成人组第一轮试题解析

本解析包含 30 道单项选择题和 3 道材料题，共 100 分。每道题后附参考答案与解析。

题目中的数量、公式、精度和操作顺序以题面为准。除特别说明，ln 为自然对数；不要求实际运行代码。前两道代码材料题各空按其余空均已正确填写来作答，所选代码须满足全部输入约定，而不仅适用于示例。第三道材料题依据给出的论文中文节译作答。

一、单项选择题（共 30 题，每题 2 分，共 60 分）

第 1 题｜强化学习奖励与真实任务指标

强化学习中的奖励是训练使用的打分信号，任务指标衡量实际目标，两者的定义可能不同。例如，“答得更长”可以增加某个奖励函数的分数，而固定测试集正确率只计算回答是否正确。判断训练是否改善实际任务，需要同时查看奖励和任务指标；两者的因果关系还需实验检验。

某训练奖励会随回答长度增加。两次检查点在同一固定测试集上，平均奖励由1.2升至2.1，任务正确率由74%降至67%，回答也变长。仅根据这些证据，以下判断最合理的是?

A. 奖励提升尚未转化为测试任务改善，应检查长度奖励是否偏离目标，并用独立任务指标验证修改

B. 奖励上升已证明推理能力提升，正确率下降无需进一步分析

C. 只要继续提高长度奖励，就能保证正确率恢复并超过原值

D. 这两个检查点已证明所有强化学习方法都无法改善模型推理

答案： A

解析： 奖励和目标指标在这里出现背离。应检查代理奖励与任务目标的一致性，并通过额外实验判断原因。现有观测不能证明增加长度一定改善推理，也不能推广为所有强化学习方法的结论。

第 2 题｜激活检查点的成本变化

激活是模型前向计算产生、反向计算可能需要的中间结果。激活检查点只保留其中一部分，需要时重新执行相应前向计算来恢复其余结果。这一机制本身不改变模型参数量。本题总存储由固定存储和保存的激活两部分相加得到；额外重算也会影响训练耗时。

某模型训练时，权重、梯度、优化器状态及其他固定存储合计32GB，保存的激活占20GB。启用激活检查点后，保存的激活降至8GB，其他存储不变。以下哪项判断正确?

A. 总存储降为 40GB；因此每步训练时间必然按 40/52 的比例缩短

B. 总存储降为 40GB；这说明可训练参数量也减少了 12GB 对应的参数

C. 总存储由 52GB 降为 40GB；反向阶段可能需要额外重算，模型参数量不因此减少

D. 总存储由 52GB 降为 20.8GB；所有存储都按激活的下降比例缩减

答案： C

解析： 启用后是 32 + 8 = 40GB，节省 12GB。激活检查点以部分重算换取激活存储下降，不会自动减少参数量，也不能仅凭存储变化推导训练时间同比缩短。

第 3 题｜指令数据筛选与配比

清洗后的指令数据有问答 900 条、抽取 500 条、澄清 400 条。只能丢弃现有记录，不得复制或新增；要求最终三类数量严格为 3：2：1。最多可以保留多少条，哪个类别限制了这个最大值?

A. 1500 条，抽取类

B. 1800 条，问答类

C. 1600 条，澄清类

D. 1200 条，抽取类

答案： A

解析： 设三类保留 3t、2t、t 条，则 $t \leqslant \min(900/3, 500/2, 400) = 250$。保留 $750 + 500 + 250 = 1500$ 条，抽取类达到上限。

第 4 题｜有效目标位置与损失分母

下列损失和有效标记都已经按“预测下一个 token”完成对齐，无需再次移动。样本甲的逐位置损失为 [0.2,0.4,0.6,9.0]，有效标记为 [0,1,1,0]；样本乙为 [0.5,1.0,1.5,2.0]，标记为 [1,1,1,0]。要求所有有效目标 token 等权，平均损失是多少?

