LMCC-青少年组-新例卷

题目结构与考纲覆盖

大题	题型	数量	难度梯度
第一大题	单选题	20 题	1-10 基础；11-15 适中；16-20 计算
第二大题	材料题	3 道材料题，共 10 小题，分布 4/3/3	T1 简单程序填空；T2 较难程序材料；T3 长篇论文阅读

一、单选题（20 * 3 分 = 60 分）

> 本大题每题只有一个正确选项。第 1-10 题考查基础概念，第 11-15 题考查中等难度的理解与辨析，第 16-20 题为计算题。每题下方给出考纲对应关系，便于回溯到能力大纲中的模块和知识点。
>
> **正式考试中题目的顺序是打乱的，需要考生自主判断题目难度**

1. 关于监督学习、无监督学习与自监督学习，下列说法正确的是：

A. 监督学习主要依赖无标签数据，自监督学习通常需要人工标注

B. 聚类通常属于监督学习，分类通常属于无监督学习

C. 自监督学习通常从数据本身构造训练信号，例如语言模型的下一个词元预测

D. 模型参数量增大后，训练数据的重要性会明显降低

答案： C

> **考纲对应：** 四. 预训练技术 - 监督学习；一. 人工智能基础概念 - 人工智能相关概念定义
> **题目标注：** 难度 1；考核方式：概念；聚焦：监督学习、无监督学习与自监督学习
> **答案解析：** 监督学习依赖标签，无监督学习常用于聚类等结构发现，自监督学习从原始数据中构造监督信号。大模型仍然依赖大量训练数据。

2. 关于模型的训练集、验证集和测试集，以下说法正确的是：

A. 训练集用于训练模型，验证集用于调整超参数，测试集用于最终评估

B. 三个数据集在小规模实验中可以灵活复用，区分不必过于严格

C. 测试集可以在训练过程中多次参考，用于持续优化模型

D. 验证集的数据量通常应尽量大于训练集

答案： A

> **考纲对应：** 一. 人工智能基础概念 - 机器学习流程及经典模型 / 验证及评测
> **题目标注：** 难度 1；考核方式：概念；聚焦：训练集、验证集与测试集
> **答案解析：** 三个数据集有明确分工：训练集用于训练模型参数，验证集用于调整超参数和早停，测试集只在最后使用一次进行最终评估。严格分离可以避免数据泄露和过拟合。

3. 与图像数据相比，自然语言作为大模型输入时最典型的底层特征是：

A. 自然语言通常表现为离散符号序列，而图像通常表现为密集像素矩阵

B. 自然语言的顺序结构较弱，而图像更天然具有语法结构

C. 图像通常可以直接作为原始输入，自然语言更依赖数字化处理

D. 自然语言中的歧义相对较少，图像理解中歧义更常见

答案： A

> **考纲对应：** 二. 大模型基础概念 - 自然语言的基础概念
> **题目标注：** 难度 1；考核方式：概念；聚焦：自然语言与图像数据特征
> **答案解析：** 自然语言通常是离散 token 序列，图像通常是二维密集数值矩阵；语言也具有顺序结构和歧义性。

4. 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

> **考纲对应：** 三. 模型架构 - 注意力机制
> **题目标注：** 难度 1；考核方式：概念；聚焦：注意力机制中的 Q/K/V
> **答案解析：** 标准注意力机制通常计算 $\text{softmax}(QK^T/\sqrt{d_k})V$。注意力权重由 Query 和 Key 的相似度决定，Value 是被加权汇总的对象。

5. 关于自回归语言模型的“下一个词元预测”任务，下列说法正确的是：

A. 它要求模型根据前文预测下一个 token，是 Decoder-only 大语言模型常见预训练目标

B. 它会利用答案右侧的 token 提供信息，因此因果掩码的重要性较低

C. 它主要适用于图像分类模型

D. 它与最大似然学习的关系较弱

答案： A

> **考纲对应：** 四. 预训练技术 - 预训练任务
> **题目标注：** 难度 1；考核方式：概念；聚焦：自回归语言模型

> **答案解析：** 下一个词元预测是典型自回归语言建模任务，常用因果掩码避免模型看到未来 token，并可用最大似然形式训练。

6. 关于大语言模型的“生成”和“理解”两类核心范式，下列说法正确的是：

A. 生成任务更侧重让模型产生新的文本，理解任务更侧重分析或判断输入内容的含义

B. 生成任务主要用于判断文本类别，理解任务主要用于续写故事或扩写文章

C. 生成和理解的差异主要体现在输出文字长短，与任务目标关系较弱

D. 大语言模型更适合生成任务，用于理解任务时参考价值较低

答案： A

> **考纲对应：** 二. 大模型基础概念 - 基本定义
> **题目标注：** 难度 1；考核方式：概念；聚焦：大语言模型的生成与理解范式
> **答案解析：** 大语言模型既可以生成新的文本，也可以对输入内容进行理解、分类、判断和分析。生成与理解的区别主要在任务目标，而不是输出长短。

7. 指令数据（Instruction Data）的典型组成是：

A. 主要包含问题描述

B. 由指令、输入（可选）、输出三部分组成

C. 主要关注模型的最终回答

D. 通常需要包含代码实现

答案： B

> **考纲对应：** 五. 指令微调 - 指令微调 / 指令数据集的构建
> **题目标注：** 难度 1；考核方式：概念；聚焦：指令微调数据格式
> **答案解析：** 指令数据通常包含任务指令、可选输入或上下文，以及期望输出。这种结构帮助模型学习理解任务并按要求作答。

