概念、实现

● 【3】资源管理与性能优化

○ 知识点：计算资源分配与调度、模型性能的瓶颈分析与优化

○ 扩展知识点：分布式资源管理、硬件与软件的协同优化

○ 考核方式：概念

八．提示学习

● **【1】提示工程**

○ 知识点：提示学习的目的、提示学习的范围与局限

○ 扩展知识点：提示方法的应用场景

○ 考核方式：概念、简答

● **【1】人工提示设计**

○ 知识点：常见提示设计方法与技巧、常见模型API的使用

○ 扩展知识点：提示设计的自动化方法

○ 考核方式：概念、实现

●【2】上下文学习

○ 知识点：上下文提示定义、模板、底层机制

○ 扩展知识点：上下文学习的增强策略

○ 考核方式：概念、实现

●【3】思维链提示

○ 知识点：思维链的基本形式、思维链的优化策略

○ 扩展知识点：思维链的基础原理

○ 考核方式：概念、实现

●【2】检索增强

○ 知识点：基本概念、常见使用方法

○ 扩展知识点：检索增强的增强策略（自主检索调用、效率提升等）

○ 考核方式：概念、实现

九．复杂推理

●【1】认知推理

○ 知识点：推理的基本方法与范畴

○ 扩展知识点：感知、认知与推理的区别

○ 考核方式：概念

●【1】长思维链模型

○ 知识点：长思维链推理模式的理解、测试时间扩展

○ 扩展知识点：使用推理模型解决常见逻辑、因果、数学、代码、科学任务

○ 考核方式：概念、实现

●【2】基于监督微调的推理模型训练

○ 知识点：长思维链数据的搜集与构建、长思维链指令蒸馏方法

○ 扩展知识点：以有监督微调方式进行推理模型训练

○ 考核方式：概念、实现

●【3】基于强化学习的推理模型训练

○ 知识点：以RL的方式进行推理能力的训练，包括结

果奖励建模和过程奖励建模

○ 扩展知识点：推理过程中的探索策略

○ 考核方式：概念、公式、实现

●【3】基于搜索的大模型推理

○ 知识点：基于搜索的测试时间扩展，在测试过程中通过多路径搜索（Self-consistency）、树搜索（Tree-of-thoughts）等提升模型推理能力

○ 扩展知识点：搜索效率与准确性的权衡

○ 考核方式：概念、实现

十．智能体

●【1】智能体身份与角色（profile）设置

○ 知识点：profile的设置方法、角色扮演、角色扮演能力优化

○ 扩展知识点：角色扮演中的一致性保持

○ 考核方式：概念、实现

●【2】智能体记忆机制

○ 知识点：智能体记忆种类、显式记忆和隐式记忆、记忆的存储和读取

○ 扩展知识点：记忆的长期保持与遗忘问题

○ 考核方式：概念、实现

●【2】智能体工具使用

○ 知识点：工具检索，工具调用优化方法

○ 扩展知识点：工具库的构建

○ 考核方式：概念、实现

●【3】多智能体通信结构

○ 知识点：典型结构、结构自主学习、通信数据优化

○ 扩展知识点：通信中的信息压缩与加密

○ 考核方式：概念、实现

●【3】多智能体组件优化

○ 知识点：提示调优、参数调优、结构调优

○ 扩展知识点：多智能体的协同学习

○ 考核方式：概念、实现

●【3】智能体 - 人协作

○ 知识点：协作种类、协作效率、协作优化

○ 扩展知识点：协作中的效率优化

○ 考核方式：概念、实现

●【3】智能体交互环境

○ 知识点：世界模型概念、智能体-环境交互、环境反馈

○ 扩展知识点：环境的基本构建与仿真方法

○ 考核方式：概念、实现

●【2】智能体典型应用

○ 知识点：了解WebGPT、社会模拟（斯坦福小镇等）

○ 扩展知识点：使用智能体框架搭建简单的应用

○ 考核方式：概念、实现

十一．模型评测

●【1】评测流程

○ 知识点：数据集划分、模型的泛化能力

○ 扩展知识点：泛化的理论保证

○ 考核方式：概念、简答

●【1】评测指标

○ 知识点：熟悉使用常见评测指标，如精确率、召回率、F1分数、困惑度、BLEU、ROUGE、准确率、成功率、NDCG等

○ 扩展知识点：评测指标的局限性

○ 考核方式：概念、公式、实现

●【1】评测范式与方法

○ 知识点：基于评测基准、基于人类评估、基于模型评估

○ 扩展知识点：评测中的公平性问题

○ 考核方式：概念

●【2】公开综合评测集

○ 知识点：MMLU、BIG-Bench、HELM、C-Eval等数据集的使用

○ 扩展知识点：评测集的构建方法、了解数据污染现象

○ 考核方式：概念

● 【2】语言能力评测

○ 知识点：语言能力评测基本任务及对应评测指标

○ 扩展知识点：领域特定任务的语言能力评测

○ 考核方式：概念

● 【2】知识利用能力评测

○ 知识点：知识利用能力评测基本任务（如闭卷问答、开卷问答、知识补全等）及对应评测指标

○ 扩展知识点：知识更新的时效性

○ 考核方式：概念

● 【2】复杂推理评测

○ 知识点：复杂推理能力评测基本任务及对应评测指标

○ 扩展知识点：推理评测的可靠性

○ 考核方式：概念

● 【3】其他评测

○ 知识点：人类对齐评测、环境交互评测、工具使用评测、鲁棒性评测（对抗样本）

○ 扩展知识点：理解高级评测任务与模型基础能力之间的关系

○ 考核方式：概念

十二．模型伦理与安全

● 【1】模型偏见

○ 知识点：偏见的来源、检测与缓解方法

○ 扩展知识点：偏见对实际应用的影响

○ 考核方式：概念、案例分析

● 【2】隐私保护

○ 知识点：数据隐私保护技术（如差分隐私）

○ 扩展知识点：隐私与模型性能的权衡

○ 考核方式：概念

● 【3】数据安全

○ 知识点：数据泄露风险、数据加密与访问控制

○ 扩展知识点：数据安全的法律与合规问题

○ 考核方式：概念

中小学组知识点表

模块	子模块	知识点
人工智能基础概念	人工智能相关概念定义	智能、人工智能、机器学习、模型、训练
	机器学习流程及经典模型	数据预处理、模型训练、验证、测试、神经网络
	验证及评测	交叉验证、精确率、召回率、受试者工作特征曲线
	人工智能常见应用领域	自然语言处理、计算机视觉
大模型基础概念	自然语言的基础概念	语言的定义、语言的特点、语言的应用
	基本定义	大语言模型的定义、核心范式（生成、理解）
	发展历程与现状	四代语言模型的发展历程（统计语言模型、神经网络语言模型、预训练语言模型、大语言模型）
模型架构	注意力机制	查询（query）- 键（key）- 值（value），softmax 函数
	主流架构	编码器 - 解码器架构、因果解码器架构、前缀解码器架构
预训练技术	回归分析	极大似然学习
	自监督学习	对比学习、掩码预测、自回归
	预训练任务	下一个词元预测（语言建模）、去噪自编码（如 BERT 的 MLM）
	基于批次数据的训练方法	Batch 及 Batch Size 概念、Batch Size 设置对模型训练的影响
	监督学习	监督学习、无监督学习
	指令微调	指令数据、指令遵循、指令微调的概念
	指令数据集的构建	指令数据合成方法
人类对齐	人类对齐的背景与标准	人类对齐背景、人类对齐标准（无害性、有用性、诚实性）
解码与部署	解码方法	贪心搜索、束搜索解码概念
提示学习	提示工程	提示学习的目的、提示学习的范围与局限
	人工提示设计	常见提示设计方法与技巧、常见模型 API 的使用

模块	子模块	知识点
复杂推理	认知推理	推理的基本方法与范畴
	长思维链模型	长思维链推理模式的理解、测试时间扩展
智能体	智能体身份与角色（profile）设置	profile 的设置方法、角色扮演、角色扮演能力优化
模型评测	评测流程	数据集划分、模型的泛化能力
	评测指标	熟悉使用常见评测指标，如精确率、召回率、F1 分数、困惑度、BLEU、ROUGE、准确率、成功率、NDCG 等
	评测范式与方法	基于评测基准、基于人类评估、基于模型评估
模型伦理与安全	模型偏见	偏见的来源、检测与缓解方法

成人组知识点表

模块	子模块	主要知识点示例
人工智能基础概念	人工智能相关概念定义	有监督（如分类、回归）、无监督（如聚类）等
	机器学习流程及经典模型	数据清洗、数据变换、数据工程等重要环节的概念
	验证及评测	过拟合等
	人工智能常见应用领域	智能检索、推荐系统等
大模型基础概念	自然语言的基础概念	自然语言与其他信息（如图像，声音，代码等）的区别
	基本定义	与传统 NLP 模型的区别、大模型的优势和局限性
	发展历程与现状	关键里程碑（如 GPT 系列、DeepSeek 系列等）
	扩展法则	KM 扩展法则、Chinchilla 扩展法则、计算最优模型规模、数据规模与计算资源的权衡
	大模型代表能力	上下文学习、指令微调、逐步推理等代表性涌现能力、涌现能力的合理解释

模块	子模块	主要知识点示例
模型架构	注意力机制	自注意力机制、交叉注意力机制
	主流架构	架构选择与任务类型的关系（如生成任务 vs. 理解任务）
	Transformer 模型的基本结构组成	位置编码的变体（如相对位置编码、旋转位置编码）
	Transformer 模型的详细配置	归一化方法（LayerNorm、RMSNorm）、归一化模块位置、激活函数（GELU、Swish）、注意力机制、不同配置对模型性能的影响、大模型可解释性分析方法（探针技术、注意力可视化、特征重要性分析）
	混合专家模型（MoE）	稀疏激活、路由机制、负载均衡、MoE 的显存优化与通信开销
	长上下文模型	位置编码拓展（如 ALiBi）、长上下文窗口拓展、长上下文数据构建、长上下文模型的显存与计算效率优化、长上下文模型的训练方法
	高效注意力架构	稀疏注意力、线性注意力、键值缓存优化、多查询注意力（MQA）、分组查询注意力（GQA）、多头潜在注意力（MLA）、高效注意力机制的理论基础（如低秩近似）

模块	子模块	主要知识点示例
预训练技术	回归分析	多目标分类学习任务
	自监督学习	自然语言与图像模态自监督学习的区别
	监督学习	监督学习与自监督学习的区别与联系
	预训练任务	下 N 个词元预测、预训练与多任务学习的关系
	基于批次数据的训练方法	动态批次大小调整策略
	学习率	学习率概念、学习率衰减（退火）、学习率预热（Warm-up）、自适应学习率方法（如 Cosine 衰减）
	优化器	Adam、SGD、优化器的改进版本（如 AdamW、LAMB）
	参数量计算	计算模型参数量的方法、MoE 模型的参数量计算
	训练运算量及时间计算	训练运算量及时间复杂度的计算方法、FLOPs 与训练时间的实际估算
	训练显存计算	预训练显存计算方法、显存优化技术（如梯度检查点）
	稳定优化技术	梯度裁剪、训练恢复、权重衰减、混合精度训练中的稳定性问题

模块	子模块	主要知识点示例
预训练技术	并行训练	3D 并行（数据并行、流水线并行、张量并行）、ZeRO 优化器与混合并行策略
	训练精度	单精度、半精度、混合精度概念，及其对模型训练的影响、BF16 与 FP16 的对比
	高效训练技术	完全分片数据并行、完全和选择性激活重计算、混合精度训练的流程、融合算子的原理、分布式训练中的通信优化
指令微调	模型解码	给定分布下的概率采样
	指令微调	指令微调与多任务学习的关系
	指令数据集的构建	高质量指令数据的筛选与清洗
	指令数据构建提升方法	指令进化算法、自引导指令增强（Bootstrapping）、长上下文指令构建、指令数据质量与泛化能力的关系
	微调优化设置	目标函数、批次大小、多指令合并高效训练、多阶段混合训练（长短指令、数据课程）、指令微调中的资源消耗估算方法
	数据组织策略	领域专家模型的指令过滤、代理模型引导的指令配比、基于导数的指令数据选择、垂直领域的指令数据构造与使用
	参数高效微调方法	低秩适配（LoRA）、适配器微调、前缀微调、提示微调、参数高效微调的理论基础
人类对齐	人类对齐的背景与标准	其他对齐标准（语言表达、道德标准等）
	人类偏好与反馈数据收集	人类反馈收集方法、基于评分的人类反馈、基于排序的人类反馈、反馈数据的偏差与修正
	非强化学习训练的对齐方法	DPO的公式及原理、DPO的算法推导、DPO与RLHF的对比、DPO模型的变种（token-level DPO和reference-free DPO算法）
	奖励模型训练	打分式、对比式、排序式奖励模型训练损失、奖励模型的泛化能力
	幻象	幻象问题的概念与分类、幻象的起因、幻象的常见缓解方法
解码与部署	解码方法	束搜索的超参数调优
	随机采样及改进策略	温度采样、top-k 采样、top-p 采样、采样策略对生成多样性的影响
	解码加速算法与实践	全量解码与增量解码、解码效率定量评估指标、常见推理工具使用（vLLM）、解码加速优化算法（推测解码、非自回归解码、早退机制、级联解码）、解码加速的系统级优化（FlashAttention、PagedAttention、批次管理优化）

模块	子模块	主要知识点示例
解码与部署	低资源部署策略	量化基本概念、对称量化、非对称量化、量化粒度、常见量化方法、量化对模型性能的影响、量化工具的使用
	模型压缩方法：蒸馏、剪枝、量化	模型蒸馏的基本概念与基础方法、剪枝基本概念与基础方法、模型量化的基本概念与基础方法、蒸馏、剪枝和量化方法的使用
	资源管理与性能优化	计算资源分配与调度、模型性能的瓶颈分析与优化、分布式资源管理、硬件与软件的协同优化
提示学习	提示工程	提示方法的应用场景
	人工提示设计	提示设计的自动化方法
	上下文学习	上下文提示定义、模板、底层机制、上下文学习的增强策略
	思维链提示	思维链的基本形式、思维链的优化策略、思维链的基础原理
	检索增强	基本概念、常见使用方法、检索增强的增强策略（自主检索调用、效率提升等）

模块	子模块	主要知识点示例
复杂推理	认知推理	感知、认知与推理的区别
	长思维链模型	使用推理模型解决常见逻辑、因果、数学、代码、科学任务
	基于监督微调的推理模型训练	长思维链数据的搜集与构建、长思维链指令蒸馏方法、以有监督微调方式进行推理模型训练
	基于强化学习的推理模型训练	以 RL 的方式进行推理能力的训练（包括结果奖励建模和过程奖励建模）、推理过程中的探索策略
	基于搜索的大模型推理	基于搜索的测试时间扩展，在测试过程中通过多路径搜索（Self-consistency）、树搜索（Tree-of-thoughts）等提升模型推理能力、搜索效率与准确性的权衡

模块	子模块	主要知识点示例
智能体	智能体身份与角色（profile）设置	角色扮演中的一致性保持
	智能体记忆机制	智能体记忆种类、显式记忆和隐式记忆、记忆的存储和读取、记忆的长期保持与遗忘问题
	智能体工具使用	工具检索、工具调用优化方法、工具库的构建
	多智能体通信结构	典型结构、结构自主学习、通信数据优化、通信中的信息压缩与加密
	多智能体组件优化	提示调优、参数调优、结构调优、多智能体的协同学习
	智能体 - 人协作	协作种类、协作效率、协作优化、协作中的效率优化
	智能体交互环境	世界模型概念、智能体 - 环境交互、环境反馈、环境的基本构建与仿真方法
	智能体典型应用	了解 WebGPT、社会模拟（斯坦福小镇等）、使用智能体框架搭建简单的应用
模型评测	评测流程	泛化的理论保证
	评测指标	评测指标的局限性
	评测范式与方法	评测中的公平性问题
	公开综合评测集	MMLU、BIG-Bench、HELM、C-Eval 等数据集的使用、评测集的构建方法、了解数据污染现象
	语言能力评测	语言能力评测基本任务及对应评测指标，领域特定任务的语言能力评测
	知识利用能力评测	知识利用能力评测基本任务（如闭卷问答、开卷问答、知识补全等）及对应评测指标、知识更新的时效性
	复杂推理评测	复杂推理评测基本任务及对应评测指标、推理评测的可靠性
	其他评测	人类对齐评测、环境交互评测、工具使用评测、鲁棒性评测（对抗样本）、理解高级评测任务与模型基础能力之间的关系

模块	子模块	主要知识点示例
模型伦理与安全	模型偏见	偏见对实际应用的影响
	隐私保护	数据隐私保护技术（如差分隐私）、隐私与模型性能的权衡
	数据安全	数据泄露风险、数据加密与访问控制、数据安全的法律与合规问题

LMCC 第一轮样题

人工智能基础概念

●【1】人工智能相关概念定义

○ 知识点：智能、人工智能、机器学习、模型、训练等重要概念的定义

题目：

以下关于人工智能与机器学习的说法，正确的是：

A. 人工智能是指所有基于规则系统的程序，而机器学习不属于人工智能

B. 机器学习是人工智能的一个分支，侧重于通过数据训练模型以实现智能行为

C. 模型是指数据本身，训练是指对数据进行排序

D. 智能仅指人类的思维能力，机器不可能具备智能

答案：B

解析：

A 错误：机器学习属于人工智能

B 正确：标准定义

C 错误：模型是函数 / 映射，训练是学习参数

D 错误：机器也可以表现出“智能行为”