A. 1.90

B. 0.50

C. 0.75

D. 0.80

答案： D

解析： 有效损失之和为 $0.4 + 0.6 + 0.5 + 1.0 + 1.5 = 4$，共 5 个有效 token，故为 0.8。先对每条样本求均值再平均会得到 0.75，属于另一种权重定义。

第 5 题｜AdamW 的解耦权重衰减

AdamW 将权重衰减与基于梯度的更新分开处理。对实施衰减且有梯度张量的参数，衰减步骤为 $w \leftarrow (1 - \text{学习率} \times \text{衰减系数}) w$。本题中，若当前梯度张量、一阶矩状态和二阶矩状态都为零，则基于梯度的更新项为零；衰减步骤仍按上述规则执行。

某标量参数 $w = 2$，本步梯度张量为 0，一阶和二阶矩状态也都为 0；学习率为 0.01，权重衰减系数为 0.1。该参数属于实施衰减的参数组，且无其他更新项。更新后的 w 是多少?

A. 1.998

B. 1.999

C. 1.800

D. 2.000

答案： A

解析： 自适应梯度更新项为 0，但解耦衰减仍执行。$w = (1 - 0.01 \times 0.1) \times 2 = 1.998$。题面给出零张量而非 grad=None，并明确动量状态为零，排除了跳过更新和历史动量的歧义。

第 6 题｜智能体工具调用与幂等重试

工具调用超时表示客户端未按时收到响应；服务端可能尚未执行，也可能已经执行而响应未返回。本题“创建工单”接口支持幂等键 idempotency_key：相同 key 的重复提交只创建一张工单，且可按该 key 查询状态。不同 key 被当作不同逻辑操作，可能各自创建工单。

智能体调用“创建工单”工具后发生超时，不知道服务端是否已成功创建。对于这一次逻辑操作，最合适的处理是？

A. 每次超时都换一个新 key 重试，以便让服务端把请求视为全新操作

B. 超时一定说明创建成功，因此向用户报告已成功且不再核实

C. 保留原 key，查询该 key 的状态，必要时用原 key 重试，并据实际结果继续

D. 超时一定说明服务端没有执行，因此直接再创建一张且无需查询

答案： C

解析： 超时只表示客户端未及时得到结果，不能判断是否已经提交。使用同一幂等键进行查询和重试，能在题设接口约定下避免因重复请求多创建工单。

第 7 题｜Prefill 与解码性能诊断

生成服务先处理输入上下文，这一阶段通常称为 prefill，随后逐个生成输出 token。首 token 延迟指从请求开始到第一个输出出现的时间；后续生成速率只统计首 token 之后的生成阶段。两项指标覆盖的时间段不同；输出正确率需要另行评测。

同一模型、设备和 batch 下，没有排队等待，输出长度固定。把输入从4k token增到32k token后，首token延迟从0.2秒变为2秒，而首token之后的生成速率仍约为30 token/秒。以下哪项是这些测量最直接支持的判断？

A. 逐 token 解码速率下降为原来的 1/10，因为首 token 延迟增加了 10 倍

B. 首 token 前的处理阶段耗时增加，应优先检查 prefill 等该阶段工作；未观察到逐 token 解码速率同步变慢

C. 模型输出的正确率下降为原来的 1/10，因为首 token 延迟增加了 10 倍

D. 所有阶段耗时都增加 10 倍，因此任意长度输出的总响应时间必然增加 10 倍

答案： B

解析： 首 token 延迟与后续生成速率是不同指标。题设显示主要变化发生在首 token 前，但仅凭这些测量仍不能把这一阶段内部的全部耗时归给某个单一算子。输出质量也没有被测量。

第 8 题｜FlashAttention 的计算与访存

标准稠密注意力把各查询与全部可见键的得分经 softmax 归一化，再对值向量加权。FlashAttention 将这些运算分块执行，逐块维护归一化统计量和加权结果，使最终结果在数学上等价于标准计算。已处理块的完整得分无需一直保留在外部显存中；所有可见查询与键的交互仍参与计算。