8. 关于温度（Temperature）参数在文本生成中的作用，以下说法正确的是：

A. 温度升高时，生成文本通常更确定和保守

B. 温度降低时，生成文本通常更随机和多样

C. 温度为 0 时，模型会选择概率最高的词

D. 温度参数对生成结果的影响通常较小

答案： C

> **考纲对应：** 七. 解码与部署 - 随机采样及改进策略
> **题目标注：** 难度 1；考核方式：概念；聚焦：温度采样
> **答案解析：** 温度参数控制生成随机性。温度低时分布更尖锐，模型更倾向选择高概率词；温度高时分布更平滑，生成更随机。温度为 0 通常对应贪心解码。

9. 关于大语言模型的幻觉（Hallucination）现象，以下说法正确的是：

A. 幻觉是指模型生成看似合理但实际错误或不存在的信息

B. 幻觉更常见于小模型，大模型中相对较少

C. 幻觉是模型的优点，体现了创造力

D. 幻觉可以主要通过增加训练数据得到解决

答案： A

> **考纲对应：** 六. 人类对齐 - 幻象
> **题目标注：** 难度 1；考核方式：概念；聚焦：大语言模型幻觉
> **答案解析：** 幻觉指模型输出流畅、看似可信，但包含事实错误、虚构信息或无法被上下文支持的内容。大模型也可能出现幻觉，只能缓解，难以完全消除。

10. 关于 Prompt（提示词）的设计原则，以下哪项不正确？

A. 提示词应该清晰明确，避免歧义

B. 提示词越长、包含的信息越多，效果通常越好

C. 提示词应该包含必要的上下文信息

D. 提示词的设计需要考虑目标模型的特点

答案： B

> **考纲对应：** 八. 提示学习 - 人工提示设计
> **题目标注：** 难度 1；考核方式：概念；聚焦：提示词设计原则
> **答案解析：** 提示词质量不是由长度决定的。好的提示词应清晰、具体、包含必要信息；过长或过于复杂的提示可能反而混淆模型。

---

11. 关于语言模型的发展历程，下列理解最准确的是：

A. 统计语言模型、神经网络语言模型、预训练语言模型和大语言模型大致体现了语言模型能力逐步增强的发展路线

B. 大语言模型先出现，随后才发展出神经网络语言模型和统计语言模型

C. 预训练语言模型主要依赖人工规则库，对从语料中学习语言规律的依赖较低

D. 统计语言模型和大语言模型的主要区别在于是否用于中文任务

答案： A

> **考纲对应：** 二. 大模型基础概念 - 发展历程与现状
> **题目标注：** 难度 2；考核方式：概念；聚焦：语言模型发展阶段与技术路线
> **答案解析：** 语言模型大致经历了统计语言模型、神经网络语言模型、预训练语言模型到大语言模型的发展过程。这个过程体现了从局部统计和浅层表示，逐步发展到大规模预训练与生成、理解统一范式。

12. 关于编码器-解码器架构、因果解码器架构和前缀解码器架构，下列说法正确的是：

A. 编码器-解码器架构常用于输入序列到输出序列的转换任务，因果解码器架构常用于从左到右生成文本

B. 因果解码器在生成当前位置时通常会利用右侧尚未生成的 token 作为主要依据

C. 前缀解码器架构与传统卷积神经网络在文本生成任务中的作用基本相同

D. 架构选择与任务类型关系较弱，生成任务和理解任务通常使用同一种结构即可

答案： A

> **考纲对应：** 三. 模型架构 - 主流架构
> **题目标注：** 难度 2；考核方式：概念；聚焦：主流 Transformer 架构与任务类型
> **答案解析：** 编码器-解码器架构适合将输入序列转换为输出序列，例如翻译、摘要等任务；因果解码器按照从左到右的方式生成文本，生成当前位置时不能依赖未来 token。不同架构与任务类型有对应关系。

13. 关于自监督学习与预训练任务，下列说法最准确的是：

A. 自回归语言建模通常根据前文预测下一个词元，掩码预测通常根据上下文恢复被遮盖的词元

B. 对比学习主要要求模型记住每个样本的人工类别标签，与表示学习关系较弱

C. 掩码预测和自回归预测的训练目标差异较小，主要只是名称不同

D. 预训练任务通常只用于图像模型，在语言模型中参考价值较小

答案： A

> **考纲对应：** 四. 预训练技术 - 自监督学习 / 预训练任务
> **题目标注：** 难度 2；考核方式：概念；聚焦：自回归、掩码预测与预训练任务辨析
> **答案解析：** 自监督学习可以从数据本身构造训练信号。自回归语言建模通常根据已有上下文预测下一个词元；掩码预测则把部分词元遮盖起来，让模型根据上下文恢复它们。二者都是语言模型中常见的预训练任务。

14. 关于指令数据集的构建，下列做法最合理的是：

A. 可以从少量种子指令出发合成更多指令-回答样本，并通过规则检查或人工抽查过滤低质量样本

B. 指令数据主要保存任务问题，期望输出或参考回答的重要性相对较低

C. 合成指令数据时，样本数量越多通常越能代替质量控制

D. 为了便于训练，指令样本通常更适合来自较少的任务类型和较统一的表达方式

答案： A

> **考纲对应：** 四. 预训练技术 - 指令数据集的构建
> **题目标注：** 难度 2；考核方式：概念；聚焦：指令数据合成与质量控制
> **答案解析：** 指令数据通常包含指令、可选输入以及期望输出。构建数据集时，可以用种子指令扩展生成更多样本，但仍需要质量检查、去重和过滤，避免低质量或重复样本影响训练效果。