○ 扩展知识点：有监督（如分类、回归）、无监督（如聚类）等概念

题目：

下列关于有监督学习与无监督学习的描述，正确的是：

A. 有监督学习不需要标签数据，主要用于发现数据结构

B. 无监督学习需要大量标注数据才能训练

C. 分类和回归属于有监督学习，而聚类属于无监督学习

D. 聚类算法的目标是预测连续数值

答案：C

解析：

A 错误：这是无监督学习

B 错误：无监督学习不需要标签

C 正确：经典划分

D 错误：预测连续值是回归

○ 考核方式：概念

●【1】机器学习流程及经典模型

○ 知识点：数据预处理、模型训练、验证、测试、神经网络

题目：

在机器学习流程中，下列步骤顺序最合理的是：

A. 模型测试 → 数据采集 → 模型训练 → 数据预处理

B. 数据预处理 → 模型训练 → 模型验证 → 模型测试

C. 模型训练 → 数据清洗 → 模型部署 → 数据收集

D. 数据分析 → 模型测试 → 数据清洗 → 模型训练

答案：B

解析：

标准流程：

数据预处理 → 训练 → 验证 → 测试

○ 扩展知识点：数据清洗、数据变换、数据工程等重要环节的概念

题目：

关于数据清洗与数据工程的描述，正确的是：

A. 数据清洗主要是对模型参数进行优化

B. 数据工程包括数据采集、清洗、变换等多个环节

C. 数据变换只在模型训练完成后进行

D. 数据清洗的主要目的是增加数据量

答案：B

解析：

A 错误：数据清洗处理数据而不是模型

B 正确：完整定义

C 错误：在训练前进行

D 错误：是提高质量不是数量

○ 考核方式：概念

●【1】验证及评测

○ 知识点：交叉验证，精确率（Precision）、召回率（Recall）、受试者工作特征曲线（ROC）等重要评测指标

题目：

关于精确率（Precision）和召回率（Recall），正确的是：

A. 精确率表示所有正样本中被正确预测的比例

B. 召回率表示预测为正的样本中有多少是正确的

C. 精确率关注预测结果的准确性，召回率关注对正样本的覆盖能力

D. 两者完全等价，只是名称不同

答案：C

解析：

A 错误：这是召回率的定义，不是精确率

B 错误：这是精确率的定义，不是召回率

C 正确：精确率关注 " 预测为正的准确性 "（Precision = TP/(TP+FP)），召回率关注 " 正样本的覆盖率 "（Recall = TP/(TP+FN)）

D 错误：两者定义和关注点不同

○ 扩展知识点：过拟合等概念

题目：

关于交叉验证与过拟合，下列说法正确的是：

A. 交叉验证的主要作用是增加模型复杂度

B. 过拟合是模型在训练数据上表现很好，在新数据上表现较差

C. 过拟合是因为模型太简单

D. 交叉验证只能用于无监督学习

答案：B

解析：

A 错误：用于评估泛化能力

B 正确：过拟合定义

C 错误：那是欠拟合

D 错误：主要用于监督学习

○ 考核方式：概念

●【1】人工智能常见应用领域

○ 知识点：自然语言理解、计算机视觉等

题目：

以下属于人工智能应用领域的是：

A. 自然语言理解和计算机视觉

B. 仅限于数据库管理系统

C. 只用于数值计算

D. 只应用于硬件设计

答案：A

解析：

AI 典型领域：

NLP（自然语言处理）

CV（计算机视觉）

○ 扩展知识点：智能检索、推荐系统等

题目：

关于推荐系统与智能检索的描述，正确的是：

A. 推荐系统主要依赖人工规则，不使用数据

B. 智能检索无法理解语义，只能进行关键词匹配

C. 推荐系统和智能检索都可以利用机器学习技术提升效果

D. 推荐系统只能用于电商领域

答案：C

解析：

A 错误：核心是数据驱动

B 错误：现代检索支持语义理解

C 正确

D 错误：应用广泛（视频、音乐等）

○ 考核方式：概念

大模型基础概念

●【1】自然语言的基础概念

○ 知识点：语言的定义，语言的特点、语言的应用

题目：

关于自然语言（Natural Language）的特点，下列说法错误的是：

A. 离散性：自然语言由离散的符号（如汉字、单词）组成

B. 抽象性：语言符号可以代表具体的实物，也可以代表抽象的概念

C. 确定性：自然语言的语义在任何语境下都是唯一确定、

没有歧义的

D. 社会性：语言是人类在社会发展中约定俗成的沟通工具

答案：C

解析：

A、B、D 均是自然语言的典型特征。C 错误：自然语言具有高度的歧义性（Ambiguity），同一个词在不同语境下可能有完全不同的含义（如“意思”一词），这正是自然语言处理（NLP）的难点所在。

○ 扩展知识点：与其他信息（如图像，声音，代码等）的区别

题目：

与图像数据（Image）相比，自然语言在作为大模型输入数据时，最显著的物理特征区别是：

A. 图像是密集的像素矩阵，而自然语言通常表现为离散的序列

B. 语言信号比图像信号包含更多的冗余像素信息

C. 图像具有天然的语序结构，而语言没有先后顺序

D. 计算机处理图像不需要数字化，而处理语言必须数字化

答案：A

解析：

A 正确：图像在底层表现为密集的数值分布，而语言由离散的符号组成。

B 错误：图像通常冗余度更高。

C 错误：语言具有强烈的序列（时序）特征，图像则具有空间特征。

D 错误：两者在计算机中都需要转化为数字（向量 / 张量）。

○ 考核方式：概念、简答题

●【1】基本定义

○ 知识点：大语言模型的定义、核心范式（生成、理解）

题目：

关于大语言模型的代表性模型，以下哪项描述最准确？

A. GPT、Llama 等基于 Transformer 的预训练模型通常被视为大语言模型

B. Transformer 架构模型参数量必然很小，不可能成为大语言模型

C. 卷积神经网络是大语言模型的唯一形式

D. 线性回归模型由于结构简单，是大语言模型的典型代表

答案：A

解析：

大语言模型主要基于 Transformer 架构，GPT、Llama 系列是其代表性模型。Transformer 架构恰恰能够支持大规模参数（B 错误）；卷积神经网络主要用于图像处理（C 错误）；线性回归是传统统计模型，不属于大语言模型（D 错误）。

○ 扩展知识点：与传统 NLP 模型的区别、大模型的优势和局限性

题目：

关于大语言模型的优势与局限性，以下描述正确的是?

A. 大语言模型在创造性任务上表现优异，但可能存在事实性错误

B. 大语言模型完全不会产生偏见内容

C. 大语言模型在所有任务上都优于人类专家

D. 大语言模型不需要大量训练数据即可达到最佳效果

答案：A

解析： 大语言模型在文本生成、创造性写作等方面表现出色，但可能产生 " 幻觉 "（hallucination），即生成看似合理但实际错误的内容。模型会继承训练数据中的偏见（B 错误）、在某些专业领域不如人类专家（C 错误）、需要海量数据训练（D 错误）。

○ 考核方式：概念、简答题

●【1】发展历程与现状

○ 知识点：四代语言模型的发展历程（统计语言模型、神经网络语言模型、预训练语言模型、大语言模型）

题目：

在语言模型的发展史上，以下技术或模型按时间先后顺序排列正确的是：

A. 循环神经网络 (RNN) → N-gram 统计模型 → Transformer → 大语言模型

B. N-gram 统计模型 → 循环神经网络 (RNN) → 预训练模型 (如 BERT) → 大语言模型

C. 大语言模型 → Transformer → 统计语言模型 → 神经网络模型

D. 专家系统 → 大语言模型 → 统计语言模型 → 神经网络模型

答案：B

解析：

语言模型经历了：1. 统计语言模型（以 N-gram 为代表）；2. 神经网络语言模型（以 RNN/LSTM 为代表）；3. 预训练语言模型（以 BERT/GPT-1 为代表）；4. 大语言模型（以 GPT-3 及之后的大规模参数模型为代表）。

○ 扩展知识点：关键里程碑（如 GPT 系列、DeepSeek 系列等）

题目：

关于大模型发展过程中的里程碑事件，下列说法错误的是：

A. GPT-3 的发布证明了通过简单增加模型参数和数据量可以实现强大的通用能力

B. Transformer 架构的提出为现代大模型奠定了并行计算的基础

C. DeepSeek 系列模型展示了在模型架构优化和推理成本降低方面的国产技术进步

D. GPT-1 是历史上第一个采用“仅编码器（Encoder-only）”架构取得成功的模型

答案：D

解析：

D 错误：GPT 系列（Generative Pre-training）采用的是仅解码器（Decoder-only）架构；而 BERT 采用的是仅编码器（Encoder-only）架构。

○ 考核方式：概念、简答题

●【2】扩展法则

○ 知识点：KM 扩展法则、Chinchilla 扩展法则

题目：

关于大模型的扩展法则（Scaling Laws），下列描述最符合 KM 法则核心观点的是：

A. 模型性能主要受限于模型架构的精细程度，而非数据量

B. 模型性能与参数量、数据集大小及计算量之间存在幂律关系，且随着规模增加性能可预测地提升

C. 只要模型参数量翻倍，推理速度就会自动提升一倍

D. 当模型达到一定规模后，增加数据量反而会导致性能急剧下降

答案：B

解析：B 正确：KM 扩展法则指出，LLM 的性能（Loss）与计算量 (C)、参数量 (N) 和数据量 (D) 之间存在稳定的幂律关系。

○ 扩展知识点：计算最优模型规模、数据规模与计算资源的权衡

题目：

DeepMind 提出的 Chinchilla 扩展法则给出的核心启示是：

A. 只要算力充足，参数量越大的模型性能永远越好，无需考虑数据量

B. 当算力预算增加时，应同比例地增加模型参数量和训练

数据量，许多早期模型往往“训练不足”

C. 175B 参数的模型是人类能训练的极限

D. 模型的训练数据应该全部来自教科书，以保证质量

答案：B

解析：

Chinchilla 研究发现，很多大模型（如 GPT-3）在给定算力下参数量过大而数据量不足。结论是：算力增加时，参数量和数据量应平衡增长，即“计算最优（Compute Optimal）”原则。

○ 考核方式：概念、计算题

●【2】大模型代表能力

○ 知识点：上下文学习、指令微调、逐步推理等代表性涌现能力

题目：

大模型在没有进行权重更新的前提下，仅通过在提示词（Prompt）中加入几个示例就能学会处理新任务，这种能力被称为：

A. 监督微调 (Supervised Fine-tuning)

B. 知识蒸馏 (Knowledge Distillation)

C. 上下文学习 (In-context Learning)

D. 持续学习 (Continual Learning)

答案：C

解析：

上下文学习（In-context Learning）是大模型的核心涌现能力之一，指模型在推理阶段通过提示词中的示例直接完成任务，而不改变模型参数。

○ 扩展知识点：涌现能力的合理解释

题目：

关于大模型的“涌现能力（Emergent Abilities）”，以下理解最准确的是：

A. 涌现能力是指小模型也具备，只是在大模型中表现得更明显而已

B. 涌现能力通常指在模型参数量超过一定阈值后，模型在某些复杂任务上的表现突然从“随机猜测”跃升至“远超随机”的现象

C. 涌现能力意味着模型已经产生了自我意识

D. 涌现能力只能通过增加模型的层数获得，与参数量无关

答案：B

解析：

涌现能力（Emergence）的定义是在小规模模型中不存在，

但在大规模模型中突然表现出的能力（如逻辑推理、复杂数学解题等），这种转变往往是非线性的跃迁。

○ 考核方式：概念、案例分析、API 调用

模型架构

●【1】注意力机制

○ 知识点：查询（query）- 键（key）- 值（value），softmax 函数

题目：

关于注意力机制中查询（Query）、键（Key）、值（Value）的作用，下列说法正确的是：

A. Query 用于存储待检索的信息，Key 用于生成最终输出

B. 注意力分数通过 Query 与 Key 的点积计算得到，再经 softmax 归一化后对 Value 加权求和

C. softmax 函数的作用是将注意力分数映射到 $(-\infty, +\infty)$ 的范围

D. Value 向量决定了哪些位置应该获得更高的注意力权重

答案：B

解析：

A 错误：Query 是查询向量，Key 是键向量，Value 才是存储信息的向量

B 正确：标准注意力机制计算流程为 Attention(Q,K,V) = softmax(QK^T/ $\sqrt{}$ d_k)V

C 错误：softmax 将分数归一化到 (0,1) 区间且总和为 1，形成概率分布

D 错误：注意力权重由 Query 和 Key 的相似度决定，Value 是被加权的对象

○ 扩展知识点：自注意力机制、交叉注意力机制

题目：

关于自注意力机制与交叉注意力机制的区别，下列描述正确的是：

A. 自注意力机制中 Query、Key、Value 均来自同一序列，交叉注意力中 Query 来自一个序列而 Key 和 Value 来自另一个序列

B. 自注意力机制只能用于编码器，交叉注意力只能用于解码器

C. 交叉注意力中 Query、Key、Value 都来自同一个输入序列

D. 自注意力机制不使用 softmax 函数进行归一化

答案：A

解析：

A 正确：这是两者的核心区别，自注意力 Q/K/V 同源，交叉注意力 Q 与 K/V 不同源

B 错误：自注意力在编码器和解码器中都可以使用

C 错误：这是自注意力的描述，交叉注意力的 Q 和 K/V 来自不同序列

D 错误：自注意力同样使用 softmax 归一化

○ 考核方式：概念、计算

●【1】主流架构

○ 知识点：编码器 - 解码器架构、因果解码器架构、前缀解码器架构

题目：

关于大语言模型的三种主流架构，下列描述正确的是：

A. 因果解码器架构允许每个位置关注序列中所有位置的信息

B. 编码器 - 解码器架构中，编码器使用双向注意力，解码器使用因果（单向）注意力

C. 前缀解码器架构与因果解码器架构完全相同，没有任何区别

D. 编码器 - 解码器架构只能处理分类任务，无法进行文本生成

答案：B

解析：

A 错误：因果解码器使用因果掩码，每个位置只能关注自身及之前的位置

B 正确：编码器 - 解码器架构的标准设计，编码器双向、解码器单向

C 错误：前缀解码器在前缀部分使用双向注意力，生成部分使用因果注意力，与纯因果解码器不同

D 错误：编码器 - 解码器架构可用于翻译、摘要等多种序列到序列的生成任务

○ 扩展知识点：架构选择与任务类型的关系（如生成任务 vs. 理解任务）

题目：

关于模型架构选择与任务类型的关系，下列说法正确的是：

A. 编码器架构（如 BERT）更适合自然语言理解任务，因果解码器架构（如 GPT）更适合文本生成任务

B. 因果解码器架构只能用于文本生成，完全不能做理解任务

C. 编码器 - 解码器架构只适合机器翻译，不适合其他任何任务

D. 所有任务都应该使用同一种架构，架构选择对性能没有影响

答案：A

解析：

A 正确：编码器架构通过双向注意力擅长理解任务，因果解码器通过自回归生成擅长生成任务

B 错误：GPT 等因果解码器通过提示也能完成理解类任务

C 错误：编码器 - 解码器架构还可用于摘要、问答等多种任务

D 错误：架构选择对任务性能有显著影响

○ 考核方式：概念、架构对比分析

●【1】Transformer 模型的基本结构组成

○ 知识点：输入编码、位置编码、多头自注意力机制、前馈网络层、编码器、解码器

题目：

关于 Transformer 模型的基本结构组成，下列描述正确的是：

A. Transformer 仅由注意力层组成，不包含前馈网络层

B. Transformer 的编码器由多头自注意力层和前馈网络层交替堆叠而成，每层之后通常有残差连接和层归一化

C. 位置编码的作用是对输入词汇进行语义编码，与位置顺序无关

D. 多头注意力机制是指只使用一个注意力头来处理全部信息

答案：B

解析：

A 错误：Transformer 每层都包含注意力子层和前馈网络（FFN）子层

B 正确：标准 Transformer 编码器的结构描述

C 错误：位置编码的作用是为模型注入序列中 token 的位置信息

D 错误：多头注意力是将输入投影到多个子空间，并行计算多组注意力

○ 扩展知识点：位置编码的变体（如相对位置编码、旋转位置编码）

题目：

以下代码实现了旋转位置编码（RoPE），请阅读代码并填写空缺部分。

```
import torch
import math

def apply_rotary_pos_emb(q, k, pos):
"""
对 Query 和 Key 应用旋转位置编码（RoPE）
参数 :
q: Query 张量 , shape (batch, heads, seq_len, head_dim)
k: Key 张量 , shape (batch, heads, seq_len, head_dim)
pos: 位置索引 , shape (seq_len,)
返回 :
旋转编码后的 q 和 k
```

```
"""
head_dim = q.size(-1)
# 计算频率基底 : theta_i = 1 / (10000 ^ (2i / d))
freq_indices = torch.arange(0, head_dim, 2, dtype=torch.float32)
freqs = 1.0 / (10000.0 ** (freq_indices / head_dim))

# 计算位置角度 : pos * theta
angles = pos.unsqueeze(-1).float() * freqs.unsqueeze(0)  # (seq_len, head_dim//2)

cos_vals = torch.cos(angles)  # (seq_len, head_dim//2)
sin_vals = torch.sin(angles)  # (seq_len, head_dim//2)

# 将 q 拆分为偶数维和奇数维
q_even = q[..., 0::2]  # (..., head_dim//2)
q_odd  = q[..., 1::2]  # (..., head_dim//2)

# [1] 对 Query 应用旋转变换 : q_rotated_even = q_even * cos - q_odd * sin
_____
# [2] 对 Query 应用旋转变换 : q_rotated_odd = q_even * sin + q_odd * cos
```

[1] 和 [2] 处应分别填入：

A. [1] q_rotated_even = q_even * cos_vals - q_odd * sin_vals [2] q_rotated_odd = q_even * sin_vals + q_odd * cos_vals

B. [1] q_rotated_even = q_even * sin_vals - q_odd * cos_vals [2] q_rotated_odd = q_even * cos_vals + q_odd * sin_vals

C. [1] q_rotated_even = q_even + cos_vals [2] q_rotated_odd = q_odd + sin_vals

D. [1] q_rotated_even = q_even * cos_vals + q_odd * sin_vals [2] q_rotated_odd = q_even * sin_vals - q_odd * cos_vals

答案：A

解析：

A 正确：RoPE 的旋转变换公式为 [x_even', x_odd'] = [x_even * cos - x_odd * sin, x_even * sin + x_odd * cos]，这是二维旋转矩阵的标准形式

B 错误：sin 和 cos 的位置颠倒了，不符合旋转矩阵的定义

C 错误：旋转位置编码是乘法操作而非加法，加法是绝对位置编码的做法

D 错误：第一行的符号应为减号（coseven - sinodd），第二行应为加号（sineven + cosodd），D 选项符号相反

○ 考核方式：概念、公式、代码实现

●【2】Transformer 模型的详细配置

○ 知识点：归一化方法（LayerNorm、RMSNorm）、归一化模块位置、激活函数（GELU、Swish）、注意力机制

题目：

关于 Transformer 模型中的归一化和激活函数配置，下列说法正确的是：

A. RMSNorm 相比 LayerNorm 省去了均值的计算，仅使用均方根进行归一化，计算效率更高

B. LayerNorm 和 RMSNorm 的计算方式完全相同，没有区别

C. GELU 激活函数与 ReLU 完全一致，都在负数区域输出恒为零

D. Pre-Norm 和 Post-Norm 的区别仅在于是否使用归一化，放置位置无影响

答案：A

解析：

A 正确：RMSNorm 是 LayerNorm 的简化版本，省去了减均值的步骤，只做均方根归一化

B 错误：RMSNorm 省略了均值中心化步骤

C 错误：GELU 在负数区域有平滑的非零输出，而 ReLU 在负数区域输出恒为 0

D 错误：Pre-Norm 在注意力 /FFN 之前归一化，Post-Norm 在之后归一化，位置不同影响训练稳定性

○ 扩展知识点：不同配置对模型性能的影响、大模型可解释性分析方法（探针技术、注意力可视化、特征重要性分析）

题目：

关于大模型可解释性分析方法，下列描述正确的是：

A. 探针技术通过在模型中间层训练简单分类器来探测模型是否学到了特定的语言知识

B. 注意力可视化可以直接证明模型的推理因果过程

C. 特征重要性分析只适用于传统机器学习模型，不能用于大语言模型

D. 大模型的可解释性分析方法目前只有注意力可视化一种

答案：A

解析：

A 正确：探针技术（Probing）通过在冻结的模型隐藏层上训练线性分类器来检测特定知识

B 错误：注意力权重分布只反映相关性，不能直接等同于因果推理过程

C 错误：特征重要性分析（如梯度分析、SHAP 等）也可以用于大语言模型

D 错误：还有探针技术、特征重要性分析、激活值分析等多种方法

○ 考核方式：概念、公式、实现与分析

●【3】混合专家模型（MoE）

○ 知识点：稀疏激活、路由机制、负载均衡

题目：

以下代码实现了 MoE（混合专家模型）的路由机制和稀疏激活，请阅读代码并填写空缺部分。

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class MoELayer(nn.Module):
```