在使用相同 Q、K、V 和掩码时，一个标准稠密注意力实现换成 FlashAttention。测试发现输出仅有浮点舍入差异，显存峰值下降。以下哪项最准确地解释这一现象？

A. 通过分块和在线归一化减少显存读写，避免把完整注意力矩阵写入显存；稠密注意力的数学目标保持不变

B. 取消 softmax，将原始 QK^T 直接乘 V，因此输出在所有输入上都保持相同

C. 对任意序列长度，所有查询与键之间的稠密交互数都从平方级降为线性级

D. 只计算每个查询概率最高的一个键，所以注意力从稠密精确计算变成稀疏近似计算

答案：A

解析： FlashAttention 的核心是通过分块、在线 softmax 和更好的片上存储利用降低高带宽显存的读写开销，计算精确注意力。它不等于只保留一个键，也没有把标准稠密注意力的交互数量改成线性级。

第 9 题｜按来源分组划分与交叉验证

共有 240 个来源族，每族 5 条记录，同族记录不得跨集合。先固定留出 48 个族作为测试集，再将其余族均匀分为 8 折。某一轮取 1 折验证、7 折训练；此后只从该轮训练集中删除 6 个无效族，且不重新分配。该轮最终训练、验证、测试记录数分别是多少？

A. 810、150、240

B. 810、120、240

C. 840、120、240

D. 864、96、240

答案：B

解析： 测试为 $48\times5=240$ 条。剩余 192 族，每折 24 族；验证 120 条，初始训练 $7\times24\times5=840$ 条，删除 $6\times5=30$ 条后为 810 条。验证和测试不变。

第 10 题｜分组评估与准入约束

测试集分为甲组 90 条、乙组 10 条。模型 P 分别答对 81、4 条，模型 Q 分别答对 72、7 条。宏平均准确率定义为两个组准确率的等权平均。部署准入要求每组准确率都不低于 65%。以下哪项给出了正确的宏平均值及符合准入要求的模型?

A. P 为 85%，Q 为 79%；P 和 Q 均不符合准入要求

B. P 为 65%，Q 为 75%；只有 Q 符合准入要求

C. P 为 85%，Q 为 79%；只有 P 符合准入要求

D. P 为 65%，Q 为 75%；P 和 Q 均符合准入要求

答案：B

解析： P 的组准确率为 90%、40%，宏平均 65%；Q 为 80%、70%，宏平均 75%。P、Q 的总体准确率分别为 85/100 = 85%、79/100 = 79%，分母都是 100。总体准确率按各组样本数加权，宏平均给甲、乙两组相同权重。准入要求每组准确率都达到 65%，只有 Q 合格。

第 11 题｜自适应推理与预算核算

对 200 道题均先生成 100 token，其中 120 道就此结束、答对 96 道；另 80 道每道再生成 400 token，最终答对 72 道。按生成 token 计成本，不计其他开销。与“每道都生成 500 token”的固定预算相比，这次自适应方案的正确率和 token 节省比例分别是多少?

A. 80%；48%

B. 84%；52%

C. 90%；60%

D. 84%；48%

答案：D

解析： 正确率为 (96 + 72)/200 = 84%。实际使用 200×100 + 80×400 = 52000 token，固定预算为 100000，节省 48%。题面未提供固定预算方案的正确率，不能据此断言节省预算而准确率完全无损。

第 12 题｜Prompt 的上下文容量约束

模型上下文容量为 6000 token。系统指令占 400，历史有 5 段且每段 300，当前问题占 500，预留输出 800、工具结果 100。每个完整检索片段占 500 token，并额外占 20 token 分隔符。除此之外无其他开销，不删历史、不截断片段。最多能加入多少个检索片段，此后剩余多少未分配 token?