15. 关于贪心搜索和束搜索解码，下列说法正确的是：

A. 贪心搜索每一步通常只保留当前概率最高的选择，束搜索会保留多个候选序列并综合比较

B. 束搜索通常比贪心搜索计算量更小，因为候选序列管理更简单

C. 贪心搜索在每一步选择局部高概率词元，因此通常能保证整句达到全局最优

D. 增大束宽主要是为了让生成结果更随机，与候选序列数量关系较弱

答案： A

> **考纲对应：** 七. 解码与部署 - 解码方法
> **题目标注：** 难度 2；考核方式：概念；聚焦：贪心搜索与束搜索解码
> **答案解析：** 贪心搜索每一步选择当前看起来最优的词元，计算简单但可能错过整体更好的序列；束搜索保留多个候选序列，通常能探索更多可能性，但计算开销也更高。

---

16. 某训练集共有 9600 条样本，Batch Size 设置为 64，完整训练 3 个 epoch。若每个 batch 更新一次模型参数，且样本数能被 batch size 整除，则总参数更新步数为：

A. 150

B. 300

C. 450

D. 600

答案： C

> **考纲对应：** 四. 预训练技术 - 基于批次数据的训练方法
> **题目标注：** 难度 3；考核方式：计算；聚焦：Batch Size、epoch 与参数更新步数
> **答案解析：** 每个 epoch 的 batch 数为 $9600/64 = 150$。训练 3 个 epoch 时，总更新步数为 $150 \times 3 = 450$。

17. 某二分类模型在测试集上的结果为：真正例 TP=80，假正例 FP=20，假负例 FN=40。该模型的精确率（Precision）和召回率（Recall）最接近：

A. 0.67 和 0.80

B. 0.80 和 0.67

C. 0.80 和 0.80

D. 0.67 和 0.67

答案： B

> **考纲对应：** 一. 人工智能基础概念 - 验证及评测
> **题目标注：** 难度 3；考核方式：计算；聚焦：精确率与召回率
> **答案解析：** 精确率 $Precision = TP/(TP + FP) = 80/(80 + 20) = 0.80$；召回率 $Recall = TP/(TP + FN) = 80/(80 + 40) \approx 0.67$。

18. 某 14B 模型在 BF16 下权重占用约 28 GB。若改为 INT4 后，权重本体理论占用缩至 1/4；另外需要 1.6 GB 量化标尺开销与 0.9 GB 元数据开销；并预计运行时碎片化额外增

加 15%（按“量化后权重本体+固定开销”计）。则总占用最接近：

A. 9.5 GB

B. 10.2 GB

C. 10.9 GB

D. 12.4 GB

答案： C

> **考纲对应：** 七. 解码与部署 - 低资源部署策略
> **题目标注：** 难度 3；考核方式：计算；聚焦：INT4 量化部署显存估算
> **答案解析：** INT4 权重本体 = $28/4 = 7$ GB。基础占用 = $7 + 1.6 + 0.9 = 9.5$ GB。加 15% 碎片化：$9.5 \times 1.15 = 10.925$ GB，约 10.9 GB。

19. 某排序前 5 项相关性分数为 $[3,2,0,1,0]$。按 $DCG = \sum((2^{rel} - 1)/\log_2(i+1))$，NDCG@5 最接近：

A. 0.81

B. 0.90

C. 0.99

D. 1.10

答案： C

> **考纲对应：** 十一. 模型评测 - 评测指标
> **题目标注：** 难度 3；考核方式：计算；聚焦：NDCG 排序评测指标
> **答案解析：** $DCG \approx 7 + 1.893 + 0 + 0.431 = 9.324$；理想排序 $[3,2,1,0,0]$ 的 $IDCG \approx 9.393$；$NDCG \approx 9.324/9.393 \approx 0.993$。

20. 某推理服务每步设置 batch=48、平均序列长度 3072 token，单步耗时 3.0 s。系统每执行 5 步还会触发一次额外 checkpoint 同步，耗时 1.5 s。按长期平均计算，系统吞吐量最接近：

A. 38.4k token/s

B. 44.7k token/s

C. 49.2k token/s

D. 56.3k token/s

答案： B

> **考纲对应：** 七. 解码与部署 - 资源管理与性能优化
> **题目标注：** 难度 3；考核方式：计算；聚焦：推理服务吞吐量估算
> **答案解析：** 5 步总 token = $48 \times 3072 \times 5 = 737280$。5 步总耗时 = $5 \times 3.0 + 1.5 = 16.5$ s。吞吐 = $737280/16.5 \approx 44684$ token/s，约 44.7k token/s。

二、材料题 （10 * 4 分 = 40 分）

> 本大题先阅读材料，再回答材料后的单选题。材料题 1 为较简单的程序填空，包含 4 个小题；材料题 2 为较难的程序阅读与填空，包含较多补充材料和 3 个小题；材料题 3 为论文阅读题，题面较长，要求阅读翻译改写后的论文材料后完成 3 个单选小题。

材料题 1：智能体工具调用 JSON 校验器

在工具调用型智能体中，模型输出通常需要被解析为结构化 JSON，再根据工具 schema 校验工具名和参数是否合法。下面的代码实现了一个简化版工具调用校验器：它读取模型输出字符串，检查工具名、参数字段和参数类型，并返回标准化后的工具调用对象。