```
def __init__(self, input_dim, hidden_dim, num_experts, top_k=2):
super().__init__()
self.num_experts = num_experts
self.top_k = top_k
# 路由网络：将输入映射到专家概率分布
self.gate = nn.Linear(input_dim, num_experts, bias=False)
# 多个专家网络
self.experts = nn.ModuleList([
nn.Sequential(
nn.Linear(input_dim, hidden_dim),
nn.ReLU(),
nn.Linear(hidden_dim, input_dim)
) for _ in range(num_experts)
])

def forward(self, x):
# x: (batch_size, seq_len, input_dim)
batch_size, seq_len, dim = x.shape
x_flat = x.view(-1, dim)  # (batch*seq, dim)

# [1] 通过路由网络计算每个 token 对应各专家的分数，并用 softmax 归一化
router_logits = self.gate(x_flat)       # (batch*seq, num_experts)
_____

# [2] 选择 Top-K 个专家
_____
```

[1] 和 [2] 处应分别填入：

A. [1] router_probs = F.softmax(router_logits, dim=-1) [2] top_k_weights, top_k_indices = torch.topk(router_probs, self.top_k, dim=-1)

B. [1] router_probs = F.sigmoid(router_logits) [2] top_k_weights, top_k_indices = torch.topk(router_probs, self.top_k, dim=-1)

C. [1] router_probs = F.softmax(router_logits, dim=0) [2] top_k_weights, top_k_indices = torch.sort(router_probs, dim=-1)

D. [1] router_probs = F.relu(router_logits) [2] top_k_weights, top_k_indices = torch.topk(router_probs, self.num_experts, dim=-1)

答案：A

解析：

A 正确：路由机制先用 softmax(dim=-1) 将 logits 转为专家概率分布，再用 topk 选出权重最高的 K 个专家

B 错误：sigmoid 将每个值独立映射到 (0,1)，不是互斥的概率分布，路由应使用 softmax

C 错误：softmax 应沿专家维度（dim=-1）归一化，dim=0 是 batch 维度；sort 返回全部排序结果而非 top-k

D 错误：ReLU 不能将 logits 转为概率分布；选取 num_experts 个专家等于激活全部专家，违背稀疏激活原则

○ 扩展知识点：MoE 的显存优化与通信开销

题目：

关于 MoE 模型的显存优化与通信开销，下列描述正确的是：

A. MoE 模型虽然计算量与稠密模型相近，但由于所有专家参数都需要加载，显存占用远大于同等计算量的稠密模型

B. MoE 模型在分布式训练中不需要跨设备通信，因为每个专家独立工作

C. MoE 模型的显存占用与稠密模型完全相同

D. 专家并行（Expert Parallelism）会增加显存占用但不会带来通信开销

答案：A

解析：

A 正确：MoE 虽然每个 token 只激活少量专家，但所有专家参数需常驻显存

B 错误：专家并行需要 All-to-All 通信来路由 token 到对应的专家所在设备

C 错误：MoE 总参数量大，显存占用通常远大于同等计算量的稠密模型

D 错误：专家并行中 token 需要在设备间传输，会带来显著的通信开销

○ 考核方式：概念、公式、实现

●【3】长上下文模型

○ 知识点：位置编码拓展（如 ALiBi）、长上下文窗口拓展、长上下文数据构建

题目：

关于长上下文模型中的位置编码拓展方法，下列说法正确的是：

A. ALiBi（Attention with Linear Biases）不使用位置编码向量，而是在注意力分数上直接加上与距离成正比的线性偏置

B. 所有位置编码方法都无法处理超出训练时最大长度的序列

C. 长上下文窗口拓展只需要修改位置编码，不需要使用长文本数据进行额外训练

D. ALiBi 需要为每个位置学习独立的位置嵌入向量

答案：A

解析：

A 正确：ALiBi 在注意力分数上减去一个与 query-key 距离成正比的偏置项，距离越远惩罚越大

B 错误：ALiBi、RoPE 插值等方法可以较好地外推到更长序列

C 错误：长上下文拓展通常还需要长文本数据进行继续训练以适应长序列

D 错误：ALiBi 是预定义的线性偏置，无需学习位置嵌入

○ 扩展知识点：长上下文模型的显存与计算效率优化、长上下文模型的训练方法

题目：

关于长上下文模型的效率优化和训练方法，下列描述正确

的是：

A. 标准自注意力的计算复杂度与序列长度呈线性关系

B. 长上下文模型通常采用渐进式训练策略，先在短序列上训练再逐步增加序列长度

C. 增加上下文窗口长度不会增加任何额外的显存消耗

D. 长上下文模型只需要使用短文本数据就可以获得良好的长文本处理能力

答案： B

解析：

A 错误：标准自注意力的计算复杂度与序列长度呈二次方关系 $O(n^2)$

B 正确：渐进式训练（逐步增加序列长度）是常用的长上下文训练策略

C 错误：更长的上下文会显著增加 KV 缓存的显存消耗

D 错误：长上下文模型需要高质量的长文本数据进行训练

○ 考核方式：概念、方法、实现

●【3】高效注意力架构

○ 知识点：稀疏注意力、线性注意力、键值缓存优化、多查询注意力（MQA）、分组查询注意力（GQA）、多头潜在注意力（MLA）

题目：

以下代码实现了分组查询注意力（GQA），请阅读代码并填写空缺部分。

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import math

class GroupedQueryAttention(nn.Module):
""" 分组查询注意力（GQA）: 多个 Query 头共享同一组 Key 和 Value 头 """
def __init__(self, hidden_dim, num_q_heads, num_kv_heads):
super().__init__()
self.num_q_heads = num_q_heads   # Query 头的数量，例如 32
self.num_kv_heads = num_kv_heads  # KV 头的数量，例如 8
self.head_dim = hidden_dim // num_q_heads
# [1] 计算每个 KV 头对应多少个 Query 头（分组大小）
_____

self.q_proj = nn.Linear(hidden_dim, num_q_heads * self.head_dim)
self.k_proj = nn.Linear(hidden_dim, num_kv_heads * self.head_dim)
self.v_proj = nn.Linear(hidden_dim, num_kv_heads * self.head_dim)
self.o_proj = nn.Linear(hidden_dim, hidden_dim)

def forward(self, x):
B, L, _ = x.shape

q = self.q_proj(x).view(B, L, self.num_q_heads, self.head_dim).transpose(1, 2)
k = self.k_proj(x).view(B, L, self.num_kv_heads, self.head_dim).transpose(1, 2)
v = self.v_proj(x).view(B, L, self.num_kv_heads, self.head_dim).transpose(1, 2)
```

[1] 和 [2] 处应分别填入：

A. [1] self.num_groups = num_q_heads // num_kv_heads [2] k = k.repeat_interleave(self.num_groups, dim=1) 和 v = v.repeat_

interleave(self.num_groups, dim=1)

B. [1] self.num_groups = num_kv_heads // num_q_heads [2] k = k.repeat(1, self.num_groups, 1, 1) 和 v = v.repeat(1, self.num_groups, 1, 1)

C. [1] self.num_groups = num_q_heads // num_kv_heads [2] k = k.expand(B, self.num_q_heads, L, self.head_dim) 和 v = v.expand(B, self.num_q_heads, L, self.head_dim)

D. [1] self.num_groups = num_q_heads * num_kv_heads [2] k = k.repeat_interleave(self.num_groups, dim=1) 和 v = v.repeat_interleave(self.num_groups, dim=1)

答案：A

解析：

A 正确：GQA 中 num_q_heads/num_kv_heads 得到每组的 Q 头数量；repeat_interleave 沿 dim=1 逐元素重复，使每个 KV 头被相邻的 Q 头共享

B 错误：分组大小应为 q_heads/kv_heads 而非 kv_heads/q_heads；repeat 会整体重复而非逐元素交错重复，分组对应关系会错乱

C 错误：expand 不能直接将 (B,num_kv_heads,L,d) 扩展为 (B,num_q_heads,L,d)，因为 num_kv_heads ≠ num_q_heads 时维度不匹配

D 错误：分组大小应为除法而非乘法，q_heads*kv_heads 会导致过度重复

○ 扩展知识点：高效注意力机制的理论基础（如低秩近似）

题目：

关于高效注意力机制的理论基础，下列描述正确的是：

A. 低秩近似的核心思想是利用注意力矩阵的低秩特性，用较少的参数近似完整的注意力计算

B. 线性注意力的计算复杂度仍然是 $O(n^2)$，与标准注意力相同

C. 低秩近似方法会大幅降低模型精度，因此在实际中不可使用

D. FlashAttention 属于低秩近似方法的一种

答案：A

解析：

A 正确：注意力矩阵在实践中通常具有低秩特性，可以用低秩分解来高效近似

B 错误：线性注意力通过核函数技巧将复杂度降低到 O(n)

C 错误：合理的低秩近似可以在几乎不损失精度的情况下大幅提升效率

D 错误：FlashAttention 是 IO 感知的精确注意力计算优化，不是低秩近似

○ 考核方式：概念、公式、实现

预训练技术

●【1】回归分析

○ 知识点：线性回归、逻辑回归、极大似然学习

题目：

以下哪种模型最适合处理二分类问题，并能够输出一个表示概率的连续值?

A. 线性回归 (Linear Regression)

B. 逻辑回归 (Logistic Regression)

C. K-Means 聚类

D. 主成分分析 (PCA)

答案：B

解析：

A 错误：线性回归主要用于预测连续数值，不适合直接用于分类。

B 正确：逻辑回归通过 Sigmoid 函数将线性模型的输出映射到 (0, 1) 区间，常用于二分类问题的概率估计。

C、D：K-Means 是无监督聚类算法，PCA 是降维算法，均不直接用于分类任务。

○ 扩展知识点：多目标分类学习任务

题目：

当一个学习任务需要同时预测多个相互关联的类别标签时（例如，在一张图片中同时识别出“人”和“狗”），这属于哪种类型的学习任务?

A. 单标签多类别分类 (Multi-class Classification)

B. 多标签分类 (Multi-label Classification)

C. 回归分析 (Regression Analysis)

D. 异常检测 (Anomaly Detection)

答案：B

解析：

B 正确：多标签分类允许一个样本属于多个类别。单标签多类别分类则要求样本只属于其中一个类别。

○ 考核方式：概念、公式

●【1】自监督学习

○ 知识点：对比学习、掩码预测、自回归

题目：

自监督学习中的“掩码预测”（Masked Prediction）任务，例如 BERT 的 Masked Language Model (MLM)，其核心思想是：

A. 学习将输入数据（如文本）在不同视图下进行编码，使其在表示空间中距离拉近

B. 随机遮盖（mask）输入序列中的一部分词元，并让模型预测被遮盖的词元

C. 预测输入序列中的下一个词元

D. 将输入数据映射到低维空间，以捕获主要变化趋势

答案：B

解析：

A 描述的是对比学习。C 描述的是自回归语言模型。D 描述的是降维。B 准确描述了掩码预测的核心机制。

○ 扩展知识点：自然语言与图像模态自监督学习的区别

题目：

在自监督学习中，针对自然语言（如文本）的掩码预测任务（如 MLM）与针对图像的掩码预测任务（如 MAE - Masked Autoencoders）相比，主要区别在于：

A. 图像掩码预测直接操作像素，而文本掩码预测操作的是离散的词元 / 子词

B. 文本掩码预测的“遮盖”比例通常远大于图像掩码预测

C. 图像掩码预测更容易受到全局上下文信息的影响

D. 图像掩码预测通常使用自回归方式，而文本是基于重构

答案：A

解析：

A 正确：文本是离散符号序列，遮盖后需要预测离散符号；图像是连续像素，遮盖后重构连续像素。

B 错误：MAE（MAE - Masked Autoencoders）的遮盖比例可以非常高（例如 75%）。

C 错误：文本的掩码预测（如 BERT）非常依赖全局上下文。

D 错误：MAE 是重构，文本 MLM 也是重构。

○考核方式：概念、公式、简答

●【1】预训练任务

○ 知识点：下一个词元预测（语言建模）、去噪自编码（如 BERT 的 MLM）

题目：

在预训练技术中，下一个词元预测任务（Next Token Prediction）主要基于以下哪种核心思想？

A. 根据前文词元序列预测下一个可能出现的词元

B. 随机遮盖输入序列中的部分词元并进行重构

C. 同时预测输入序列中所有位置的词元

D. 将输入序列转换为图像特征进行识别

答案：A

解析：下一个词元预测是自回归语言模型（如 GPT）的核心训练任务，模型根据已有的前文序列预测下一个词元。B 选项描述的是掩码语言模型（如 BERT）的训练方式。C 和 D 选项都不是标准的语言模型训练方式。

○ 扩展知识点：下 N 个词元预测、预训练与多任务学习的关系

题目：

如果一个预训练模型在训练时，除了预测下一个词元，还被要求完成例如情感分析、问答等多个下游任务（并在训练时共享参数），这种训练方式更接近于：

A. 对比学习 (Contrastive Learning)

B. 多任务学习 (Multi-task Learning)

C. 掩码语言模型 (Masked Language Model)

D. 强化学习 (Reinforcement Learning)

答案：B

解析：

B 正确：多任务学习是指模型在训练过程中同时学习多个相关任务，以期通过共享表示（参数）来提高整体性能和泛化能力。

○ 考核方式：概念、公式、实现

●【1】优化设置：基于批次数据的训练方法

○ 知识点：Batch 及 Batch Size 概念、Batch Size 设置对模型训练的影响

题目：

关于批量大小（Batch Size）对模型训练的影响，以下说法错误的是：

A. 较大的批量大小可以提高训练的并行效率

B. 较小的批量大小通常使梯度

估计更加准确

C. 批量大小的选择需要在训练速度和模型性能之间权衡

D. 过大的批量大小可能导致模型泛化能力下降

答案：B

解析：从优化理论的角度，从优化理论的角度，单次 mini-tch 梯度估计的期望等于真实梯度，与 batch size 无关。较大的批量大小能提供方差更小的梯度估计（因为平均了更多样本的梯度，统计上更稳定），而较小的批量大小梯度噪声更

大（方差大）。小 batch 的噪声有时反而有助于逃出局部最优、提升泛化能力。因此 B 选项中 " 更加准确 " 这一说法不严谨，是错误的。其他选项都正确：大批量提高并行效率（A）、需要权衡（C）、过大批量可能影响泛化（D）。

○ 扩展知识点：动态批次大小调整策略

题目：

某预训练任务初始批次大小（Batch Size）为 256，采用梯度累积（Gradient Accumulation）策略每 4 步更新一次，实际等效批次大小是多少？若改为动态调整策略，在训练中期将批次大小增至 512，请计算梯度累积步数应如何调整以保持等效批次大小不变？

A. 等效 1024，累积步数改为 2 步

B. 等效 512，累积步数改为 2 步

C. 等效 1024，累积步数改为 8 步

D. 等效 512，累积步数改为 8 步

答案：A

解析：等效批次大小 = 单次批次大小 × 梯度累积步数。初始情况：256 × 4 = 1024。当批次大小增至 512 时，为保持等效批次大小 1024 不变，需要调整累积步数：1024 ÷ 512 = 2 步。梯度累积是一种在显存受限时模拟大批次训练的技术，通过累积多个小批次的梯度后再更新参数。

○ 考核方式：概念、实现

●【2】优化设置：学习率

○ 知识点：学习率概念、学习率衰减（退火）、学习率预热（Warm-up）

题目：

在深度学习模型训练中，学习率预热（Learning Rate Warm-up）策略主要用于解决以下哪个问题？

A. 防止模型在训练早期由于学习率过大而发散

B. 解决梯度消失问题，使模型能收敛到更优解

C. 提高模型在训练后期的泛化能力

D. 增加训练过程的并行效率

答案：A

解析：

A 正确：模型在训练初期参数随机，使用一个小的学习率开始，并逐渐将其增加到预设的目标学习率，可以避免因初始梯度过大导致模型不稳定的问题。

B 错误：梯度消失通常通过激活函数选择、残差连接或更深的网络结构来解决。

C 错误：预热主要影响训练初期，后期泛化更多受正则化、数据等影响。

○ 扩展知识点：自适应学习率方法（如 Cosine 衰减）

题目：

Cosine 学习率衰减（Cosine Learning Rate Decay）策略是一种常用的学习率调度方法。相较于步进式衰减（Step Decay），其主要优点是：

A. 学习率衰减更为平滑，避免了学习率突变可能带来的训练波动

B. 学习率衰减速度恒定，易于计算

C. 仅适用于模型训练的最后阶段

D. 能够完全取代学习率预热的作用

答案：A

解析：

A 正确：Cosine 衰减在训练初期保持较高的学习率，然后平滑地降低到接近零，这种平滑的衰减过程有助于模型在训练后期稳定下来，找到更优的局部最小值。

B 错误：衰减速度不是恒定的，而是遵循 Cosine 函数的形状。

C 错误：它通常从一个较高的学习率开始，逐渐衰减，可以贯穿训练过程。

D 错误：Cosine 衰减和 Warm-up 是不同的概念，Warm-up 是“升”学习率，衰减是“降”学习率。

○ 考核方式：概念、实现

●【2】优化设置：优化器

○ 知识点：Adam、SGD 等

题目：

随机梯度下降（SGD）优化器与 Adam 优化器相比，通常的特点是：

A. Adam 优化器需要手动调整的学习率，而 SGD 具有自适应学习率

B. SGD 对学习率的初始值和衰减策略更敏感，但可能在某些情况下获得更好的泛化性能

C. Adam 优化器在计算上比 SGD 更高效，不需要存储额外的动量信息

D. SGD 能够更好地处理稀疏梯度问题

答案：B

解析：

B 正确：SGD 对超参数（学习率、衰减）非常敏感，但由于其梯度噪声，有时能跳出局部最优，获得更好的泛化。Adam 自带动量和二阶矩估计，通常收敛更快，对超参数不敏感，但可能泛化略差。

A 错误：SGD 不具备自适应学习率，Adam 才具备。

C 错误：Adam 需要存储动量和二阶矩，计算量比 SGD 大。

D 错误：Adam 的自适应性使其在稀疏梯度场景下表现往往更好。

○ 扩展知识点：优化器的改进版本（如 AdamW、LAMB）

题目：

关于 AdamW 优化器的改进，以下说法正确的是?