A. 5 个；200 token

B. 4 个；620 token

C. 5 个；100 token

D. 6 个；380 token

答案：C

解析： 固定部分为 400 + 5×300 + 500 + 800 + 100 = 3300，可用 6000 − 3300 = 2700。每段计 520，5 段占 2600，余 100；6 段需 3120，超过可用容量。因此最多 5 段，余 100 token。余 200 漏算分隔符；4 段虽能放下但不是最多；6 段余 380 漏掉了输出预留。

第 13 题｜前缀与局部因果掩码

序列由 4 个前缀 token 和其后 3 个回答 token 构成。每个前缀 token 可看全部 4 个前缀，不能看回答；每个回答 token 可看全部前缀，以及“自己和紧邻的前一个回答 token”（若存在）。计入看自己的位置，整个掩码允许的查询—键位置对共有多少个?

A. 49

B. 33

C. 34

D. 28

答案：B

解析： 前缀部分有 $4\times4=16$ 个位置对；3 个回答位置依次可看 5、6、6 个位置，共 $16+5+6+6=33$。这里统计的是掩码允许的位置对，不是具体实现的 FLOPs。

第 14 题｜Beam Search 的累计概率比较

生成恰好两个 token，不使用 EOS 或长度惩罚，按条件概率乘积从大到小排序，beam width = 2。第 1 步 P(A)、P(B)、P(C) 分别为 0.5、0.3、0.2，保留 A 和 B。第 2 步，A 后 x、y、z 的概率为 0.4、0.35、0.25，B 后为 0.8、0.1、0.1。从这两个保留前缀的全部扩展中选出最终两个 beam，依次是什么?

A. Ax（0.20）、Ay（0.175）

B. Bx（0.24）、Ax（0.20）

C. Ax（0.40）、Bx（0.80）

D. Bx（0.80）、By（0.10）

答案：B

解析： A 分支的累计概率为 0.20、0.175、0.125；B 分支为 0.24、0.03、0.03。降序前两项是 Bx 和 Ax。每一步只按局部最大值选择的贪心路径则为 Ax。

第 15 题｜温度变换与禁用候选

三个候选 token 的 logit 为 $[0,2\ \ln2,2\ \ln3]$。先除以温度 $T=2$，再将第 2 个候选禁用为负无穷，最后做 softmax。第 3 个候选的最终概率是多少?

A. 9/10

B. 1/2

C. 3/5

D. 3/4

答案：D

解析： 除以温度后为 $[0,\ln2,\ln3]$，禁用第 2 个后，剩余指数权重为 1 和 3。第 3 个概率为 $3/(1+3)=3/4$。

第 16 题｜连续采样截断与重新归一化

候选 a、b、c、d、e 的概率依次为 0.38、0.27、0.18、0.12、0.05。先执行 top-k = 3 并重新归一化，再对归一化后的概率执行 top-p = 0.75：按降序保留累计概率首次达到或超过阈值的最短前缀，最后再次归一化。b 的最终概率是多少？结果保留分数，无需计算小数近似值。

A. 27/83

B. 27/65

C. 18/83

D. 27/100

答案： B

解析： top-k 后总质量为 0.83。a 的归一化概率为 38/83 < 0.75，a + b 为 65/83 ⩾ 0.75，因此保留 a、b。比较阈值时可用 0.75 = 3/4，再交叉相乘，无需求小数近似。再次归一化后 b 为 27/65。

第 17 题｜优化器状态分片的显存估算

训练存储包括模型权重、梯度和优化器为每个参数维护的状态。本题优化器状态包含一阶矩、二阶矩各一个数。“仅将优化器状态均匀分片”表示每张卡完整保留权重和梯度，而两个矩状态均由各卡分担。BF16 每个数占2字节，FP32 每个数占4字节；$1B = 10^9$ ，$1GB = 10^9$ 字节。

某简化训练系统有1B参数，4张GPU采用上述分片策略。权重和梯度均用BF16，两个矩状态均用FP32。没有额外FP32主权重副本，也不计激活、通信缓冲和框架开销。每张卡这三类状态合计多少GB?