请阅读代码，并根据描述完成空缺部分。

```
python
import json
from typing import Any


TOOL_SCHEMAS = {
    "weather": {
        "required": {"city": str, "date": str},
        "optional": {"unit": str},
    },
    "calculator": {
        "required": {"expression": str},
        "optional": {},
    },
}


def load_tool_call(raw: str) -> dict[str, Any]:
    """将模型输出字符串解析为工具调用对象。"""
    try:
        obj = ____[1]____
    except json.JSONDecodeError as exc:
        raise ValueError("invalid json") from exc

    if not isinstance(obj, dict):
        raise ValueError("tool call must be a JSON object")
    if set(obj.keys()) != {"tool_name", "arguments"}:
        raise ValueError("tool call must contain exactly tool_name and
arguments")
    if obj["tool_name"] not in TOOL_SCHEMAS:
        raise ValueError("unknown tool")
    if not isinstance(obj["arguments"], dict):
        raise ValueError("arguments must be an object")
    return obj


def validate_arguments(tool_name: str, arguments: dict[str, Any]) -> bool:
    """根据 schema 校验参数字段和参数类型。"""
```

```
    spec = TOOL_SCHEMAS[tool_name]
    required = spec["required"]
    optional = spec["optional"]

    allowed_keys = ____[2]____
    missing_keys = ____[3]____
    extra_keys = set(arguments) - allowed_keys
    if missing_keys or extra_keys:
        return False

    type_rules = {**required, **optional}
    for name, expected_type in type_rules.items():
        if name in arguments and ____[4]____:
            return False
    return True


def normalize_tool_call(raw: str) -> dict[str, Any]:
    obj = load_tool_call(raw)
    tool_name = obj["tool_name"]
    arguments = obj["arguments"]
    if not validate_arguments(tool_name, arguments):
        raise ValueError("bad arguments")
    return {"tool_name": tool_name, "arguments": arguments}
```

[1] 需要将模型输出字符串解析为 Python 对象。请选择正确实现：

A. `json.dumps(raw)`

B. `json.loads(raw)`

C. `dict(raw)`

D. `raw.split(",")`

答案： B

> **考纲对应：** 十. 智能体 - 智能体工具使用
> **题目标注：** 难度 1；考核方式：程序填空；聚焦：工具调用 JSON 解析与参数校验
> **答案解析：** `json.loads` 将 JSON 字符串解析为 Python 对象；`json.dumps` 是反方向，把 Python 对象序列化为字符串。

[2] 需要得到该工具允许出现的全部参数名。请选择正确实现：

A. `set(required) | set(optional)`

B. `set(required) & set(optional)`

C. `list(required.values()) + list(optional.values())`

D. `required == optional`

答案： A

> **考纲对应：** 十. 智能体 - 智能体工具使用
> **题目标注：** 难度 1；考核方式：程序填空；聚焦：工具调用 JSON 解析与参数校验
> **答案解析：** 必需参数和可选参数都属于允许字段，因此需要对二者的键集合取并集。

[3] 需要找出缺失的必需参数。请选择正确实现：

A. `set(arguments) - set(required)`

B. `set(required) - set(arguments)`

C. `set(optional) - set(required)`

D. `set(arguments) | set(optional)`

答案： B

> **考纲对应：** 十. 智能体 - 智能体工具使用
> **题目标注：** 难度 1；考核方式：程序填空；聚焦：工具调用 JSON 解析与参数校验
> **答案解析：** 缺失的必需参数是“required 中有、arguments 中没有”的字段。

[4] 需要检查参数值是否符合 schema 中声明的类型。请选择正确实现：

A. `arguments[name] is expected_type`

B. `expected_type in arguments[name]`

C. `not isinstance(arguments[name], expected_type)`

D. `type_rules[name] == arguments[name]`

答案： C

> **考纲对应：** 十. 智能体 - 智能体工具使用
> **题目标注：** 难度 1；考核方式：程序填空；聚焦：工具调用 JSON 解析与参数校验
> **答案解析：** `isinstance(value, expected_type)` 用于判断值是否属于指定类型；这里需要在“不符合类型”时返回 `False`，因此使用 `not isinstance(...)`。

材料题 2：API 文档检索助手

API 文档检索助手是一类典型的检索增强系统。它先把 API 文档片段编码成向量并建立索引；当用户提出问题时，再把问题编码成向量，与文档向量计算相似度，选出最相关的文档片段，用于后续回答生成。

补充材料：

1. `tokenizer(texts, return_tensors="pt", padding=True, truncation=True)` 会返回一个字典，常见字段包括 `input_ids` 和 `attention_mask`。
2. 嵌入模型输出的 `last_hidden_state` 形状通常为 `[batch_size, seq_len, hidden_dim]`。
3. `attention_mask` 的形状通常为 `[batch_size, seq_len]`，其中 1 表示真实 token，0 表示 padding token。
4. 对带 padding 的序列做平均池化时，应排除 padding 位置，否则短文本向量会被填充值稀释。

5. 若查询向量和文档向量都已做 L2 归一化，则二者点积等价于余弦相似度。

6. 如果查询向量形状为 `[1, hidden_dim]`，文档矩阵形状为 `[num_docs, hidden_dim]`，则一次性计算所有文档相似度可使用矩阵乘法，结果形状为 `[1, num_docs]`。