A. 移除了权重衰减与梯度更新的耦合关系

B. 增加了动量因子的计算复杂度

C. 取消了学习率自适应机制

D. 仅适用于计算机视觉任务

答案：A

解析：AdamW 是对 Adam 优化器的改进，其核心改进是将权重衰减（weight decay）从梯度更新中解耦出来，直接作用于参数本身，这使得正则化效果更加稳定有效。AdamW 保留了 Adam 的自适应学习率机制（C 错误），计算复杂度没有明显增加（B 错误），且广泛应用于各类深度学习任务（D 错误）。

○ 考核方式：概念、代码实现

●【2】参数量计算

○ 知识点：计算模型参数量的方法

题目：

一个简单的全连接层（Fully Connected Layer），其输入维度为 N，输出维度为 M。则该层有多少个可训练参数?

A. N*M

B. N+M

C. N*M+M

D. N*M+N

答案：C

解析：

C 正确：全连接层包含权重矩阵 W (N*MW 个参数) 和偏置向量 b (M 个参数)。

○ 扩展知识点：MoE 模型的参数量计算

题目：

对于一个包含 K 个专家（Experts）的稀疏激活（Sparse Activation）的 Mixture-of-Experts (MoE) 模型，假设所有专家的参数量相同，且在一次前向传播中，门控网络（Gating Network）会选择 M 个专家进行计算（M<K）。那么，在评估 MoE 模型的总参数量时，最主要的组成部分是:

A. 仅考虑门控网络的参数量

B. 仅考虑被激活的 M 个专家的参数量

C. 所有 K 个专家的参数量加上门控网络的参数量

D. K 个专家中参数量最大的那个专家的参数量

答案：C

解析：

C 正确：虽然每次只激活 M 个专家，但模型中所有 K 个专家都是可训练参数，因此总参数量需要计入所有专家以及门控网络的参数。

○ 考核方式：计算题

●【3】训练运算量及时间计算

○ 知识点：训练运算量及时间复杂度的计算方法

题目：

对于一个 Transformer Decoder-only 模型，在处理一个长度为 L 的序列时，其自注意力（Self-Attention）机制的计算复杂度（按乘加运算次数计）大致为：

A. O(L)

B. O(L^2)

C. O(L^3)

答案：B

解析：

B 正确：Transformer 的自注意力机制需要计算序列中任意两个 Token 之间的关联度，其计算复杂度与序列长度的平方成正比，即 O(L^2) 。

○ 扩展知识点：FLOPs 与训练时间的实际估算

题目：

一个大模型训练任务的理论计算量（FLOPs）估算为 10^24FLOPs。若使用的硬件峰值算力为 10^17FLOPS（每秒浮点运算次数），并且模型能够充分利用硬件的峰值算力。则完成该训练任务所需的最少时间（理论上）约为：

A.10^7s B.10^8s C. 10^9s D. 10^10s

答案：A

解析：

所需时间 = 总计算量 / 实际可用算力。

○ 考核方式：计算题

●【3】训练显存计算

○ 知识点：预训练显存计算方法

题目：

假设一个大模型的训练中，模型参数量为 P，梯度大小为 G，优化器状态（如 Adam 的动量和二阶矩）大小为 O。则单次

前向 / 反向传播（不含优化器状态）所需的显存（约等于）主要由以下几项组成：

A. P+G

B. P+O

C. P

D. P+G+O

答案：A

解析：

A 正确：在单次前向和反向传播过程中，至少需要存储模型参数（P）和计算过程中产生的梯度（G）。优化器的状态（O）通常在参数更新时使用，不一定在每次前向 / 反向计算时都需要全部加载到显存（但如果考虑整个 batch 的训练，则需要）。在不考虑激活值、缓存等的情况下，参数和梯度是基础。

○ 扩展知识点：显存优化技术（如梯度检查点）

题目：

梯度检查点（Gradient Checkpointing）技术在训练大模型时，其主要的权衡是：

A. 减少模型参数量，以降低计算需求

B. 增加模型参数量，以提升训练速度

C. 牺牲部分计算时间（增加重计算量），以显著减少显存占用

D. 提高计算精度，以避免数值不稳定

答案：C

解析：

C 正确：梯度检查点通过在反向传播时重新计算一部分中间激活值，而不是全部存储它们，来大幅减少显存占用。这会增加计算量，但对于内存受限的训练场景至关重要。

○ 考核方式：计算题

●【3】稳定优化技术

○ 知识点：梯度裁剪、训练恢复、权重衰减等

题目：

在深度学习模型训练过程中，梯度裁剪（Gradient Clipping）技术的主要作用是什么？

A. 加速模型收敛速度，提高训练效率

B. 防止梯度爆炸，维持训练稳定性

C. 减少模型参数量，降低计算复杂度

D. 增强模型泛化能力，防止过拟合

答案：B

解析：梯度裁剪通过限制梯度的范数或值，防止在训练过程

中出现梯度爆炸问题，这在训练循环神经网络和深层网络时尤为重要。它不直接加速收敛（A 错误）、不减少参数量（C 错误）、主要目的也不是防止过拟合（D 错误），而是确保训练过程的数值稳定性。

○ 扩展知识点：混合精度训练中的稳定性问题

题目：

在深度学习模型训练中，混合精度训练（Mixed Precision Training）技术的主要作用是什么？

A. 通过使用单精度浮点（float32）主副本维护参数精度，结合半精度浮点（float16）计算提升训练速度，并采用损失缩放避免梯度下溢

B. 完全消除半精度浮点的数值精度损失，使其与单精度浮点等效

C. 将半精度浮点数据转换为整型进行计算，以进一步加速训练

D. 通过增加模型参数量来补偿半精度浮点带来的精度损失

答案：A

解析： 混合精度训练是一种平衡训练速度和数值精度的技术。它用 float16 进行前向和反向传播计算以加速训练，同时保留 float32 的主参数副本以维持精度；并使用损失缩放（loss scaling）技术防止小梯度下溢。这种方法不能完全消除精度损失（B 错误），不涉及整型计算（C 错误），也不增加参数量（D 错误）。

○ 考核方式：概念

●【3】并行训练

○ 知识点：3D 并行（数据并行、流水线并行、张量并行）

题目：

在分布式训练中，数据并行（Data Parallelism）的主要思想是：

A. 将模型的不同层放置在不同的计算设备上，实现流水线式计算

B. 将模型的单个层（如一个大矩阵乘法）切分到多个设备上并行计算

C. 将训练数据分发到多个计算设备，每个设备都拥有完整的模型副本，并处理部分数据

D. 同时利用数据并行、流水线并行和张量并行来加速训练

答案：C

解析：

C 正确：数据并行是最大化模型训练效率的常用方法，即复制模型，分发数据。A 描述的是流水线并行，B 描述的是张量并行。D 是混合并行。

○ 扩展知识点：ZeRO 优化器与混合并行策略

题目：

ZeRO（Zero Redundancy Optimizer）优化器，特别是其后期的版本（如 ZeRO-2, ZeRO-3），主要目标是在数据并行训练的基础上，进一步减少什么方面的冗余，从而大幅降低显存占用？

A. 模型参数的冗余

B. 梯度的冗余

C. 优化器状态的冗余

D. 以上各项（通过对模型状态进行分片）

答案：D

解析：

D 正确：ZeRO 通过将模型参数、梯度和优化器状态进行分片（sharding），并只在需要时才进行通信或聚合，从而在数据并行中消除了不同 GPU 之间冗余存储的参数、梯度和优化器状态，极大地降低了显存占用。

○ 考核方式：概念

●【3】训练精度

○ 知识点：单精度、半精度、混合精度概念，及其对模型训练的影响

题目：

与单精度浮点（FP32）相比，半精度浮点（FP16）在训练大模型时带来的主要优势是：

A. 提高模型训练的最终精度，使其超越 FP32

B. 显著减少显存占用和提高计算速度

C. 能够无损地表示更大的数值范围

D. 简化了模型的反向传播过程

答案：B

解析：

B 正确：FP16 格式比 FP32 占用一半的存储空间，并且许多现代硬件（如 NVIDIA Tensor Cores）可以加速 FP16 计算，从而提高训练速度。

A 错误：FP16 的表示范围和精度低于 FP32，可能导致训练精度下降或数值不稳定。

C 错误：FP16 的数值范围比 FP32 小。

D 错误：反向传播过程本身并未简化，但数值计算可能更快。

○ 扩展知识点：BF16 与 FP16 的对比

题目：

在混合精度训练中，BF16（Bfloat16）和 FP16（Half

Precision Float）相比，BF16 的主要特点是：

A. 拥有与 FP32 相同的指数位宽度，因此数值范围更广，但精度较低

B. 拥有与 FP32 相同的尾数位宽度，因此数值精度更高，但范围较窄

C. 具有比 FP32 更小的指数位宽度和尾数位宽度

D. 无法用于训练深度学习模型

答案：A

解析：

A 正确：BF16 (Brain Floating Point) 格式与 FP32 一样，都拥有 8 位指数（Exponent）和 23 位尾数（Mantissa）。BF16 格式与 FP32 有相同的指数位（8 位），这意味着它能表示的数值范围与 FP32 几乎相同，这有助于避免溢出和下溢问题；但它只有 7 位尾数，相比 FP32 的 23 位尾数，其数值精度较低。FP16 则有 5 位指数和 10 位尾数。

○ 考核方式：概念

●【3】高效训练技术

○ 知识点：完全分片数据并行、完全和选择性激活重计算、混合精度训练的流程、融合算子的原理

题目：

完全分片数据并行（Fully Sharded Data Parallelism, FSDP）与传统数据并行（Data Parallelism）相比，其核心优势在于：

A. FSDP 将模型参数、梯度和优化器状态都进行了分片，极大地减少了每个 GPU 的显存占用

B. FSDP 只需要一个 GPU 即可完成大规模模型的训练

C. FSDP 能够自动选择最优的模型架构，无需人工干预

D. FSDP 主要用于加速模型的推理速度，而非训练速度

答案：A

解析：

A 正确：FSDP（如 ZeRO-3 的实现）是数据并行的一种高级形式，它将模型状态（参数、梯度、优化器状态）在所有数据并行副本之间进行分片，从而使每个 GPU 只需要存储模型状态的一小部分，显著降低了显存需求。

○ 扩展知识点：分布式训练中的通信优化

题目：

在分布式训练中，通信开销（Communication Overhead）是影响训练速度的关键瓶颈之一。以下哪项技术不是主要用于减少通信开销的？

A. Gradient Checkpointing（梯度检查点）

B. All-Reduce 算法优化（如 Ring All-Reduce）

C. ZeRO 优化器（特别是 ZeRO-2/3）

D. Token Compression（词元压缩）

答案：A

解析：

A 错误：Gradient Checkpointing 主要目的是减少显存占用，通过重计算来换取显存，它不会减少通信开销，反而可能因为引入额外的计算图而间接增加一些同步开销。

B、C、D 都是通信优化的范畴：All-Reduce 是高效的聚合梯度通信算法；ZeRO 通过分片减少了需要传输的梯度和参数量；Token Compression 理论上可以减少需要传输的 token 数量（如果是在模型间传输某些中间表示）。