A. 12GB

B. 4GB

C. 6GB

D. 3GB

答案： C

解析： 每卡权重 2GB、梯度 2GB，两个矩状态合计 8GB，分片后每卡 2GB。总计 2 + 2 + 2 = 6GB。只有题设指定的优化器状态被分片。

第 18 题｜DPO 的相对偏好间隔

DPO 比较同一输入下优选与劣选回答的相对偏好。设当前模型给回答的概率为 π、固定参考模型的对数概率差为 r，定义 $\Delta = [\log \pi(\text{优选回答}) - \log \pi(\text{劣选回答})] - r$，损失 $L = -\log \sigma(\beta\Delta)$，$\beta > 0$。σ 的取值在0和1之间且严格递增；log 和 exp 也严格递增。本题 log 为自然对数，概率 = exp(对数概率)。

同一偏好对的 $r = 0.5$，β 固定不变。训练前两回答的 log π 分别为 −2、−3，训练后分别为 −2.5、−4。以下判断正确的是?

A. Δ 从 0.5 降至 −0.5，损失上升，优选回答自身的概率下降了

B. 只要优选回答自身的概率下降，DPO 损失就必然上升，无需比较劣选回答

C. Δ 从 1.0 增至 1.5，损失下降，优选回答自身的概率上升了

D. Δ 从 0.5 增至 1.0，损失下降，但优选回答自身的概率下降了

答案： D

解析： 训练前 $\Delta = (-2 + 3) - 0.5 = 0.5$；训练后 $\Delta = (-2.5 + 4) - 0.5 = 1$。损失随 Δ 增大而减小，但 exp(−2.5) 小于 exp(−2)。相对偏好改善不等于优选回答的绝对概率一定增大。

第 19 题｜大模型评审的顺序偏差

模型裁判接收两个候选回答，并返回所选回答的呈现位置。交换位置只改变显示顺序，回答内容与其所属模型不变。“顺序敏感”指判断受呈现顺序影响；“盲评”指评审者看不到回答来自哪个模型。比较模型时，需要将位置判决对应回回答的真实来源，并检查影响判决的因素。

用大模型裁判比较同一批固定回答对。随机安排回答位置后，第一轮裁判有80%的比较选择了位置1；把每对回答的位置互换后，仍有78%的比较选择位置1。以下哪种做法最有助于得到可信的模型优劣结论?

A. 对同一回答对核对换位前后的判决一致性，报告顺序敏感性，并以盲评等方式复核

B. 把 80% 直接记为第一个模型的胜率，无需记录每次位置对应的模型身份

C. 选择更长的回答作为正确答案，以此直接替代顺序偏差检查

D. 只保留第一轮判决，删除第二轮，因为换位后的结果没有评估价值

答案： A

解析： 换位后仍大量偏向位置 1，提示明显的顺序敏感性，需要做回答对层面的检查。位置胜率不是某个固定模型的胜率；仅凭这两个汇总数也不能完成所有偏差的因果归因。

第 20 题｜混合精度训练与梯度裁剪

混合精度训练中，GradScaler 将损失乘缩放因子 S，反向传播得到的梯度也相应乘 S。unscale_ 将当前梯度除以 S。范数裁剪的规则是：若当前梯度的二范数 n 大于上限 M，就将各分量乘 M/n，否则不变；二维向量的二范数是两个分量平方和的平方根。scaler.step 执行参数更新：若本轮尚未反缩放则先反缩放，已反缩放则不重复。scaler.update 在更新后调整后续步骤的缩放因子。本题梯度均为有限值。