请阅读代码，并根据描述完成空缺部分。

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

class DocumentRetriever:
    def __init__(self, embedding_model, tokenizer, document_corpus):
        self.embedding_model = embedding_model
        self.tokenizer = tokenizer
        self.document_corpus = document_corpus
        self.doc_embeddings = self.index_documents(document_corpus)

    def encode_texts(self, texts):
        tokens = ____[1]____
        tokens = {k: v.to(self.embedding_model.device) for k, v in
tokens.items()}

        with torch.no_grad():
            model_output = self.embedding_model(**tokens)
            hidden_states = model_output.last_hidden_state
            mask = tokens["attention_mask"]
            sentence_embeddings = ____[2]____
            sentence_embeddings = F.normalize(sentence_embeddings, p=2, dim=1)

        return sentence_embeddings

    def index_documents(self, documents):
        doc_vectors = self.encode_texts(documents)
        return doc_vectors.cpu()

    def retrieve_relevant_docs(self, query_embedding, top_k=3):
        if query_embedding.dim() == 1:
            query_embedding = query_embedding.unsqueeze(0)

        query_normalized = F.normalize(query_embedding, p=2,
dim=1).to(self.doc_embeddings.device)
        similarity_scores = ____[3]____
        similarity_scores = similarity_scores.squeeze(0)