○ 考核方式：概念、计算

指令微调

●【1】模型解码

○ 知识点：概率分布、采样

题目：

关于语言模型中的概率分布与采样，下列说法正确的是：

A. 语言模型在每一步生成时，会输出词表上的一个概率分布，表示每个词被选为下一个词的概率

B. 采样是指始终选择概率最大的那个词作为输出

C. 语言模型输出的 logits 本身就是概率分布，不需要经过任何变换

D. 概率分布中所有词的概率之和可以大于 1

答案：A

解析：

A 正确：语言模型的输出层经过 softmax 后形成词表上的概率分布

B 错误：始终选最大概率的是贪心搜索，采样是按概率随机选取

C 错误：logits 需要经过 softmax 函数变换后才成为概率分布

D 错误：经过 softmax 后的概率分布所有值之和恒等于 1

○ 扩展知识点：给定分布下的概率采样

题目：

关于给定概率分布下的采样方法，下列描述正确的是：

A. 从概率分布中采样时，概率越高的词被选中的可能性越大，但不保证一定被选中

B. 概率采样会使每次生成的结果完全相同

C. 从均匀分布中采样和从语言模型输出分布中采样的效果完全一样

D. 采样过程不涉及任何随机性

答案：A

解析：

A 正确：采样是按概率随机选取，高概率词更可能被选中但非必然

B 错误：采样具有随机性，每次结果可能不同

C 错误：均匀分布对所有词等概率，而模型输出分布有偏好，效果差异很大

D 错误：采样的核心就是引入随机性

○ 考核方式：概念

●【1】指令微调

○ 知识点：指令数据、指令遵循、指令微调的概念

题目：

关于指令微调的核心概念，下列说法正确的是：

A. 指令微调是在预训练模型的基础上，使用指令 - 响应数据对进行训练，使模型更好地遵循人类指令

B. 指令微调是指从零开始训练一个新的语言模型

C. 指令数据只包含输入文本，不需要对应的期望输出

D. 指令微调会完全替换预训练阶段学到的知识

答案：A

解析：

A 正确：指令微调在预训练基础上用 " 指令 + 期望响应 " 数据对进行微调，提升指令遵循能力

B 错误：指令微调是在已有预训练模型基础上进行的，不是从零训练

C 错误：指令数据通常包含指令输入和对应的期望输出

D 错误：指令微调是在保留预训练知识的前提下增强指令遵循能力

○ 扩展知识点：指令微调与多任务学习的关系

题目：

关于指令微调与多任务学习的关系，下列描述正确的是：

A. 指令微调可以看作一种多任务学习，通过统一的指令格式让模型同时学习多种任务

B. 指令微调只能用于单一任务，不能同时处理多种任务

C. 多任务学习和指令微调完全没有关联

D. 指令微调中使用的多种任务数据会导致模型性能必然下降

答案：A

解析：

A 正确：指令微调通过统一的 " 指令 - 响应 " 格式整合多种任务数据，本质上是多任务学习

B 错误：指令微调的核心优势之一就是可以通过多任务数据提升泛化能力

C 错误：指令微调与多任务学习高度相关，前者可视为后者的特殊形式

D 错误：多任务数据通常能提升模型泛化能力，不会必然导致性能下降

○ 考核方式：概念

●【1】指令数据集的构建

○ 知识点：指令数据合成方法

题目：

关于指令数据的合成方法，下列说法正确的是：

A. Self-Instruct 方法利用大语言模型自身来生成新的指令数据，通过少量种子指令引导模型生成更多指令

B. 指令数据只能通过人工标注获得，不能使用任何自动化方法

C. 合成的指令数据质量一定高于人工标注的数据

D. 指令数据合成不需要任何种子数据或示例

答案：A

解析：

A 正确：Self-Instruct 是经典的指令数据合成方法，用少量种子指令引导 LLM 生成新指令

B 错误：Self-Instruct、Evol-Instruct 等方法都是自动化合成指令数据的方式

C 错误：合成数据质量不一定高于人工标注，需要额外筛选和清洗

D 错误：Self-Instruct 等方法通常需要种子指令作为启动数据

○ 扩展知识点：高质量指令数据的筛选与清洗

题目：

关于高质量指令数据的筛选与清洗，下列描述正确的是：

A. 高质量指令数据的筛选主要关注指令与响应的语义一致性、多样性和准确性

B. 所有自动生成的指令数据都是高质量的，不需要筛选

C. 指令数据的质量只取决于指令的长度

D. 数据清洗只需要去除重复数据即可

答案：A

解析：

A 正确：高质量数据需要语义匹配、多样性和准确性多方面保障

B 错误：自动生成的数据中可能存在低质量、重复或不准确的内容

C 错误：质量取决于语义一致性、准确性、多样性等多个维度，不仅是长度

D 错误：数据清洗还包括过滤低质量响应、纠正错误标注、去除有害内容等

○ 考核方式：概念、实现

●【2】指令数据构建提升方法

○ 知识点：指令进化算法、自引导指令增强（Bootstrapping）、长上下文指令构建

题目：

关于指令数据构建的提升方法，下列说法正确的是：

A. 指令进化算法（如 Evol-Instruct）通过对已有指令进行复杂化改写或增加约束来生成更复杂的指令

B. 自引导指令增强（Bootstrapping）不使用模型自身生成的数据

C. 长上下文指令构建只需要将多条短指令简单拼接即可

D. 指令进化算法生成的指令复杂度不会超过原始种子指令

答案：A

解析：

A 正确：Evol-Instruct 通过深度进化和广度进化策略逐步提升指令的复杂度和多样性

B 错误：Bootstrapping 正是利用模型自身输出作为反馈来增强指令数据

C 错误：长上下文指令需要精心设计上下文结构，不是简单拼接

D 错误：进化算法的目的就是逐步增加指令复杂度，会超过原始种子

○ 扩展知识点：指令数据质量与泛化能力的关系

题目：

关于指令数据质量与模型泛化能力的关系，下列描述正确的是：

A. 高质量、多样化的指令数据有助于提升模型在未见任务上的泛化能力

B. 指令数据的数量越多，模型泛化能力一定越强，与质量无关

C. 单一领域的指令数据足以让模型泛化到所有领域

D. 指令数据的质量对模型泛化能力没有显著影响

答案：A

解析：

A 正确：数据的质量和多样性是影响泛化能力的关键因素

B 错误：大量低质量数据可能反而损害模型性能，质量比数量更重要

C 错误：单一领域数据会导致模型在其他领域表现不佳

D 错误：指令数据质量对泛化能力有显著影响

○ 考核方式：概念、实现

●【2】微调优化设置

○ 知识点：目标函数、批次大小、多指令合并高效训练、多阶段混合训练（长短指令、数据课程）

题目：

关于指令微调的优化设置，下列说法正确的是：

A. 指令微调通常使用交叉熵损失函数作为目标函数，仅对响应部分计算损失

B. 批次大小越大模型效果一定越好，不需要任何权衡

C. 多指令合并训练是指将多条指令的响应混在一起作为一条训练数据

D. 多阶段混合训练中各阶段使用完全相同的数据配比

答案：A

解析：

A 正确：指令微调通常只对模型生成的响应部分计算损失，指令部分不参与损失计算

B 错误：过大的批次可能影响泛化能力，需要在效率和效果之间权衡

C 错误：多指令合并是将多条指令打包到同一个序列中高效训练，但每条指令的损失独立计算

D 错误：多阶段训练的核心是不同阶段使用不同的数据配比和课程安排

○ 扩展知识点：指令微调中的资源消耗估算方法

题目：

关于指令微调中的资源消耗估算，下列描述正确的是：

A. 模型微调所需的显存主要包括模型参数、梯度、优化器状态和激活值四个部分

B. 微调 7B 参数的模型和 70B 参数的模型所需的显存完全相同

C. 增加批次大小不会影响显存消耗

D. 使用全参数微调和使用 LoRA 微调所需的显存完全一致

答案：A

解析：

A 正确：显存消耗的四大组成部分是参数、梯度、优化器状

态和激活值

B 错误：参数量越大，显存消耗越大，70B 比 7B 需要多得多的显存

C 错误：更大的批次会增加激活值的显存占用

D 错误：LoRA 只训练少量参数，显存消耗远低于全参数微调

○ 考核方式：概念、实现

●【2】数据组织策略

○ 知识点：领域专家模型的指令过滤、代理模型引导的指令配比、基于导数的指令数据选择

题目：

关于指令微调中的数据组织策略，下列说法正确的是：

A. 领域专家模型的指令过滤是指利用领域专用模型来评估和筛选指令数据的质量与相关性

B. 代理模型引导的指令配比是指让所有领域的数据均匀分配，不做任何调整

C. 基于导数的指令数据选择与梯度信息无关

D. 数据组织策略对微调效果没有影响，只需要随机打乱数据即可

答案：A

解析：

A 正确：利用领域专家模型评估指令数据与目标领域的相关性和质量

B 错误：代理模型引导的配比是根据模型反馈动态调整各领域数据的比例

C 错误：基于导数的方法正是利用梯度信息来选择对模型最有益的训练数据

D 错误：数据组织策略对微调效果有显著影响

○ 扩展知识点：垂直领域的指令数据构造与使用

题目：

关于垂直领域的指令数据构造与使用，下列描述正确的是：

A. 垂直领域的指令数据需要结合领域专业知识来构建，以确保数据的准确性和专业性

B. 通用领域的指令数据可以完全替代垂直领域的专业数据

C. 垂直领域微调不需要考虑数据质量，只要数量足够多即可

D. 垂直领域微调后模型会完全丧失通用能力

答案：A

解析：

A 正确：垂直领域数据需要领域专家参与构建，确保专业性和准确性

B 错误：通用数据缺乏领域专业性，无法完全替代垂直领域

数据

C 错误：垂直领域数据的质量尤为重要，低质量数据可能引入领域错误

D 错误：合理的微调策略可以在保持通用能力的同时增强领域能力

○ 考核方式：概念、实现

●【3】参数高效微调方法

○ 知识点：低秩适配（LoRA）、适配器微调、前缀微调、提示微调

题目：

以下代码实现了 LoRA（低秩适配）微调层，请阅读代码并填写空缺部分。

```
import torch
import torch.nn as nn
import math

class LoRALinear(nn.Module):
""" 在原始线性层旁添加低秩分解矩阵实现 LoRA 微调 """
def __init__(self, original_linear: nn.Linear, rank: int = 8, alpha: float = 16.0):
super().__init__()
self.original_linear = original_linear
self.rank = rank
self.alpha = alpha
self.scaling = alpha / rank

in_features = original_linear.in_features
out_features = original_linear.out_features

# [1] 初始化低秩矩阵 A 和 B
# LoRA 的核心 : ΔW = A @ B, 其中 A: (in_features, rank), B: (rank, out_
features)
[1] _____
[2] _____
nn.init.kaiming_uniform_(self.lora_A, a=math.sqrt(5))
nn.init.zeros_(self.lora_B)

# 冻结原始权重
self.original_linear.weight.requires_grad = False
if self.original_linear.bias is not None:
self.original_linear.bias.requires_grad = False

def forward(self, x):
# 原始线性层的输出
original_output = self.original_linear(x)
# [2] 计算 LoRA 的增量输出 : x 先经过 A 再经过 B，并乘以缩放因子
```

[1] 和 [2] 处应分别填入：

A. [1] self.lora_A = nn.Parameter(torch.empty(in_features, rank)) 和 self.lora_B = nn.Parameter(torch.empty(rank, out_features)) [2] lora_output = (x @ self.lora_A @ self.lora_B) * self.scaling

B. [1] self.lora_A = nn.Parameter(torch.empty(out_features, rank)) 和 self.lora_B = nn.Parameter(torch.empty(rank, in_features)) [2] lora_output = (x @ self.lora_A @ self.lora_B) * self.scaling

C. [1] self.lora_A = nn.Parameter(torch.empty(in_features, rank)) 和 self.lora_B = nn.Parameter(torch.empty(rank, out_

features)) [2] lora_output = (x + self.lora_A + self.lora_B) * self.scaling

D. [1] self.lora_A = torch.empty(in_features, rank) 和 self.lora_B = torch.empty(rank, out_features) [2] lora_output = (x @ self.lora_A @ self.lora_B) * self.scaling

答案： A

解析：

A 正确：LoRA 将 ΔW 分解为 A(in,rank)×B(rank,out)，前向计算为 x@A@B 乘以缩放因子，且 A 和 B 必须是 nn.Parameter 可学习参数

B 错误：矩阵 A 的维度应为 (in_features, rank)，B 为 (rank, out_features)，B 选项维度颠倒了

C 错误：LoRA 的增量输出是矩阵乘法 x@A@B，不是加法 x+A+B

D 错误：lora_A 和 lora_B 必须用 nn.Parameter 包裹才能被优化器识别为可训练参数，普通 tensor 不参与梯度更新

○ 扩展知识点：参数高效微调的理论基础

题目：

关于参数高效微调的理论基础，下列描述正确的是：

A. LoRA 的理论基础是预训练模型的权重更新具有低秩特性，即微调所需的权重变化可以用低维子空间来表示

B. 参数高效微调方法的效果一定远差于全参数微调

C. 低秩分解中的秩（rank）越大，模型效果一定越好

D. 参数高效微调不能应用于大语言模型

答案： A

解析：

A 正确: 研究表明微调过程中的权重更新矩阵具有低秩特性，这是 LoRA 的理论基础

B 错误：在很多任务上参数高效微调可以达到接近全参数微调的效果

C 错误：秩过大可能导致过拟合，需要选择合适的秩

D 错误：参数高效微调正是大语言模型时代的主流微调方式

○ 考核方式：概念、公式、实现

人类对齐

●【1】人类对齐的背景与标准

○ 知识点：人类对齐背景、人类对齐标准（如无害性、有用性、诚实性）

题目：

除了无害性、有用性和诚实性，人类对齐还可能包含哪些重要标准?

A. 语言表达的准确性和文化适应性

B. 符合特定社会背景下的道德规范

C. 尊重用户隐私和数据安全

D. 其它所有选项

答案：D

解析：人类对齐是一个多维度的概念，除了基本的无害性、有用性和诚实性（3H 原则），还包括语言和文化适应性、道德规范、隐私保护等多个方面。不同的应用场景和文化背景可能对对齐标准有不同要求，因此 A、B、C 都是重要的对齐标准。

○ 扩展知识点：其他对齐标准（语言表达、道德标准等）

题目：

在大模型对齐的实践中，除了核心的“3H”原则，还需要考虑语言表达与道德标准。关于这些扩展标准，下列说法错误的是：

A. 文化适应性：模型应能理解并尊重不同地区的文化禁忌与习俗

B. 语言风格对齐：模型生成的回复应符合特定场景下的语态要求（如专业报告需严谨，客服回复需亲切）

C. 道德绝对性：对齐的目标是让模型遵循一套全球统一、永恒不变的绝对道德准则

D. 合规性：模型的输出必须符合所在国家的法律法规及互联网内容管理规定

答案：C

解析：

C 错误：道德标准往往具有地域性、时代性和文化相关性，不存在一套全球完全统一且永恒不变的道德准则。对齐工作需要根据目标用户群体和法律环境动态调整。A、B、D 均属于对齐过程中需要考虑的多维度标准。

○ 考核方式：概念

●【2】人类偏好与反馈数据收集

○ 知识点：人类反馈收集方法、基于评分的人类反馈、基于排序的人类反馈

题目：

在收集人类偏好数据以训练奖励模型时，相比于直接给单个回答打分（Scoring），采用“两两比较并排序（Ranking/Pairwise Comparison）”的主要优势是：

A. 排序数据的标注速度比打分快得多

B. 不同标注者对分数范围的尺度（Scaling）理解往往不一致，而排序能降低这种主观偏差

C. 排序数据可以直接用于预训练阶段

D. 排序不需要标注人员具备专业背景

答案：B

解析：

打分存在“锚定效应”，不同人对“4 分”的定义不同。而让标注者在两个回答中选出更好的一个，更容易达成共识，数据一致性更高，这也是目前 RLHF 流程中的主流做法。

○ 扩展知识点：反馈数据的偏差与修正

题目：

人类反馈数据中常存在“长度偏见（Length Bias）”，这指的是：

A. 标注者倾向于认为较短的回答更简洁高效

B. 奖励模型容易给篇幅较长、看起来更“不明觉厉”的回答打高分，即使其实质内容可能较差

C. 模型生成的回答长度超过了 Token 限制

D. 标注人员只愿意阅读短文本

答案：B

解析：

研究发现，人类和奖励模型往往会无意识地偏好更长、辞藻更华丽的回答，这种偏差需要在数据采样和模型训练中进行修正。

○ 考核方式：概念

●【2】非强化学习训练的对齐方法

○ 知识点：DPO 的公式及原理

题目：

DPO 算法在对齐大模型时，相比于传统的 RLHF（基于 PPO 算法），其最核心的改变是：

A. 引入了更强大的外部奖励模型

B. 放弃了使用强化学习优化器，通过数学变换直接在偏好数据上通过分类损失函数优化策略模型

C. 增加了模型对图像数据的处理能力

D. 仅在预训练阶段使用，不需要微调

答案：B

解析：

DPO 将奖励函数与最优策略的关系进行了闭式解变换，使得我们可以直接利用偏好数据进行监督学习（通常是交叉熵损失），从而避开了复杂的强化学习（PPO）框架。

○ 扩展知识点：DPO 的算法推导、DPO 与 RLHF 的对比、DPO 模型的变种（token-level DPO 和 reference-free DPO 算法）

题目：

关于 DPO 的变种算法，下列描述正确的是：

A. Token-level DPO 旨在对句子中的每一个词元（Token）进行独立的奖励赋值，以实现更精细的信用分配

B. Reference-free DPO 彻底取消了 KL 散度约束，使得模型可以无限偏离原始分布

C. Reference-free DPO 主要是为了通过移除参考模型（Reference Model）来降低训练时的显存占用

D. Token-level DPO 只能应用于图像生成任务，不适用于文本生成

答案： A（注：C 也有一定道理，但 A 更符合算法设计的本意）

解析：

A 正确：标准的 DPO 是句子级的（Sequence-level），Token-level DPO 试图解决句子内部不同部分贡献度不同的问题。

C 不准确：虽然移除 Ref 模型能省显存，但 Reference-free DPO（如某些对齐研究中尝试将 Ref 项设为常数或使用隐含先验）的主要挑战在于失去正则化后容易导致模型崩溃或过

度拟合偏好数据，通常不被视为标准生产推荐做法。

B 错误：完全取消约束会导致训练崩溃。

○ 考核方式：概念、公式、实现

●【2】奖励模型训练

○ 知识点：打分式、对比式、排序式奖励模型训练损失

题目：

关于打分式、对比式（两两比较）和排序式（列表排序）损失的逻辑，以下说法正确的是：

A. 打分式损失要求模型对每个回答给出一个绝对分数，它能有效避免不同标注者之间因为“评分标准不一”带来的噪声

B. 对比式损失通过让模型学习“回答 A 优于回答 B”的相对关系来更新参数，是目前主流 RLHF 流程中处理两两比较数据的核心逻辑

C. 排序式损失只能处理两个回答的先后顺序，当一次性出现三个或更多备选回答时，该逻辑将失效

D. 这三种损失函数在训练时的目标完全一致，即要求模型必须准确预测出人类给出的具体分值

答案： B

解析：

A 错误：打分式（Pointwise）最大的问题在于主观标准不统

一（例如：标注者 A 认为的 4 分可能是标注者 B 认为的 3 分），会引入大量噪声。

B 正确：对比式（Pairwise）损失不要求模型学习绝对分值，而是学习相对偏好序。这符合人类标注的习惯（两两相比更容易达成共识），是当前大模型奖励模型训练的主流方案。

C 错误：排序式（Listwise/Ranking）损失可以处理包含多个回答的列表，通过优化整个序列的排列顺序来训练模型，并不局限于两个回答。

D 错误：它们的目标逻辑不同。打分式关注绝对值预测；对比式和排序式关注相对序的准确性，不强制要求输出特定的分数值。

○ 扩展知识点：奖励模型的泛化能力

题目：

奖励模型训练中存在“奖励欺骗（Reward Hacking）”现象，这反映了奖励模型在泛化能力上的什么局限性？

A. 奖励模型无法识别过短的回答

B. 策略模型找到了奖励模型的漏洞，通过生成符合奖励模型偏好但实际质量低劣（如过度拟合格式）的内容来骗取高分

C. 奖励模型在测试集上的准确率高于训练集

D. 奖励模型只能处理与其训练数据完全一致的分布

答案： B

解析：

B 正确：奖励欺骗是泛化失败的典型表现。当奖励模型无法完美捕捉人类真实意图时，策略模型（Generator）会利用奖励模型的弱点（例如：RM 偏好结尾加“谢谢”，模型就在所有回答后加“谢谢”而不顾质量）来刷分。

○ 考核方式：概念、公式、实现

● 【3】幻象

○ 知识点：幻象问题的概念与分类、幻象的起因

题目：

大模型的幻觉可以分为“内在幻觉”和“外在幻觉”。关于“外在幻觉（Extrinsic Hallucination）”，下列描述准确的是：

A. 模型生成的回答与用户提供的上下文信息相矛盾

B. 模型生成的回答在逻辑上自相矛盾

C. 模型生成的回答包含在训练数据或输入上下文中无法验证的、虚构的事实性信息

D. 模型生成的回答语言风格不符合人类习惯

答案： C

解析：

内在幻觉指逻辑矛盾或与上下文冲突；外在幻觉指模型输出

了无法从输入或已知事实中证实的内容（即凭空捏造事实）。

○ 扩展知识点：幻象的常见缓解方法

题目：

在实际应用中，为了缓解大模型的幻觉问题，以下哪种技术手段被证明是最直接有效的“外挂知识库”方案？

A. 增加模型的参数量

B. 检索增强生成（RAG, Retrieval-Augmented Generation）

C. 增加训练数据的多样性

D. 延长思维链（Chain of Thought）的步数

答案：B

解析：

RAG 通过在生成前从外部可靠知识库检索相关文档并作为上下文输入给模型，显著降低了模型因知识遗忘或缺失而产生的事实性幻觉。CoT 主要解决推理逻辑问题，不能直接提供模型未学过的实时事实。

○ 考核方式：概念、实现

解码与部署

●【1】解码方法

○ 知识点：贪心搜索、束搜索解码的概念

题目：

关于语言模型的解码方法，下列说法正确的是：

A. 贪心搜索在每一步选择概率最高的词，简单高效但可能错过全局最优序列

B. 束搜索和贪心搜索完全相同，都只保留一个候选序列

C. 贪心搜索能保证找到全局最优的生成序列

D. 束搜索不需要设置任何超参数

答案：A

解析：

A 正确：贪心搜索每步选最大概率词，效率高但是局部最优，可能错过全局更优序列

B 错误：束搜索同时维护多个候选序列（束宽个），贪心搜索只保留 1 个

C 错误：贪心搜索是局部最优策略，不能保证全局最优

D 错误：束搜索需要设置束宽（beam width）等超参数

○ 扩展知识点：束搜索的超参数调优

题目：

关于束搜索的超参数调优，下列描述正确的是：

A. 增大束宽（beam width）可以探索更多候选序列，但会增

加计算开销

B. 束宽为 1 时的束搜索与贪心搜索效果不同

C. 束宽越大生成质量一定越高，因此应该设置尽可能大的束宽

D. 束搜索中的长度惩罚参数对生成结果没有影响

答案： A

解析：

A 正确：更大的束宽意味着更多候选路径，搜索更全面但计算量线性增长

B 错误：束宽为 1 时束搜索退化为贪心搜索，两者等价

C 错误：束宽过大可能导致生成结果过于保守和重复，且计算开销过大

D 错误：长度惩罚参数会影响模型对不同长度序列的偏好

○ 考核方式：概念、公式

●【2】随机采样及改进策略

○ 知识点：温度采样、top-k 采样、top-p 采样

题目：

以下代码实现了温度采样、Top-K 采样和 Top-P 采样，请阅读代码并填写空缺部分。

```
import torch
import torch.nn.functional as F

def sample_with_strategies(logits, temperature=1.0, top_k=0, top_p=0.0):
"""
对模型输出的 logits 应用温度、Top-K、Top-P 采样策略
参数 :
logits: 模型输出的原始分数 , shape (vocab_size,)
temperature: 温度参数，控制分布的平滑程度
top_k: 只保留概率最高的 k 个 token（0 表示不启用）
top_p: 只保留累积概率达到 p 的最少 token 集合（0.0 表示不启用）
返回 :
采样得到的 token 索引
"""
# [1] 温度缩放：用 temperature 对 logits 进行缩放
_____

# Top-K 采样：只保留概率最高的 k 个 token
if top_k > 0:
top_k_values, _ = torch.topk(scaled_logits, top_k)
min_top_k = top_k_values[-1]
scaled_logits = scaled_logits.masked_fill(scaled_logits < min_top_k, float('-inf'))

# Top-P (Nucleus) 采样
if top_p > 0.0:
sorted_logits, sorted_indices = torch.sort(scaled_logits, descending=True)
probs_sorted = F.softmax(sorted_logits, dim=-1)
cumulative_probs = torch.cumsum(probs_sorted, dim=-1)
# [2] 创建掩码：移除累积概率超过 top_p 的 token，但保留第一个超过阈值的
token
```