某一步的缩放因子为 1024，未缩放的梯度向量为 (3,4)，希望按真实梯度范数上限 1 裁剪。反向传播已经完成，下一段操作应如何安排？忽略舍入误差。

A. 先 unscale_，再按上限 1 裁剪，再 scaler.step，最后 scaler.update；裁剪后梯度为 (0.6,0.8)

B. 先 scaler.step，再 unscale_ 和裁剪；本次更新已受到上限 1 的约束

C. 只需把学习率除以 1024，无需反缩放或裁剪；所得更新等同于按范数上限 1 裁剪

D. 先按上限 1 裁剪缩放后的梯度，再 unscale_，再 scaler.step；裁剪后用于更新的梯度为 (0.6,0.8)

答案： A

解析： 真实梯度范数为 5，反缩放后乘 1/5 得 (0.6,0.8)。如果先把缩放后的梯度裁到范数 1，再反缩放，用于更新的梯度范数会变成 1/1024。裁剪必须发生在参数更新之前。

第 21 题｜共享权重与混合精度存储

某简化模型只有两部分：词嵌入矩阵 E 的形状为 [200,40]，输出投影与 E 完全共享权重，不另存副本且无偏置；另有 40→10 的仿射层，包含权重和 10 个偏置。E 以 INT4 紧密打包，每参数 4 bit；仿射层以 FP16 保存，每参数 2 字节，1 字节 = 8 bit。不计量化元数据、对齐、运行时副本或其他存储。共需多少字节？

A. 4800 字节

B. 4205 字节

C. 4820 字节

D. 8820 字节

答案： C

解析： E 有 $200\times40=8000$ 个参数，INT4 每两个参数占 1 字节，因此占 4000 字节；共享输出投影不重复计数。仿射层有 $40\times10+10=410$ 个参数，占 $410\times2=820$ 字节。合计 $4000+820=4820$ 字节。4800 漏算偏置；4205 将仿射层也按 INT4 计；8820 重复存储了共享的 E。

第 22 题｜阈值选择与精确率召回率

8 条记录的分数从高到低为 [0.95,0.85,0.75,0.65,0.55,0.45,0.35,0.25]，真实标签依次为 [正,负,正,负,正,负,正,负]。分数≥阈值判为正，要求精确率 P≥0.60 且召回率 R≥0.70。P = TP/(TP + FP)，R = TP/(TP + FN)，F1 = 2TP/(2TP + FP + FN)。下列哪个“阈值与对应 F1”满足要求?

A. 阈值 0.50；F1 = 2/3

B. 阈值 0.30；F1 = 8/11

C. 阈值 0.70；F1 = 4/7

D. 阈值 0.90；F1 = 2/5

答案： A

解析： 阈值 0.50 时 TP = 3、FP = 2、FN = 1，P = 0.60、R = 0.75、F1 = 6/9 = 2/3。0.90 和 0.70 的召回率不足；0.30 时精确率为 4/7 < 0.60。其余选项的 F1 数值本身均正确，必须检查双重约束。

第 23 题｜困惑度上限与数据混合比例

同一模型使用相同tokenizer和评估规则。甲组固定有900个有效token，困惑度PPL = 4，必须全部保留；另有足量乙组数据，每批200个有效token，每批PPL均为16。所有批次互不重叠，只能加入完整批次，不能删改甲组或截取乙组。各组平均负对数似然NLL = ln(PPL)，整体PPL = exp(按有效token数加权的平均NLL)。要求混合后的整体PPL不超过8，最多可加入多少批乙组数据、共多少个乙组token?已知ln4 = 2ln2、ln8 = 3ln2、ln16 = 4ln2。

A. 5批；1000个

B. 2批；400个

C. 4批；800个

D. 批数没有上限；可加入任意多个

答案： C

解析： 设加入k批。由exp单调递增，(900×2 + 200k×4)/(900 + 200k)≤3，即200k≤900。k必须为整数，故最多4批、800个乙组token。k = 4时加权NLL为(50/17)ln2，小于3ln2；k = 5时为(58/19)ln2，大于3ln2。直接算术平均PPL会错误得到最多2批；将900÷200向上取整会错误得到5批。