        top_scores, top_indices = torch.topk(
            similarity_scores,
            k=min(top_k, len(self.document_corpus)),
        )
        return [
            {"document": self.document_corpus[idx], "relevance":
float(top_scores[i])}
            for i, idx in enumerate(top_indices)
        ]
```

[1] 在 `encode_texts` 方法中，需要将输入文本列表转换为模型所需的 token 格式。请选择正确实现：

A. `self.tokenizer.encode(texts, return_tensors="pt", padding=True, truncation=True, max_length=512)`

B. `self.embedding_model(texts, return_tensors="pt", padding=True, truncation=True, max_length=512)`

C. `self.tokenizer(texts, return_tensors="pt", padding=True, truncation=True, max_length=512)`

D. `self.tokenizer.tokenize(texts, return_tensors="pt", padding=True, truncation=True)`

答案： C

> **考纲对应：** 八. 提示学习 - 检索增强
> **题目标注：** 难度 2；考核方式：程序填空；聚焦：向量检索、掩码池化与相似度计算
> **答案解析：** tokenizer 的调用接口会返回包含 `input_ids` 和 `attention_mask` 的字典；模型不能直接处理原始文本字符串。

[2] 需要对序列隐藏状态做 masked mean pooling，得到句子级向量。请选择正确实现：

A. `hidden_states.mean(dim=1)`

B. `hidden_states[:, 0, :]`

C. `(hidden_states * mask.unsqueeze(-1)).sum(1) / mask.sum(1, keepdim=True)`

D. `torch.max(hidden_states, dim=1)[0]`

答案： C

> **考纲对应：** 八. 提示学习 - 检索增强
> **题目标注：** 难度 2；考核方式：程序填空；聚焦：向量检索、掩码池化与相似度计算
> **答案解析：** 由于输入经过 padding，对序列维度求平均时应使用 `attention_mask` 屏蔽 padding 位置。C 选项先将 mask 扩展到隐藏维度，再对有效 token 求和并除以有效 token 数。

[3] 需要一次性计算查询向量与全部文档向量的相似度。请选择正确实现：

A. `torch.matmul(query_normalized, self.doc_embeddings)`

B. `torch.matmul(self.doc_embeddings, query_normalized)`

C. `torch.matmul(query_normalized, self.doc_embeddings.T)`

D. `(query_normalized * self.doc_embeddings).sum(dim=0)`

答案： C

> **考纲对应：** 八. 提示学习 - 检索增强
> **题目标注：** 难度 2；考核方式：程序填空；聚焦：向量检索、掩码池化与相似度计算
> **答案解析：** 查询向量形状为 `[1, hidden_dim]`，文档矩阵转置后形状为 `[hidden_dim, num_docs]`，矩阵乘法得到 `[1, num_docs]` 的相似度矩阵。归一化后点积等价于余弦相似度。

材料题 3：论文阅读：思维链忠实性与早退策略

请阅读下面根据论文 **Reasoning Theater: Disentangling Model Beliefs from Chain-of-Thought**（arXiv:2603.05488v4，2026-05-28）整理的中文阅读材料，然后回答 3 个问题。材料为面向考试的中文译写版，保留论文的问题设定、方法比较、实验现象和工程启发。

中文阅读材料：

1. 研究背景：思维链为什么看起来重要

近年来，思维链提示被广泛用于提升大语言模型在数学、科学和复杂推理任务上的表现。特别是经过强化学习训练的推理模型，常常会在给出答案前生成较长的中间推理文本，形式上像是在逐步分析问题、排除错误选项、检查条件并修正判断。直观上，如果模型把推理过程写出来，人们似乎就可以通过阅读这些文字判断模型是否真的在推理、哪里出现了逻辑错误、是否有潜在风险。因此，一些安全和可解释性方案把“监控思维链文本”视为理解模型内部过程的重要窗口。

这篇论文首先提醒读者：思维链很有用，但它不一定等于模型真实内部过程的逐字转录。一个推理模型可能写出结构清楚、逻辑连贯、看似诚实的推理文本，但这些文本仍可能没有及时反映模型内部已经形成的判断。换句话说，外显思维链更像模型在任务过程中“说出来的内容”，而不是模型内部状态的完整记录。理解这一点，是阅读整篇论文的基础。

2. 核心概念：什么是“表演式思维链”

论文把一种关键现象称为“表演式思维链”。它并不是说模型完全随机输出，也不是说所有长思维链都没有价值，而是指：模型在内部激活中已经较早形成了对最终答案的高置信判断，但外部输出仍继续生成看似犹豫、探索、推导或自我检查的文字。也就是说，模型可能在内部“已经知道答案”，却在文本上继续表现为“还在一步步想”。

这里的关键矛盾不是模型最终是否答对，而是“内部答案形成时间”和“外部文本承认答案的时间”之间是否存在明显错位。如果内部状态很早就指向某个答案，而文本要到很后面才透露这个答案，那么只阅读文本的监控者就会滞后。对于安全监控和可解释性来说，这种滞后很重要：如果模型已经形成某种决策或意图，却没有在思维链里及时表现出来，那么监控者可能会误判模型仍处于开放推理状态。

3. 研究设计：三种方法比较“模型何时知道答案”

为了研究这种错位，论文比较了三类“早期解码”方法。第一类是激活探针。研究者在模型生成思维链的不同位置截取内部激活，并训练一个轻量探针预测模型最终会选择的答案。论文使用的是注意力池化探针：它不是只读取某个 token 的激活，而是对一段前缀中的隐藏状态做可学习的加权汇聚，再输出 A、B、C、D 等选项的概率。这样做是因为长推理轨迹中的答案信息可能分散在多个位置，单点线性探针不一定能稳定读出答案。