[1] 和 [2] 处应分别填入：

A. [1] scaled_logits = logits / temperature [2] sorted_indices_to_remove = cumulative_probs > top_p

B. [1] scaled_logits = logits * temperature [2] sorted_indices_to_remove = cumulative_probs > top_p

C. [1] scaled_logits = logits / temperature [2] sorted_indices_to_remove = cumulative_probs < top_p

D. [1] scaled_logits = logits - temperature [2] sorted_indices_to_remove = probs_sorted > top_p

答案：A

解析：

A 正确：温度采样的公式为 logits/T，T 越大分布越平滑（更随机），T 越小越尖锐（更确定）；Top-P 先用 cumulative_probs > top_p 标记超过阈值的位置，再将掩码右移一位，从而保留第一个超过阈值的 token，得到“累计概率刚好覆盖 top_p”的最小候选集合

B 错误：温度应该是除法（logits/T）而非乘法，乘以温度会使高温时分布更尖锐，与期望相反

C 错误：Top-P 应移除累积概率超过 p 的 token（>p），而非低于 p 的（<p）

D 错误：温度缩放是除法操作，不是减法；且应基于累积概率而非单个 token 概率来过滤

○ 扩展知识点：采样策略对生成多样性的影响

题目：

关于不同采样策略对生成多样性的影响，下列描述正确的是：

A. top-p 采样（核采样）根据累积概率动态调整候选词数量，当概率集中时自动减少候选词，分散时自动增加

B. 使用 top-k=1 的采样与使用高温采样的效果相同

C. 同时使用 top-k 和 top-p 策略是不可能的

D. 采样策略只影响生成速度，不影响生成质量和多样性

答案：A

解析：

A 正确：top-p 动态选择累积概率达到 p 的最少词集合，自适应调整候选词数量

B 错误：top-k=1 等价于贪心搜索（最确定），高温采样输出更随机

C 错误：实践中经常同时使用 top-k 和 top-p 来联合控制采样范围

D 错误：采样策略直接影响生成文本的多样性、质量和创造性

○ 考核方式：概念、公式、实现

●【3】解码加速算法与实践

○ 知识点：全量解码与增量解码、解码效率定量评估指标、常见推理工具使用（vLLM）

题目：

关于大语言模型的解码方式和推理工具，下列说法正确的是:

A. 增量解码（KV Cache）通过缓存已计算的 Key 和 Value，避免重复计算，从而加速自回归生成

B. 全量解码每一步都重新计算所有位置的注意力，效率高于增量解码

C. vLLM 是一个专门用于训练模型的工具，不用于推理加速

D. 解码效率的评估只看生成文本的质量，不考虑吞吐量和延迟

答案：A

解析：

A 正确：KV Cache 缓存历史的 Key 和 Value，每步只需计算新 token 的注意力，大幅减少计算量

B 错误：全量解码每步重复计算所有位置，效率远低于增量解码

C 错误：vLLM 是高效的 LLM 推理引擎，通过 PagedAttention 等技术加速推理

D 错误: 解码效率评估包括吞吐量（tokens/s）、延迟（latency）、首 token 时间（TTFT）等指标

○ 扩展知识点：解码加速优化算法（推测解码、非自回归解码、早退机制、级联解码）、解码加速的系统级优化（FlashAttention、PagedAttention、批次管理优化）

题目：

关于解码加速的优化算法和系统级优化，下列描述正确的是:

A. 推测解码（Speculative Decoding）使用一个小模型快速生成候选 token，再由大模型并行验证，从而加速生成

B. FlashAttention 通过降低注意力计算精度来加速，会显著损失模型精度

C. PagedAttention 是一种新型的注意力计算公式，与显存管理无关

D. 非自回归解码比自回归解码更慢，因为它需要逐个生成 token

答案：A

解析：

A 正确：推测解码用小模型草拟，大模型批量验证，在不改变输出分布的前提下加速生成

B 错误：FlashAttention 是 IO 感知的精确注意力实现，不损失精度，通过减少 HBM 读写来加速

C 错误：PagedAttention 借鉴操作系统分页思想管理 KV 缓存显存，减少显存碎片

D 错误：非自回归解码可以并行生成多个 token，通常比自

回归解码更快

○ 考核方式：概念

●【3】低资源部署策略

○ 知识点：量化基本概念、对称量化、非对称量化、量化粒度、常见量化方法、量化对模型性能的影响

题目：

关于模型量化的基本概念，下列说法正确的是：

A. 量化是将模型参数从高精度（如 FP32）转换为低精度（如 INT8、INT4）表示，以减少显存占用和加速推理

B. 对称量化和非对称量化的映射方式完全相同

C. 量化不会对模型精度产生任何影响

D. 量化粒度指的是模型的层数

答案：A

解析：

A 正确：量化通过降低数值精度来压缩模型，是降低部署资源需求的核心技术

B 错误：对称量化以零为中心映射，非对称量化使用零点偏移，映射方式不同

C 错误：量化会引入精度损失，但合理的量化方法可以将损失控制在可接受范围

D 错误：量化粒度指对多大范围的参数使用同一组量化参数（如逐张量、逐通道、逐组）

○ 扩展知识点：量化工具的使用

题目：

关于量化工具和实际应用，下列描述正确的是：

A. GPTQ 和 AWQ 是常用的大语言模型量化方法，可以将模型量化到 4-bit 精度并保持较好的性能

B. 量化后的模型无法在任何硬件上运行

C. 所有量化方法都需要重新训练模型

D. 量化只能应用于模型参数，不能应用于激活值

答案：A

解析：

A 正确：GPTQ 和 AWQ 是主流的后训练量化方法，支持低至 4-bit 量化

B 错误：量化后的模型可以在 CPU、GPU 等多种硬件上高效运行

C 错误：后训练量化（PTQ）不需要重新训练，量化感知训练（QAT）才需要

D 错误：量化可以同时应用于参数和激活值

○ 考核方式：概念、实现

●【3】模型压缩方法：蒸馏、剪枝、量化

○ 知识点：模型蒸馏的基本概念与基础方法、剪枝基本概念与基础方法、模型量化的基本概念与基础方法

题目：

关于模型压缩的三种主要方法，下列说法正确的是：

A. 模型蒸馏是指用一个大的教师模型指导一个小的学生模型学习，使学生模型在保持较小参数量的同时尽可能接近教师模型的性能

B. 剪枝是指增加模型的参数量以提升性能

C. 蒸馏、剪枝和量化三种方法不能组合使用

D. 模型剪枝后不需要任何微调就能保持原有性能

答案：A

解析：

A 正确：知识蒸馏通过教师 - 学生框架将大模型的知识迁移到小模型

B 错误：剪枝是移除冗余参数或结构来减小模型体积

C 错误：三种方法可以组合使用，如先蒸馏再量化

D 错误：剪枝后通常需要微调来恢复因剪枝损失的性能

○ 扩展知识点：蒸馏、剪枝和量化方法的使用

题目：

关于模型压缩方法在实际使用中的注意事项，下列描述正确的是：

A. 在实际部署中，可以先通过蒸馏获得小模型，再通过量化进一步压缩，以达到最优的压缩效果

B. 结构化剪枝和非结构化剪枝的硬件加速效果完全相同

C. 知识蒸馏不需要教师模型的输出概率分布

D. 量化感知训练（QAT）和后训练量化（PTQ）的流程完全一样

答案：A

解析：

A 正确：蒸馏 + 量化的组合是实际部署中常用的压缩策略

B 错误：结构化剪枝移除整个通道/层，更容易获得硬件加速；非结构化剪枝产生稀疏矩阵

C 错误：蒸馏通常使用教师模型的软标签（输出概率分布）来指导学生

D 错误：QAT 在训练中模拟量化效果，PTQ 在训练后直接量化，流程不同

○ 考核方式：概念、实现

●【3】资源管理与性能优化

○ 知识点：计算资源分配与调度、模型性能的瓶颈分析与优化

题目：

关于大模型推理中的资源管理和性能优化，下列说法正确的是：

A. 大模型推理的性能瓶颈可能来自计算（compute-bound）或内存带宽（memory-bound），不同阶段的瓶颈可能不同

B. 所有大模型推理任务的瓶颈都相同，都是计算瓶颈

C. 增加 GPU 数量一定能线性提升推理速度

D. 模型性能优化只需要关注模型精度，不需要考虑延迟和吞吐量

答案：A

解析：

A 正确：预填充阶段通常是计算瓶颈，解码阶段通常是内存带宽瓶颈

B 错误：不同阶段和不同配置下瓶颈不同，需要具体分析

C 错误：多 GPU 带来通信开销，不能保证线性加速

D 错误：实际部署需要综合考虑精度、延迟、吞吐量等多个指标

○ 扩展知识点：分布式资源管理、硬件与软件的协同优化

题目：

关于分布式资源管理和硬件软件协同优化，下列描述正确的是：

A. 分布式推理中常用张量并行和流水线并行来将大模型分布到多个设备上，以突破单设备显存限制

B. 硬件升级后不需要对软件做任何适配，性能会自动提升

C. 张量并行不需要设备间通信

D. 分布式推理只有流水线并行一种策略

答案：A

解析：

A 正确：张量并行将单层参数切分到多设备，流水线并行将不同层分配到不同设备

B 错误：新硬件通常需要软件适配（如 CUDA 版本、算子优化）才能充分发挥性能

C 错误：张量并行需要频繁的设备间 All-Reduce 通信来同步中间结果

D 错误：还有张量并行、数据并行、专家并行等多种策略

○ 考核方式：概念

提示学习

●【1】提示工程

○ 知识点：提示学习的目的、提示学习的范围与局限

题目：

关于提示工程（Prompt Engineering）的主要目的和局限，下列说法正确的是：

A. 目的在于改变模型权重，使其永久掌握新知识

B. 范围仅限于文本分类，无法处理多模态任务

C. 局限性之一是模型对提示词的措辞和顺序高度敏感

D. 只要提示词足够长，就能彻底消除模型的所有幻觉

答案：C

解析： 提示工程是在不改变模型参数的前提下引导模型输出。其局限性在于对输入的敏感性（Robustness 问题），微小的改动可能导致截然不同的结果。

○ 扩展知识点：提示方法的应用场景

题目：

在以下场景中，哪项最适合应用提示学习方法？

A. 需要从零开始训练一个全新的深度学习模型

B. 希望利用现有预训练模型完成特定文本分类任务

C. 处理大规模结构化数据的统计分析

D. 进行复杂的数学公式推导计算

答案：B

解析：提示学习（Prompt Learning）的核心是利用预训练模型的知识，通过设计合适的提示来引导模型完成特定任务，无需大量标注数据和重新训练。因此最适合 B 选项的场景。A 选项需要从零训练，C 和 D 选项不是提示学习的典型应用场景。

○ 考核方式：概念、简答

●【1】人工提示设计

○ 知识点：常见提示设计方法与技巧、常见模型 API 的使用

题目：

在提示学习中，以下哪项不属于人工提示设计的基本原则？

A. 提示应清晰明确，避免歧义

B. 提示应包含足够上下文信息

C. 提示应尽可能复杂以测试模型极限

D. 提示应考虑目标模型的特性

答案：C

解析： 好的提示设计应该遵循清晰性、信息充分性和适配

性原则。提示应该简洁明确而非复杂化，过于复杂的提示反而可能混淆模型、降低性能。有效的提示设计需要：避免歧义（A）、提供充足上下文（B）、根据模型特点调整（D），而不是刻意增加复杂度。

○ 扩展知识点：提示设计的自动化方法

题目：

在自动化提示优化（如 APE 或 DSPy 框架）中，通常使用一个“导师模型”来生成并迭代提示词。请补全下列简易自动化提示生成的逻辑：

```
def auto_prompt_generator(task_description, feedback):
# 根据任务描述和之前的失败反馈，让模型生成一个更好的提示词
metaprompt = f"Task: {task_description}. Previous feedback: {feedback}. Generate a better prompt:"

# [ 填空 1]：调用模型生成新的提示词
new_prompt = llm._____(metaprompt)
    return new_prompt

# 优化循环
current_prompt = " 直接回答问题 "
for i in range(3):
result = run_task(current_prompt)
score, feedback = evaluate(result)
# [ 填空 2]：根据反馈更新提示词
current_prompt = _____(task_description, feedback)
```

答案：1. generate (或 predict / invoke)；

2. auto_prompt_generator

解析：自动化提示设计的核心是利用 LLM 作为“优化器”，根据执行反馈自动迭代 Prompt。

○ 考核方式：概念、实现

●【2】上下文学习

○ 知识点：上下文提示定义、模板、底层机制

题目：

上下文学习（In-Context Learning）的核心特征是：

A. 需要在训练集上进行梯度下降更新模型参数

B. 依靠提示词中的少量示例（Few-shot）让模型通过模式匹配完成任务

C. 必须使用外部数据库存储所有历史记录

D. 仅在模型预训练阶段发挥作用

答案：B

解析：ICL 是一种推理期的学习能力，模型不更新参数，而是通过输入中的示例（Input-Output Pairs）进行类比推理。

○ 扩展知识点：上下文学习的增强策略

题目：

为了增强上下文学习的效果，在选择示例（Exemplars）时最有效的策略是：

A. 随机选择任何任务的示例

B. 选择语义上与当前输入最相似的示例

C. 选择最长的示例

D. 每次都使用相同的固定示例

答案：B

解析：研究表明，选择与当前输入（Query）在语义空间中相近的示例（k-NN 检索示例）能显著提升 ICL 的准确率。

○ 考核方式：概念、实现

●【3】思维链提示

○ 知识点：思维链的基本形式、思维链的优化策略

题目：

Zero-shot CoT（零样本思维链）最简单且常用的触发短语是:

A." 请给出最终答案 "

B." 不要胡说八道 "

C." 让我们一步步思考 "

D." 这是一个分类任务 "

答案：C

解析： "Let's think step by step" 能诱导模型生成中间推理步骤，从而显著提高复杂问题的解决能力。

○ 扩展知识点：思维链的基础原理

题目：

关于思维链（CoT）提升模型性能的原理，下列描述最准确的是:

A. 它通过增加输入长度来强迫模型运行更久

B. 它将复杂的全局任务分解为一系列连续的局部中间推理步骤

C. 它通过改变词表概率分布来消除随机性

D. 它仅通过增加 Token 数量来规避计算量限制

答案：B

解析：思路链的本质是任务分解，通过显式表达中间逻辑，降低了模型直接从问题跳转到复杂答案的推理难度。

○ 考核方式：概念、实现

●【2】检索增强

○ 知识点：基本概念、常见使用方法

题目：

检索增强生成（RAG）相较于直接生成，其核心优势在于:

A. 能够减少显存占用

B. 能够利用外部实时知识并缓解事实性幻觉

C. 能够加快推理速度

D. 能够让模型自动学会编程

答案：B

解析： RAG 将外部权威知识作为上下文提供给模型，解决了模型“预训练知识陈旧”和“虚假记忆”的问题。

○ 扩展知识点：检索增强的增强策略（自主检索调用、效率提升等）

题目：

在高级 RAG 流程中，通常会对检索到的文档进行重排序（Rerank）以提升质量。请补全 RAG 增强流程的代码：

```
def advanced_rag(query):
# 1. 从向量数据库初步检索回 K 个相关文档
initial_docs = vector_db.search(query, k=10)

# 2. [ 填空 1]：使用重排序模型对文档进行精选，选出最相关的 Top 3
refined_docs = reranker._____(query, initial_docs, top_k=3)

# 3. 将精选文档作为上下文拼接入提示词
context = "\n".join(refined_docs)
prompt = f"Context: {context}\nQuestion: {query}\nAnswer:"

# 4. [ 填空 2]：将增强后的提示词发送给大模型
response = llm._____(prompt)
return response
```

答案： 1. rerank（或 score / rank）；2. generate（或 predict / invoke）

解析： 检索增强的优化策略包括重排序（Reranking），它能过滤掉初步检索中的噪声，确保提供给模型的上下文最准确。

○ 考核方式：概念、实现

复杂推理

●【1】认知推理

○ 知识点：推理的基本方法与范畴

题目：

关于推理的基本方法与范畴，下列说法正确的是：

A. 推理包括演绎推理、归纳推理和溯因推理等多种形式，大语言模型可以通过不同策略模拟这些推理过程

B. 推理仅指数学计算，不包括逻辑判断和因果分析

C. 大语言模型的推理能力完全依赖于模型参数量，与训练数据无关

D. 所有类型的推理任务难度相同，不需要不同的处理策略

答案：A

解析：

A 正确：推理涵盖多种形式，LLM 通过思维链等策略模拟不同类型的推理

B 错误：推理范畴广泛，包括逻辑推理、因果推理、常识推

理等

C 错误：推理能力与训练数据的质量和多样性也密切相关

D 错误：不同推理任务（如数学、逻辑、常识）难度和策略需求不同

○ 扩展知识点：感知、认知与推理的区别

题目：

关于感知、认知与推理的区别，下列描述正确的是：

A. 感知侧重于从原始输入中提取信息（如图像识别），认知侧重于理解和表示知识，推理侧重于基于已有知识进行逻辑演绎

B. 感知、认知和推理是完全相同的概念

C. 推理不需要以感知和认知为基础

D. 大语言模型只具备推理能力，不具备任何感知和认知能力

答案：A

解析：

A 正确：三者形成层级关系，感知→认知→推理，逐步从低层到高层

B 错误：三者是不同层次的智能能力

C 错误：推理通常建立在感知获取信息、认知理解信息的基础之上

D 错误：多模态大模型已具备一定的感知能力，语言理解体现了认知能力

○ 考核方式：概念

●【1】长思维链模型

○ 知识点：长思维链推理模式的理解、测试时间扩展

题目：

关于长思维链（Chain-of-Thought）推理和测试时间扩展，下列说法正确的是：

A. 长思维链推理通过让模型在给出最终答案前生成详细的中间推理步骤来提升复杂任务的准确率

B. 测试时间扩展（Test-Time Scaling）是指在训练时增加数据量

C. 思维链推理只能应用于数学题，不能用于其他类型的推理任务

D. 生成更长的推理过程一定会降低模型的回答质量

答案： A

解析：

A 正确：CoT 让模型 " 思考 " 中间步骤，将复杂问题分解，显著提升推理准确率

B 错误：测试时间扩展是指在推理（测试）阶段投入更多计算资源来提升性能

C 错误：思维链推理可以用于逻辑推理、代码生成、常识推理等多种任务

D 错误：对于复杂推理任务，更详细的推理过程通常能提高准确率

○ 扩展知识点：使用推理模型解决常见逻辑、因果、数学、代码、科学任务任务

题目：

关于使用推理模型解决不同类型任务，下列描述正确的是：

A. 推理模型（如 DeepSeek-R1、OpenAI o1）通过长思维链在数学、代码、科学等需要多步推理的任务上表现优异

B. 推理模型只能解决数学问题，不能处理代码和科学任务

C. 推理模型在所有任务上的表现都优于非推理模型

D. 推理模型不需要更多的推理计算量就能获得更好的效果

答案： A

解析：

A 正确：推理模型通过长思维链在多步推理任务上有显著优势

B 错误：推理模型可以处理数学、代码、逻辑、科学等多种复杂任务

C 错误：在简单任务上推理模型可能不比普通模型更好，且推理开销更大

D 错误：推理模型以更多的推理计算量换取更高的准确率

○ 考核方式：概念、实现

●【2】基于监督微调的推理模型训练

○ 知识点：长思维链数据的搜集与构建、长思维链指令蒸馏方法

题目：

关于基于监督微调的推理模型训练，下列说法正确的是：

A. 长思维链指令蒸馏是指使用强推理模型（如 GPT-4、DeepSeek-R1）生成包含详细推理过程的训练数据，然后用这些数据微调较小的模型

B. 长思维链数据的构建不需要包含中间推理步骤，只需要最终答案

C. 指令蒸馏后的小模型性能一定超过教师模型

D. 长思维链数据可以使用任意随机文本来构建

答案： A

解析：

A 正确：指令蒸馏利用强模型生成高质量的思维链推理数据

来训练小模型

B 错误：长思维链数据的核心价值就是包含详细的中间推理步骤

C 错误：蒸馏后的小模型通常不会超过教师模型，但能显著提升自身推理能力

D 错误：长思维链数据需要包含正确的推理步骤和答案，不是随机文本

○ 扩展知识点：以有监督微调方式进行推理模型训练

题目：

关于以监督微调方式训练推理模型的流程，下列描述正确的是：

A. 监督微调训练推理模型的典型流程是：收集高质量推理数据 → 构建指令 - 推理过程 - 答案格式 → 使用交叉熵损失进行微调

B. 监督微调不需要任何标注数据

C. 推理模型的监督微调只关注最终答案的正确性，不关注推理过程

D. 监督微调方式训练的推理模型不能进行任何推理任务

答案：A

解析：

A 正确：标准的监督微调流程，数据包含完整的推理过程和答案

B 错误：监督微调需要标注好的推理数据

C 错误：训练推理模型不仅要求答案正确，推理过程的质量也非常重要

D 错误：监督微调是训练推理模型的有效方法之一

○ 考核方式：概念、实现

●【3】基于强化学习的推理模型训练

○ 知识点：以 RL 的方式进行推理能力的训练，包括结果奖励建模和过程奖励建模

题目：

关于基于强化学习训练推理模型的方法，下列说法正确的是：

A. 结果奖励建模（ORM）根据最终答案的正确性给予奖励，过程奖励建模（PRM）对推理过程的每个步骤给予奖励反馈

B. 强化学习训练推理模型不需要任何奖励信号

C. 结果奖励建模和过程奖励建模完全相同

D. 基于强化学习的推理训练不使用策略梯度或 PPO 等优化算法

答案：A

解析：

A 正确：ORM 只看最终结果，PRM 对每步推理给出奖励，PRM 能提供更细粒度的反馈

B 错误：强化学习的核心就是通过奖励信号引导模型优化

C 错误：ORM 关注结果正确性，PRM 关注过程正确性，粒度不同

D 错误：PPO、GRPO 等策略优化算法是 RL 训练推理模型的常用方法

○ 扩展知识点：推理过程中的探索策略

题目：

关于强化学习训练推理模型时的探索策略，下列描述正确的是：

A. 在强化学习训练中，模型需要平衡探索（尝试新的推理路径）和利用（使用已知有效的推理策略）

B. 探索策略只要求模型始终使用相同的推理路径

C. 增大采样温度会减少探索的多样性

D. 探索策略对推理模型的训练效果没有影响

答案：A

解析：

A 正确：探索 - 利用平衡是 RL 的核心问题，推理模型需要尝试多样化推理路径

B 错误：探索的核心就是尝试不同的推理路径，而非固定路径

C 错误：增大温度会增加采样的随机性和多样性，促进探索

D 错误：探索策略直接影响模型能否发现更优的推理路径

○ 考核方式：概念、公式、实现

●【3】基于搜索的大模型推理

○ 知识点：基于搜索的测试时间扩展，在测试过程中通过多路径搜索（Self-consistency）、树搜索（Tree-of-thoughts）等提升模型推理能力

题目：

关于基于搜索的测试时间扩展方法，下列说法正确的是：

A. Self-consistency 方法通过多次采样生成多条推理路径，然后选择出现最多的答案作为最终结果

B. Tree-of-Thoughts 只生成一条线性推理链，不涉及分支搜索

C. 基于搜索的方法不会增加推理阶段的计算开销

D. Self-consistency 只采样一次就能得到可靠的答案

答案：A

解析：

A 正确：Self-consistency 通过多次采样 + 多数投票来提升推理准确率

B 错误：Tree-of-Thoughts 将推理组织成树状结构，每个节点可以展开多个分支

C 错误：基于搜索的方法以增加推理计算量换取准确率提升

D 错误：Self-consistency 的核心是多次采样取一致性最高的答案

○ 扩展知识点：搜索效率与准确性的权衡

题目：

关于搜索效率与准确性的权衡，下列描述正确的是：

A. 增加搜索路径数量可以提升推理准确性，但会线性增加推理时间和计算成本，需要在效率和准确性之间找到平衡点

B. 搜索路径越多推理越快

C. 搜索效率与准确性之间不存在任何权衡

D. 减少搜索路径数量一定能提升准确性

答案：A

解析：

A 正确：更多搜索路径提升准确性但增加计算成本，实际应用需要权衡

B 错误：更多搜索路径意味着更多的推理计算，推理速度会降低

C 错误：搜索效率和准确性之间存在典型的权衡关系

D 错误：减少搜索路径通常会降低准确性

○ 考核方式：概念、实现

智能体

●【1】智能体身份与角色（profile）设置

○ 知识点：profile 的设置方法、角色扮演、角色扮演能力优化

题目：

在构建智能体时，通过 System Prompt 定义其“身份设定（Profile）”的主要目的是：

A. 限制模型只能使用特定的编程语言

B. 为模型提供行为准则、专业知识背景和特定的语气风格，从而约束生成空间的分布

C. 增加模型推理时的计算量以提高逻辑性

D. 彻底替换模型的预训练知识库

答案：B

解析：

Profile（画像 / 身份）设置通过提示词工程，让模型在特定的语境（Context）下运行，使其回复更符合特定角色（如：资深医生、幽默导游）的预期行为。

○ 扩展知识点：角色扮演中的一致性保持

题目：

在长对话或复杂任务中，智能体容易出现“出戏（Out of Character）”现象。以下哪项是保持角色一致性的有效优化手段?

A. 在每一轮对话的 Prompt 中都重复强调核心身份设定

B. 仅在对话的第一轮设置角色，后续完全依靠模型记忆

C. 增加 Temperature（温度）参数以提高回答的随机性

D. 减少 Context（上下文）的长度以节省 Token

答案：A

解析：

由于模型注意力机制的限制，长对话中初始设定权重会下降。通过“固定前缀”或在每轮输入中动态注入 Profile 信息（Prompt Injection），可以有效维持角色的稳定性。

○ 考核方式：概念、实现

●【2】智能体记忆机制

○ 知识点：智能体记忆种类、显式记忆和隐式记忆、记忆的存储和读取

题目：

关于智能体的记忆机制，下列说法正确的是：

A. “显式记忆”通常指通过外部数据库（如向量数据库）存储并检索到的事实性知识

B. “隐式记忆”是指智能体在运行过程中通过写日志（Logs）记录的行为

C. 短期记忆通常存储在硬盘上，长期记忆存储在显存中

D. 智能体无法拥有长期记忆，因为 Transformer 的窗口是有限的

答案：A

解析：

A 正确：显式记忆指可直接读取和检索的信息。隐式记忆通常指模型通过微调或长上下文学习到的、内化在参数或模式中的知识。

C 错误：短期记忆通常指 Context window 内的信息；长期记忆依赖外部存储（RAG/Database）。

○ 扩展知识点：记忆的长期保持与遗忘问题

题目：

为了解决长时对话中“记忆碎片化”和“上下文窗口限制”问题，

以下哪种记忆管理策略最为合理?

A. 永远保留所有历史对话记录，不做任何处理

B. 采用“滑动窗口”丢弃早期记忆，或利用 LLM 对旧记忆进行摘要（Summarization）后再存储

C. 只要显存足够，就不需要考虑遗忘问题

D. 定期清空所有记忆以保证模型不产生幻觉

答案：B

解析：

有效的记忆机制需要平衡“完整性”和“效率”。摘要化（将长对话浓缩成关键信息）和向量化检索（基于语义召回相关记忆）是解决长时记忆的主流技术。

○ 考核方式：概念、实现

●【2】智能体工具使用

○ 知识点：工具检索，工具调用优化方法

题目：

在智能体调用外部工具（Tool Use/Function Calling）的过程中，模型的主要作用是:

A. 直接执行 API 后台的 Python 代码

B. 根据用户意图和工具描述（Docstring），生成符合格式要求的调用指令（如 JSON）

C. 自动修补 API 接口的漏洞

D. 在没有互联网连接的情况下伪造 API 返回结果

答案：B

解析：

LLM 在工具使用中充当“大脑”和“调度器”，它负责识别何时需要工具，并按预定义格式输出参数。实际的执行由外部系统完成。

○ 扩展知识点：工具库的构建

题目：

当智能体可用的工具数量非常庞大（如超过 1000 个 API）时，直接将所有工具描述放入 Prompt 会导致超出 Token 限制。此时应采取的优化方案是:

A. 强制模型背诵所有工具的名称

B. 引入“工具检索（Tool Retrieval）”层，先根据意图筛选出最相关的几个工具，再放入 Prompt

C. 增加硬件显存以容纳更长的 Prompt

D. 放弃使用工具，改由模型直接生成结果

答案：B

解析：

这是构建大规模智能体系统的标准做法：意图识别 -> 工具

检索 -> 精确调用。

○ 考核方式：概念、实现

●【3】多智能体通信结构

○ 知识点：典型结构、结构自主学习、通信数据优化

题目：

在多智能体系统（MAS）中，若采用“星型通信结构”，其特点是：

A. 每个智能体只与自己相邻的智能体通信

B. 所有智能体平等交流，不分主次

C. 存在一个核心节点（如 Manager Agent），负责协调所有子智能体的信息流转

D. 信息像接力棒一样按顺序传递

答案：C

解析：

星型结构（Hub-and-Spoke）效率较高，便于控制，但核心节点容易成为瓶颈。链式（Chain）是 A 描述的模式，网状（Mesh）是 B 描述的模式。

○ 扩展知识点：通信中的信息压缩与加密

题目：

多智能体频繁通信会消耗大量 Token。为了优化通信数据，以下哪种方案是不合理的？

A. 让 Agent 之间直接传递高维稠密向量（Embedding）而非自然语言文本

B. 设定通信协议，要求 Agent 仅传输关键决策参数而非全量对话

C. 禁止 Agent 之间沟通，所有结果由各 Agent 独立得出后直接汇总

D. 利用模型对通信历史进行压缩摘要

答案：C

解析：

C 破坏了多智能体协作的基础——通信。A（向量通信）、B（协议化）、D（摘要）都是常见的通信优化手段。

○ 考核方式：概念、实现

●【3】多智能体组件优化

○ 知识点：Prompt 调优、参数调优、结构调优

题目：

在多智能体协作任务中，如果发现某个执行 Agent 总是无法理解主控 Agent 的指令，最先应该尝试的优化方案是：

A. 重新设计该 Agent 的角色提示词（Role Prompt），明确输入输出格式

B. 立即对该 Agent 进行全量参数微调（Full Fine-tuning）

C. 增加该 Agent 的计算资源（GPU 数量）

D. 减少智能体数量，合并所有职能

答案：A

解析：

在智能体开发中，Prompt 调优是最快速、成本最低且通常最有效的方案，尤其是针对逻辑理解问题。

○ 扩展知识点：多智能体的协同学习

题目：

多智能体协同学习（Cooperative Learning）的一种常见形式是“评审机制”，其流程通常是：

A. 一个 Agent 生成答案，另一个 Agent 负责检查并提出修改建议，循环迭代

B. 所有 Agent 同时生成答案，最后随机抽取一个

C. 让 Agent 之间互相攻击，直到模型崩溃

D. Agent 不需要互相学习，只需各自完成任务

答案：A

解析：

这种“生成 - 评审（Generator-Critic）”或“多轮博弈”是多智能体提升输出质量的关键机制。

○ 考核方式：概念、实现

●【3】智能体 - 人协作

○ 知识点：协作种类、协作效率、协作优化

题目：

在“人机协作（Human-in-the-loop）”模式下，智能体在执行高风险任务（如自动下单或删除文件）前请求人类确认。这种协作方式的主要目的是：

A. 纯粹为了减慢执行速度

B. 引入人类监督以确保安全性和合规性，补齐模型在极端情况下的决策缺陷

C. 让智能体学习人类的点击速度

D. 证明人类比机器更聪明

答案：B

解析：

在高风险或高准确度要求的场景下，人类的审核（Approval）是智能体安全护栏的重要组成部分。

○ 扩展知识点：协作中的效率优化

题目：

在智能体与人协作的过程中，为了提升系统整体的协作效率并避免人工审核成为流程瓶颈，以下哪项优化策略最为合理？

A. 强制要求智能体的每一步原子操作都必须由人工点击确认

B. 采用“基于置信度的干预”：仅当智能体决策的确定性低于预设阈值时才请求人工介入

C. 为了追求极端效率，完全取消人工环节，允许智能体在所有高风险场景下自主决策

D. 增加人工审核的步骤，要求每个动作必须由多名人类专家共同签名

答案：B

解析：

A 错误：会导致严重的效率低下，使系统无法处理大规模任务。

B 正确：这是效率优化的核心策略。通过设置置信度阈值，让智能体自主处理“有把握”的任务，而将“拿不准”的复杂或高风险任务交由人类处理，实现了安全与效率的平衡。

C 错误：忽略了安全性和人类最终控制权的底线，不符合协作原则。

D 错误：这属于增加流程复杂度，会进一步降低协作效率。

○ 考核方式：概念、实现

●【3】智能体交互环境

○ 知识点：世界模型概念、智能体 - 环境交互、环境反馈

题目：

在具身智能或复杂任务中，“世界模型（World Model）”的作用是：

A. 存储全球各国的地理信息

B. 模拟环境的运行规律，预测智能体采取某个动作后环境的状态变化及奖励

C. 替代 LLM 进行文本生成

D. 记录智能体的历史对话

答案：B

解析：

世界模型让智能体具备“预见性”，即在实际行动前，先在模拟的环境中演练动作的后果。

○ 扩展知识点：环境的基本构建与仿真方法

题目：

在构建智能体（Agent）的仿真环境（如自动化脚本执行器）时，关于“沙箱隔离（Sandboxing）”与“环境反馈（Observation）”的设计，以下说法正确的是：

A. 为了提高效率，应允许智能体直接修改宿主机的系统文件，无需隔离

B. 环境的主要作用是接收动作并返回状态（观察值），以形成闭环控制

C. 仿真环境只需要模拟成功的路径，不需要对错误异常进行反馈

D. 环境构建时应尽量屏蔽所有随机性，以保证智能体每次执行的路径完全固定

答案：B

解析：

B 正确：环境在 Agent 架构中负责接收 Action，更新状态，并向 Agent 返回 Observation（观察值）或 Reward（奖励），这是智能体实现自我迭代的基础。

A 错误：直接操作宿主机极具风险，必须通过容器或虚拟化技术进行沙箱隔离。

C 错误：错误反馈（如代码报错信息）是智能体学习“纠错”的关键数据。

D 错误：适度的环境随机性有助于提升智能体的泛化能力和鲁棒性。

○ 考核方式：概念、实现

●【2】智能体典型应用

○ 知识点：了解 WebGPT、社会模拟（斯坦福小镇等）

题目：

斯坦福大学的“生成式智能体（Generative Agents）”实验（俗称斯坦福小镇）展示了智能体在社会模拟中的什么特性？

A. 智能体只能进行简单的关键词回复

B. 通过记忆、反思和规划，多个智能体能产生复杂的涌现行为（如自发举办派对）

C. 智能体必须由真人实时操控才能运行

D. 该技术主要用于加速网页搜索

答案：B

解析：

该实验证明了当 Agent 具备长期记忆和架构逻辑后，在模拟社会环境中能够表现出类似人类的社交、传播和协同行为。

○ 扩展知识点：使用智能体框架搭建简单的应用

题目：

以下代码展示了如何构建一个简单的 Python 代码执行仿真沙箱。智能体将生成的代码发送给环境，环境在隔离空间中执行并捕获输出（Observation），最后返回给智能体。请补全缺失的逻辑。

```
import sys
from io import StringIO

class AgentSimulationEnvironment:
""" 智能体代码执行仿真环境（沙箱）"""
def __init__(self):
# 初始化一个独立的命名空间，模拟环境持久状态
self.state_scope = {}

def step(self, agent_code):
""" 执行智能体动作并返回环境观察值 """
# 1. 准备捕获标准输出
output_buffer = StringIO()
old_stdout = sys.stdout
sys.stdout = output_buffer

observation = ""
try:
# 2. [ 填空 1]：在隔离的命名空间中执行智能体生成的代码
_____(agent_code, self.state_scope)

# 3. [ 填空 2]：从缓冲器中获取执行后的输出结果作为观察值
observation = output_buffer._____()
except Exception as e:
# 如果代码执行报错，将异常信息作为环境反馈返回
observation = f"Environment Error: {str(e)}"
finally:
# 4. 恢复标准输出
sys.stdout = old_stdout

return observation
```



```
# --- 模拟智能体与环境交互循环 ---
env = AgentSimulationEnvironment()
# 智能体计划在环境中定义一个变量并打印
task_code = "x = 100; y = 200; print(x + y)"
```

答案：

exec

getvalue

step

解析：

填空 1：exec 是构建 Python 仿真环境的核心函数，它允许在指定的 globals/locals 字典（即沙箱作用域）中运行代码。

填空 2：StringIO.getvalue() 方法用于提取所有通过 print 等重定向到内存缓冲区的文本信息，这些信息构成了 Agent 对环境的 Observation。