第二类方法是强制提前作答。研究者在模型生成到某个中间步骤时截断思维链，然后提示同一个模型立刻给出最终选项。这个方法用于观察模型在该时刻是否已经具备回答能力。如果模型在很早的位置被打断后仍能给出最终答案，说明答案信息可能已经存在于内部状态中，即使原本的思维链文本还没有明确写出答案。

第三类方法是思维链监测器。监测器是另一个语言模型，它只能阅读被测模型已经生成的部分思维链，并判断这些文本是否足以推出最终选项；如果信息不足，监测器可以回答“不足以判断”。这三种方法的信息权限不同：激活探针能读内部表示，强制提前作答能把潜在信息压出来，思维链监测器只能看外部文字。因此，三者之间的时间差就能反映内部信念和外显文本之间的错位。

4. 实验设置：模型、数据集与任务难度

论文主要分析 DeepSeek-R1 671B 和 GPT-OSS 120B 两个开放权重推理模型，并在部分实验中加入 DeepSeek-R1 的 Qwen2.5 蒸馏模型，用来观察模型规模和能力的影响。实验任务包括 MMLU-Redux 2.0 和 GPQA-Diamond。原文说明，MMLU-Redux 2.0 原始包含 5700 道题，经过错误标注过滤后使用 5280 道；GPQA-Diamond 包含 198 道更困难的科学问题。MMLU-Redux 覆盖多个学科，很多题目偏知识回忆或概念识别；GPQA-Diamond 则更难，包含需要较强专业知识和多步推理的科学问题。

这两个数据集都可以表示为四选一，因此适合用统一方式比较模型在不同推理位置上对 A、B、C、D 的预测。论文在探针实验中使用了 MMLU 的保留测试集 N=528 和 GPQA-Diamond 的测试集 N=157，并提到用 20 道 GPQA 题进行少量微调时，对整体探针准确率改善很小，因此主要报告直接迁移的结果。论文关注的不是“模型最后是否答对”这么简单，而是：在思维链生成到 0%、20%、50%、80% 等不同位置时，激活探针、强制提前作答和文本监测器分别能多早看出模型最终会选什么。

5. 主要发现：容易题更容易出现“表演式推理”

论文的一个核心发现是，表演式思维链具有明显的任务差异。在 MMLU-Redux 这类相对容易、偏知识回忆的题目上，激活探针和强制提前作答常常在思维链很早的位置就能较高准确率地预测模型最终答案；但思维链监测器要晚得多才从文本中看到答案线索。这说明模型内部答案信息已经出现，后续较长的思维链更多是在铺陈、解释或延迟显露答案，而不是持续提高内部准确率。原文还指出，DeepSeek-R1 的后半层、GPT-OSS 的后四分之三层都能较好解码最终答案；附录对比中，注意力探针测试准确率为 87.98%，明显高于传统线性探针的 31.85%。

论文用探针/强制提前作答与文本监测器之间的信息增益差距来刻画“表演性”。在 DeepSeek-R1 上，MMLU 的“探针相对监测器”差距为 0.417，“强制作答相对监测器”差距为 0.505；而 GPQA-Diamond 上对应数值只有 0.012 和 0.010，接近 0，表示文本监测器和内部信号的增长更同步。GPT-OSS 上也有类似趋势：MMLU 的两项差距为 0.435 和 0.334，GPQA-Diamond 为 0.227 和 0.185。直观理解就是：对一些偏记忆的题，模型可能在读完题后很快就知道答案，只是仍继续写出一段像推理过程的文字；这些文字未必完全无用，但它们不一定是模型形成答案所必需的计算过程。

相反，在 GPQA-Diamond 这类更困难的多跳科学问题上，三种方法的准确率通常从接近随机水平开始，并随着推理过程逐步提高。探针和强制提前作答有时仍领先文本监测器，但领先幅度更小，三者增长趋势更一致。这意味着在困难任务中，模型确实可能需要通过生成过程整合条件、排除错误路径并逐步形成答案。此时长思维链更可能承载真实计算或推理，而不只是把已经形成的答案包装成推导。

6. 模型规模：能力越强，不一定越“坦白”

论文还讨论了模型规模的影响。在同样的 MMLU 任务上，更大的模型往往更早就能从内部激活中解码出最终答案，说明它们可能更多依赖参数中已有的知识和先验。较小的蒸馏模型在前半段推理中探针准确率更低，后面才逐渐提升，表现得更像“需要现场推理”。

这并不表示小模型一定更可靠，也不表示大模型的思维链一定不忠实。更准确的结论是：“是否表演式”不是模型的固定属性，而是模型能力、任务难度和答案是否容易被已有知识快速确定之间的交互结果。能力强的模型在容易题上可能很早形成答案，因此外显推理更容易滞后；困难题则会迫使模型在生成过程中实际计算，文本和内部信念更可能同步变化。

7. 转折点：长思维链并不总是表演

论文没有把所有长思维链都归为表演。作者特别研究了思维链中的“转折点”，例如回溯、重新考虑、突然意识到某个条件的重要性，或类似“啊，原来应该这样”的领悟时刻。如果模型内部已经非常确信答案，却仍频繁写出这些转折点，那么这些转折点也可能只是表演；但实验结果并不支持这种简单判断。

在 DeepSeek-R1 的 MMLU 样本中，作者把“从思维链开始时探针置信度就达到 90% 以上，并且之后从未低于 90%”的回答标为高置信回答；在 522 条可分析回答中，有 215 条属于这一类。结果显示，高置信

回答中的转折点更少：所有转折点按步骤归一化后的总比例为 0.020，而非高置信回答为 0.045，约为前者两倍多。分类型看，reconsideration 的比例从 0.015 上升到 0.033，realization 从 0.004 上升到 0.009，backtrack 从 0.001 上升到 0.003。这说明许多回溯和重新考虑并非纯粹表演，而可能对应真实的不确定性消解。

不过，论文对转折点也保持谨慎。作者进一步检查局部时间关系：内部探针置信度发生较大跳变之后，文本中是否一定很快出现回溯或领悟？或者文本先出现转折点之后，内部信念是否一定随之变化？结果并不稳定。不同模型、不同数据集和不同窗口设置下，局部对应关系会变化。因此，比较稳妥的结论是：转折点在统计上更常出现在不确定回答中，说明它们常常与真实推理有关；但不能把每一个转折点都机械地解释为某个精确内部信念变化的直接标记。

8. 讨论重点：思维链监控的价值与边界

论文讨论部分的重点不是简单否定思维链，而是要求更谨慎地解释思维链。对于容易题，模型可能已经在内部形成答案，但继续生成长篇推导；只读文本的监测器直到答案接近显露时才知道模型要选什么。在这种情形下，思维链文本可能无法可靠用于发现“模型是否已经承诺了某个答案”、衡量模型当前不确定性，或审计模型为什么做出决策。

对于困难题，情况不同。如果答案信息确实在生成过程中逐步出现，那么文本推理可能更好地追踪模型内部信念变化。换句话说，思维链监控并不是完全无效，而是有适用条件：当任务需要真实的逐步计算，且模型在文本中确实逐步显露线索时，文本监控更有价值；当任务主要依赖已有知识或快速回忆时，模型内部可能早于文本形成答案，文本监控就会滞后。

论文还从“合作交流”的角度解释了这种现象。普通交流中，一个合作的说话者会尽量说相关、真实、适量、清楚的话；但推理模型的训练目标主要是完成任务、得到正确答案，而不是作为一个完全合作的“自我解释者”向监控者汇报内部状态。因此，它写出的思维链可能是相关的，也可能局部上看起来有依据，却仍然过长、过于包装化，或者没有及时暴露早已形成的内部承诺。这正是只依赖思维链文本做安全监控时容易遇到的风险。

这个讨论对考试和工程实践都很重要。不能因为模型写了很长的解释，就自动认为它经历了同样长的内部推理；也不能因为某些思维链存在表演成分，就否定所有思维链的可解释价值。更合理的做法是把思维链文本看作一个有用但不完备的证据源，并结合内部激活、强制提前作答、任务难度和答案稳定性等信号一起判断。

9. 工程启发：探针置信度早退

论文提出的工程启发是“基于探针置信度的早退”。如果注意力探针经过校准，能够在中间位置以较高置信度预测最终答案，并且答案在连续步骤中保持稳定，那么系统可以提前停止生成，不再让模型继续写完整思维链。这种策略把探针从解释工具变成自适应计算工具：简单题少生成，复杂题多生成。

论文报告，在 MMLU 上，探针置信度达到较高阈值时可以在保留接近原始性能的同时节省大量 token。更具体地说，DeepSeek-R1 在 MMLU 上以 95% 置信度阈值早退时，可以保留原始性能的 97%，同时节省 68% 的生成 token；摘要中进一步概括，MMLU 上最多可节省约 80% 的生成 token。在 GPQA-Diamond 上，80% 置信度阈值早退可以同样保留约 97% 的性能，并节省 50% 的 token；摘要中概括为最多约 30% 的生成 token。这个差异也呼应了前面的讨论：容易题更容易提前确定答案，因此早退空间更大；困难题需要更多真实推理，因此不能过早停止。

早退并不等于越早停止越好。若探针没有校准，或者任务确实需要多步推理，过早停止可能让模型错过后续关键计算，损害准确率。更合理的策略是设置较高置信度阈值，例如要求探针置信度达到某一水平，同时答案在相邻步骤中保持稳定；如果置信度低、答案频繁改变，或任务属于高难度多跳推理，就继续生成。由此可见，探针早退的本质不是粗暴截断思维链，而是在“内部信号足够可靠”时减少不必要的文本生成。

10. 总结：区分两种长思维链

总体而言，这篇论文并没有否定思维链的价值，而是提醒人们区分两种情况：一种是模型已经内部确定答案，后续思维链主要起解释、包装或延迟显露作用；另一种是模型确实在生成过程中逐步形成答案，思维链更忠实地反映了推理过程。理解这一区分，对评测推理模型、设计安全监控、节省推理成本都有意义。

对于本题阅读，最重要的结论可以概括为三点：第一，表演式思维链关注内部信念与外显文本的错位；第二，激活探针、强制提前作答和思维链监测器可以从不同视角比较这种错位；第三，校准良好的探针可以支持置信度早退，但必须结合任务难度和答案稳定性谨慎使用。

---

[1] 某道 MMLU 类题目中，模型生成到思维链 20% 位置时，激活探针已经以较高置信度预测最终选项为 B；此时强制模型提前作答，模型也输出 B；但只阅读已生成文本的思维链监测器回答“不足以判断”。直到思维链接近结尾时，文本才明确支持 B。根据阅读材料，对这一现象最合理的解释是：

A. 文本监测器难以判断时，通常说明模型内部也尚未形成明确答案倾向

B. 激活探针和强制提前作答都可能受到随机噪声影响，因此内部信号的参考价值较有限

C. 模型内部答案信息可能早于外显文本出现，说明思维链文本相对内部信念存在滞后

D. 该现象主要说明模型最终选项为 B，与思维链忠实性的关系较弱

答案： C

> **考纲对应：** 九. 复杂推理 - 长思维链模型；十一. 模型评测 - 评测范式与方法
> **题目标注：** 难度 3；考核方式：论文阅读与实验解释；聚焦：激活探针、强制提前作答、思维链监测器的信息权限差异
> **答案解析：** 材料强调三种方法观察“模型何时知道答案”的信息权限不同。激活探针能读取内部表示，强制提前作答能迫使模型外显当前可用信息，文本监测器只能读已经生成的文字。若前两者明显早于文本监测器预测出同一答案，最合理解释是内部答案信息早于文本显露，体现外显思维链相对内部信念的滞后。

[2] 阅读材料给出 DeepSeek-R1 上的“探针相对监测器”差距：MMLU 为 0.417，GPQA-Diamond 为 0.012；“强制作答相对监测器”差距：MMLU 为 0.505，GPQA-Diamond 为 0.010。对这些数字的解读，最准确的是：

A. MMLU 上内部信号显著领先文本信号，更支持“容易知识题上更容易出现表演式思维链”的结论

B. GPQA-Diamond 上差距更小，说明 GPQA-Diamond 中真实推理需求较低

C. 差距较大通常说明模型最终答案更可能不可靠

D. 两组数据更倾向说明文本监测器比激活探针更稳定

答案： A

> **考纲对应：** 九. 复杂推理 - 长思维链模型；十一. 模型评测 - 评测指标 / 评测范式与方法
> **题目标注：** 难度 3；考核方式：论文阅读与数据解释；聚焦：表演式思维链的信息增益差距、任务难度比较
> **答案解析：** MMLU 上两类内部或提前外显方法相对文本监测器的差距很大，说明模型内部或可被迫外

> 显的答案信息比文本线索出现得早；GPQA-Diamond 上差距接近 0，说明文本与内部信号更同步。这支持材料中的核心结论：较容易、偏知识回忆的题更容易出现表演式思维链，而困难多跳题更可能需要真实逐步推理。

[3] 某系统希望使用探针置信度做早退。连续 4 个检查点的探针输出如下，其中阈值设为 0.90，要求“同一答案连续两个检查点置信度均不低于阈值”才允许早退：

检查点	预测答案	置信度
20%	B	0.86
40%	C	0.91
60%	B	0.93
80%	B	0.94

根据阅读材料中“高置信度 + 答案稳定性”的原则，以下策略最合理的是：

A. 在 40% 处早退，因为此时置信度已经超过阈值，可以优先节省后续 token

B. 在 60% 处早退，因为 B 的置信度超过阈值，并且相较 40% 检查点已经发生修正

C. 在 80% 处早退，因为 B 在相邻两个检查点均超过阈值且答案稳定

D. 从 20% 处早退，因为 0.86 已接近阈值且可以最大化节省 token

答案： C

> **考纲对应：** 七. 解码与部署 - 解码加速算法与实践（早退机制）；九. 复杂推理 - 长思维链模型
> **题目标注：** 难度 3；考核方式：论文阅读与策略判断；聚焦：探针置信度早退、答案稳定性、阈值策略
> **答案解析：** 材料强调早退不能只看某一时刻置信度是否超过阈值，还要看答案是否稳定。40% 处虽然置信度超过 0.90，但预测为 C，后续又变为 B，不满足稳定性；60% 处 B 刚刚超过阈值，也尚未形成连续稳定证据；80% 处 B 在 60% 和 80% 两个检查点均超过阈值，因此更符合“高置信度 + 答案稳定性”的早退原则。