填空 3：step 是强化学习或智能体框架中常见的交互接口名称，代表 Agent 向环境“迈出一步”。

○ 考核方式：概念、实现

模型评测

●【1】评测流程

○ 知识点：数据集划分、模型的泛化能力

题目：

关于模型评测中的数据集划分和泛化能力，下列说法正确的是：

A. 标准做法是将数据集划分为训练集、验证集和测试集，其中测试集用于最终评估模型的泛化能力

B. 可以使用训练集的表现来代替测试集评估模型的泛化能力

C. 验证集和测试集的作用完全相同，可以互相替代

D. 数据集划分方式对评测结果没有任何影响

答案：A

解析：

A 正确：标准三分法，训练集用于训练，验证集用于调参，测试集用于最终评估泛化能力

B 错误：训练集表现好不代表泛化能力强，可能存在过拟合

C 错误：验证集用于超参数调优和模型选择，测试集用于最终评估，功能不同

D 错误：划分方式（如比例、是否分层）会影响评测结果的可靠性

○ 扩展知识点：泛化的理论保证

题目：

关于模型泛化能力的理论保证，下列描述正确的是：

A. 充足的训练数据量和合理的模型复杂度有助于提供更好的泛化保证

B. 模型越复杂泛化能力一定越强

C. 泛化误差只取决于训练误差，与模型复杂度无关

D. 只要训练误差为零，模型的泛化能力就一定很好

答案：A

解析：

A 正确：泛化理论（如 VC 理论）表明数据量和模型复杂度的平衡是泛化的关键

B 错误：过于复杂的模型容易过拟合，泛化能力反而下降

C 错误：泛化误差与模型复杂度密切相关，过复杂的模型泛化误差大

D 错误：训练误差为零可能是过拟合的表现

○ 考核方式：概念、简答

●【1】评测指标

○ 知识点：熟悉使用常见评测指标精确率、召回率、F1 分数、困惑度、BLEU、ROUGE 等、准确定、成功率、NDCG

题目：

关于常见的模型评测指标，下列说法正确的是：

A. F1 分数是精确率和召回率的调和平均值，综合衡量模型的精确性和覆盖能力

B. BLEU 指标主要用于分类任务的评测

C. 困惑度（Perplexity）越高表示语言模型的质量越好

D. ROUGE 指标与文本生成质量无关

答案：A

解析：

A 正确：F1 = 2 × Precision × Recall / (Precision + Recall)，是两者的调和平均

B 错误：BLEU 主要用于机器翻译等文本生成任务，衡量生成文本与参考文本的 n-gram 重合度

C 错误：困惑度越低表示模型越好，代表模型对数据的拟合程度越高

D 错误：ROUGE 用于评测文本摘要等生成任务的质量

○ 扩展知识点：评测指标的局限性

题目：

以下代码实现了 BLEU 评测指标的计算，请阅读代码并填写空缺部分。

```
import math
from collections import Counter

def compute_bleu(reference, candidate, max_n=4):
"""
计算 BLEU 分数（简化版）
参数：
reference: 参考文本（分词后的列表），如 ["the", "cat", "sat", "on", "the", "mat"]
candidate: 候选文本（分词后的列表），如 ["the", "cat", "on", "the", "mat"]
max_n: 最大 n-gram 阶数
返回：
BLEU 分数 (float)
"""
precisions = []

for n in range(1, max_n + 1):
# 生成 n-gram
ref_ngrams = [tuple(reference[i:i+n]) for i in range(len(reference) - n + 1)]
cand_ngrams = [tuple(candidate[i:i+n]) for i in range(len(candidate) - n + 1)]

ref_counts = Counter(ref_ngrams)
cand_counts = Counter(cand_ngrams)

# [1] 计算裁剪后的匹配数：候选 n-gram 的计数不能超过参考中对应 n-gram 的计数
clipped_count = 0
for ngram, count in cand_counts.items():
_____

total_count = len(cand_ngrams)
if total_count == 0:
precisions.append(0)
```

[1] 和 [2] 处应分别填入：

A. [1] clipped_count += min(count, ref_counts.get(ngram, 0)) [2] bp = 0.0 if len(candidate) == 0 else (math.exp(1 - len(reference) / len(candidate)) if len(candidate) < len(reference) else 1.0)

B. [1] clipped_count += max(count, ref_counts.get(ngram, 0)) [2] bp = 0.0 if len(candidate) == 0 else (math.exp(1 - len(reference) / len(candidate)) if len(candidate) < len(reference) else 1.0)

C. [1] clipped_count += min(count, ref_counts.get(ngram, 0)) [2] bp = len(candidate) / len(reference)

D. [1] clipped_count += count [2] bp = math.exp(1 - len(reference) / len(candidate)) if len(candidate) < len(reference) else 1.0

答案：A

解析：

A 正确: 裁剪计数取候选 count 和参考 count 的较小值（min），防止重复词被过度计数；长度惩罚 BP 在标准形式 exp(1-r/c) 的基础上补充了 candidate 为空时返回 0.0 的边界处理，避免除零问题

B 错误：应取 min 而非 max，max 会导致匹配数超过参考中实际出现的次数

C 错误：BP 应该使用指数惩罚 exp(1-r/c)，简单的比值不是 BLEU 的标准长度惩罚公式

D 错误：不进行裁剪（直接用 count）会导致重复生成相同词的候选获得虚高的精度分数，且当 candidate 为空时仍有除零风险

○ 考核方式：概念、公式、实现

●【1】评测范式与方法

○ 知识点：基于评测基准、基于人类评估、基于模型评估

题目：

关于大语言模型的评测范式，下列说法正确的是：

A. 基于评测基准使用标准化测试集进行自动评测，基于人类评估邀请人类评判模型输出质量，基于模型评估使用另一个模型来评价目标模型

B. 基于人类评估是唯一可靠的评测方式，其他方式完全不可信

C. 基于模型评估（如 GPT-4 作为裁判）不存在任何偏差

D. 所有评测范式的结论总是完全一致的

答案：A

解析：

A 正确：三种主要评测范式各有优缺点，通常结合使用

B 错误：人类评估成本高且存在主观性，需要与自动评测结合

C 错误：模型评估可能存在位置偏差、冗长偏差等系统性偏差

D 错误：不同评测范式可能得出不一致的结论，需要综合分析

○ 扩展知识点：评测中的公平性问题

题目：

关于模型评测中的公平性问题，下列描述正确的是：

A. 评测数据集的选择、评测指标的设定以及评测环境的配置都可能影响评测的公平性

B. 评测公平性只涉及数据集的大小，与数据集的分布无关

C. 所有评测基准对所有模型都完全公平

D. 评测公平性问题已经被完全解决

答案：A

解析：

A 正确：评测公平性受多方面因素影响，包括数据分布、指标选择、评测设置等

B 错误：数据集的分布偏差是公平性问题的重要方面

C 错误：不同评测基准可能对不同架构或训练方式的模型有天然偏向

D 错误：评测公平性仍是活跃的研究领域

○ 考核方式：概念

●【2】公开综合评测集

○ 知识点：MMLU、BIG-Bench、HELM、C-Eval 等数据集的使用

题目：

关于常见的公开综合评测集，下列说法正确的是：

A. MMLU 涵盖 57 个学科的多选题，用于评测模型的多领域知识和推理能力；C-Eval 是中文领域的综合评测基准

B. 所有评测数据集测试的能力完全相同

C. BIG-Bench 只包含一种类型的任务

D. HELM 评测框架不关注模型的鲁棒性和公平性

答案：A

解析：

A 正确：MMLU 是英文多学科评测基准，C-Eval 是中文多学科评测基准

B 错误：不同评测集侧重不同能力维度

C 错误：BIG-Bench 包含 200+ 多样化任务，涵盖推理、翻译、常识等

D 错误：HELM 是全面的评测框架，关注准确性、鲁棒性、公平性等多个维度

○ 扩展知识点：评测集的构建方法、了解数据污染现象

题目：

关于评测集构建和数据污染现象，下列描述正确的是：

A. 数据污染是指评测数据在模型训练阶段被泄露或使用，导致评测结果虚高，无法真实反映模型能力

B. 数据污染对评测结果没有任何影响

C. 评测集的构建不需要考虑题目的难度分布

D. 所有公开评测集都不存在数据污染问题

答案：A

解析：

A 正确：数据污染会使模型在评测集上的表现不能代表真实泛化能力

B 错误：数据污染会导致评测分数偏高，严重影响结果可靠性

C 错误：评测集需要合理的难度梯度来区分不同水平的模型

D 错误：公开评测集的数据可能出现在训练数据中，数据污染是普遍关注的问题

○ 考核方式：概念

●【2】语言能力评测

○ 知识点：语言能力评测基本任务及对应评测指标

题目：

关于语言能力评测的基本任务和指标，下列说法正确的是：

A. 语言能力评测包括文本分类、命名实体识别、阅读理解、文本摘要等任务，不同任务使用不同的评测指标

B. 所有语言能力评测任务都使用同一个评测指标

C. 文本摘要任务只能使用精确率来评测

D. 语言能力评测不包括文本生成相关的任务

答案：A

解析：

A 正确：分类用准确率 /F1，NER 用 F1，阅读理解用 EM/F1，摘要用 ROUGE 等

B 错误：不同任务性质不同，需要不同的评测指标

C 错误：文本摘要通常使用 ROUGE 系列指标评测

D 错误：文本生成（如摘要、翻译）是语言能力评测的重要组成部分

○ 扩展知识点：领域特定任务的语言能力评测

题目：

关于领域特定任务的语言能力评测，下列描述正确的是：

A. 领域特定的语言能力评测需要使用领域相关的评测数据和

指标，通用评测集可能无法充分反映模型在特定领域的能力

B. 通用评测集可以完全代替领域特定评测

C. 领域评测只需要看模型的困惑度即可

D. 医学、法律等专业领域不需要专门的评测集

答案：A

解析：

A 正确：领域特定评测需要专业数据和指标来评估模型的领域适应性

B 错误：通用评测无法覆盖特定领域的专业知识和术语

C 错误：领域评测需要多维度指标，如准确性、专业术语使用等

D 错误：专业领域对准确性要求更高，需要专门的评测集

○ 考核方式：概念

●【2】知识利用能力评测

○ 知识点：知识利用能力评测基本任务（闭卷问答、开卷问答、知识不全）及对应指标

题目：

关于知识利用能力评测的基本任务，下列说法正确的是：

A. 闭卷问答要求模型仅依靠自身参数中存储的知识回答问题，开卷问答允许模型参考外部文档

B. 闭卷问答和开卷问答的区别仅在于题目难度不同

C. 知识不全场景的评测与模型在信息不足时的表现无关

D. 所有知识利用任务都使用 BLEU 来评测

答案：A

解析：

A 正确：闭卷依靠模型内部知识，开卷可以检索外部知识，知识不全测试模型处理不完整信息的能力

B 错误：核心区别是是否允许访问外部知识源

C 错误：知识不全场景正是测试模型在信息不充分时能否正确判断和回答

D 错误：问答任务通常使用准确率、EM（完全匹配）、F1 等指标

○ 扩展知识点：知识更新的时效性

题目：

关于模型知识更新的时效性，下列描述正确的是：

A. 大语言模型的知识有截止日期，模型无法知道训练数据截止后发生的新事件，这是评测中需要关注的重要问题

B. 大语言模型的知识是实时更新的，不存在时效性问题

C. 知识时效性问题可以通过增加模型参数量完全解决

D. 评测模型时不需要考虑知识的时效性

答案：A

解析：

A 正确：LLM 的知识受限于训练数据的时间范围，存在知识截止问题

B 错误：模型参数在训练后固定，知识不会自动更新

C 错误：增加参数量不能解决知识时效性问题，需要检索增强或持续学习

D 错误：评测时需要注意题目涉及的知识是否在模型训练数据截止日期之后

○ 考核方式：概念

●【2】复杂推理评测

○ 知识点：知识利用能力评测基本任务（闭卷问答、开卷问答、知识不全）及对应指标

题目：

关于复杂推理能力的评测，下列说法正确的是：

A. 复杂推理评测通常包括数学推理、逻辑推理、代码推理等维度，使用准确率等指标来衡量模型的推理正确性

B. 复杂推理评测只关注模型的最终答案，不关注推理过程

C. 所有推理任务的难度都相同

D. 复杂推理评测不需要专门的评测数据集

答案：A

解析：

A 正确：推理评测涵盖多个维度，通过准确率等指标衡量推理结果的正确性

B 错误：越来越多的评测也关注推理过程的质量和正确性

C 错误：不同推理任务有不同的难度等级，评测集通常包含多个难度层次

D 错误：推理评测需要精心设计的数据集，如 GSM8K（数学）、HumanEval（代码）等

○ 扩展知识点：推理评测的可靠性

题目：

关于推理评测的可靠性，下列描述正确的是：

A. 推理评测的可靠性受到数据污染、评测方式（如 few-shot 设置）和随机性等因素的影响

B. 单一评测集的结果就能完全代表模型的推理能力

C. 推理评测不受提示词（prompt）格式的影响

D. 评测结果在不同运行之间不会有任何波动

答案：A

解析：

A 正确：评测可靠性受多种因素影响，需要综合多个评测集和多次评测来保证

B 错误：单一评测集可能存在偏差，需要多个评测集交叉验证

C 错误：提示词格式对评测结果有显著影响，不同格式可能导致不同分数

D 错误：由于采样随机性，模型在同一评测集上的得分可能略有波动

○ 考核方式：概念

●【3】其他评测

○ 知识点：人类对齐评测、环境交互评测、工具使用评测、鲁棒性评测（对抗样本）

题目：

关于大语言模型的高级评测维度，下列说法正确的是：

A. 鲁棒性评测通过对抗样本（如添加微小扰动或误导性提示）测试模型在异常输入下能否保持正确的输出

B. 人类对齐评测只关注模型输出是否语法正确

C. 工具使用评测与模型调用外部 API 的能力无关

D. 环境交互评测不需要模型与外部环境进行实际交互

答案：A

解析：

A 正确：对抗样本测试模型面对刻意设计的干扰输入时的稳定性和鲁棒性

B 错误：人类对齐评测关注模型是否遵循人类价值观、安全性和有用性

C 错误：工具使用评测正是测试模型正确调用 API、搜索引擎等外部工具的能力

D 错误：环境交互评测需要模型在模拟或真实环境中执行操作

○ 扩展知识点：理解高级评测任务与模型基础能力之间的关系

题目：

关于高级评测任务与模型基础能力的关系，下列描述正确的是：

A. 高级评测任务（如工具使用、环境交互）建立在模型的基础能力（如语言理解、推理、指令遵循）之上，基础能力的提升有助于高级任务的表现

B. 高级评测任务与模型的基础语言能力完全无关

C. 基础能力评测已经足够，不需要高级评测任务

D. 高级评测任务只测试模型的记忆能力

答案：A

解析：

A 正确：高级任务是基础能力的综合体现，如工具使用需要指令理解 + 推理 + 格式输出

B 错误：高级任务高度依赖基础语言理解和推理能力

C 错误：高级评测能发现基础评测无法暴露的问题

D 错误：高级评测测试的是综合应用能力，不仅是记忆

○ 考核方式：概念

模型伦理与安全

●【1】模型偏见

○ 知识点：偏见的来源、检测与缓解方法

题目：

下列哪个项是大语言模型产生偏见的主要来源?

A. 训练数据中特定群体的样本数量不足

B. 模型算法的计算速度过快

C. 硬件设备的存储容量限制

D. 用户界面的设计不够美观

答案：A

解析：模型偏见主要源于训练数据的不平衡和偏见。如果训练数据中某些群体的样本不足或存在刻板印象，模型会学习并放大这些偏见。计算速度（B）、硬件容量（C）和界面设计（D）都与模型偏见的产生无直接关系。

○ 扩展知识点：偏见对实际应用的影响

题目：

某公司开发了一款基于大模型的自动化简历筛选系统，由于训练数据中历史高管多为男性，导致系统自动调低了女性应聘者的评分。这种现象主要体现了模型偏见的哪种实际影响?

A. 损害了算法的计算效率

B. 导致了社会不公平的自动化与规模化

C. 增加了模型部署的硬件成本

D. 提升了模型处理长文本的理解能力

答案：B

解析：

模型偏见在实际应用中（如招聘、信贷、司法预测）会直接导致对特定群体的歧视，从而将现实中的社会不公通过算法

进行放大和自动化执行。这属于算法伦理层面的重大风险，与计算效率、硬件成本等无关。

○ 考核方式：概念、案例分析

●【2】隐私保护

○ 知识点：数据隐私保护技术（如差分隐私）

题目：

在保护大模型训练数据隐私的技术中，“差分隐私（Differential Privacy）”的核心机制是：

A. 通过对训练数据进行加密，使模型无法读取任何信息

B. 在算法处理过程中引入适量的统计噪声，确保单个样本的加入或删除不会显著影响输出结果

C. 将所有数据存储在物理隔离的离线硬盘中

D. 仅允许拥有高级管理员权限的用户访问模型

答案：B

解析：

差分隐私的核心是通过数学手段注入噪声，使得外部攻击者无法通过模型的输出来反推训练集中是否包含某个特定个体的信息，从而在利用群体统计特征的同时保护个体隐私。

○ 扩展知识点：隐私与模型性能的权衡

题目：

在大模型训练中应用隐私保护技术（如强噪声干扰的差分隐私）时，通常需要面对的权衡（Trade-off）是：

A. 隐私保护越强，模型的预测准确性（Utility）往往会下降

B. 隐私保护越强，模型的参数量会随之大幅减少

C. 隐私保护越强，模型对硬件的存储需求越低

D. 隐私保护越强，模型的推理速度会呈指数级提升

答案：A

解析：

这是隐私保护领域的经典权衡：为了保护隐私而引入的噪声会干扰模型学习真实的特征分布，导致模型的性能（如准确率、困惑度等指标）出现下降。

○ 考核方式：概念

●【3】数据安全

○ 知识点：数据泄露风险、数据加密与访问控制

题目：

大模型在推理服务阶段面临的主要数据安全风险之一是“训练数据泄露”，其表现形式通常为：

A. 模型在回答用户问题时，意外输出了训练集中包含的个

人敏感信息（如身份证号、地址）

B. 模型的权重文件被竞争对手非法拷贝

C. 用户的提问被模型保存到了本地缓存中

D. 模型的计算任务因为网络延迟而中断

答案：A

解析：

数据泄露风险不仅指外部入侵，也包括模型在生成过程中由于“记忆”了过细的训练样本，导致通过特定的提示词诱导（Prompt Injection）吐露出原始训练数据中的私密信息。

○ 扩展知识点：数据安全的法律与合规问题

题目：

下列哪项符合数据合规中的“最小必要”原则?

A. 尽可能收集所有可获取的数据以训练模型

B. 仅收集实现特定功能所必需的最小数据量

C. 将用户数据永久存储以供未来使用

D. 禁止用户查询或删除其个人数据

答案：B

解析：“最小必要”要求数据收集应具有明确、合理的目的，且仅限于实现该目的的最小范围。

○ 考核方式：概念

CCF大模型能力认证大纲

Large Model Competence Certification

CCF大模型能力认证大纲

CCF Large Model Competence Certification