第 24 题｜Batch 与梯度累积边界

每个 epoch 有 936 个训练样本，均匀分给 3 张数据并行 GPU，不重复、不补样本。每张卡按每批 32 个样本顺序分批，最后不足 32 个的尾批也保留；每张卡各处理一个 micro-batch 算作一个同步 micro-step。每累计 4 个同步 micro-step 更新一次参数并清空累积；epoch 末不足 4 步也立即更新并清空累积。训练 3 个 epoch，参数更新共几次？每个 epoch 最后一次更新汇总多少个全局样本？

A. 8 次；168 个

B. 6 次；192 个

C. 9 次；384 个

D. 9 次；168 个

答案：D

解析： 每卡有 $312 = 9 \times 32 + 24$ 个样本，即 10 个 micro-step。每 epoch 更新 $\text{ceil}(10/4) = 3$ 次，3 个 epoch 共 9 次。最后一组是每卡 $32 + 24 = 56$ 个样本，3 卡合计 168 个。

第 25 题｜注意力形状与加权结果

单头注意力中 Q 的形状为 [2,4]，K 为 [3,4]，V 为 [3,2]。采用 $\text{softmax}(QK^{\mathrm{T}}/\sqrt{4})V$，softmax 沿键所在维度逐行归一化，无掩码。QK^{T} 的第一行为 [0,2 ln2,2 ln3]，V 的三行为 (6,0)、(0,3)、(2,2)。输出形状及第一行分别是什么？

A. [2,3]；(2,2)

B. [2,2]；(2,2)

C. [2,2]；(8/3,5/3)

D. [2,2]；(12/7,15/7)

答案：B

解析： 分数矩阵形状为 [2,3]，除以 2 后第一行 softmax 权重为 (1,2,3)/6。加权 V 得 $((6+6)/6,(6+6)/6) = (2,2)$，输出形状为 [2,2]。

第 26 题｜扩展法则与计算预算

训练计算预算可用 $C = 6ND$ 估算，其中 N 为模型参数量，D 为该阶段处理的 token 数，C 的单位为 FLOPs（浮点运算次数）。多个训练阶段的计算量相加。全参数领域续训是继续更新已有模型的全部参数，本题仍用同一公式估算。忽略其他开销，$1B = 10^9$ 。

某团队先预训练，再做全参数领域续训，两阶段采用同一参数量 N。N 只能从 2B、3B、4B、6B 中选择；预训练 token 数 D_1 至少为 60B、至多为 90B，续训必须再处理 10B token。两阶段总计算量不得超过 1.8×10^{21} FLOPs。规划目标是先选择满足全部约束的最大 N，再在该 N 下使 D_1 最大。应选择哪组配置?

A. $N = 6B$，$D_1 = 40B$

B. $N = 3B$，$D_1 = 90B$

C. $N = 4B$，$D_1 = 65B$

D. $N = 4B$，$D_1 = 75B$

答案： C

解析： 两阶段预算为 $6N(D_1 + 10B) \leqslant 1.8\times10^{21}$。若 $N = 6B$，D_1 最多40B，达不到60B下限；$N = 4B$ 时，$D_1 \leqslant 65B$，满足60B～90B。故最大可行参数量为4B，再取$D_1 = 65B$。3B、90B虽可行，却不是最大参数量；4B、75B只在漏算续训开销时看似可行，实际总量为2.04×10^{21} FLOPs。

第 27 题｜RoPE 的相对位置性质

RoPE 用二维旋转把位置信息加入查询和键。在同一二维子空间中，把两者的位置增加同一偏移，等价于再施加同一个旋转矩阵 R，且 $R^TR = I$。两个列向量的点积记为 q^Tk。可直接使用矩阵恒等式 $(AB)^T = B^TA^T$，以及单位矩阵 I 乘任意向量等于该向量；无需记忆 RoPE 的具体构造公式。

某查询与键经过原有位置旋转后分别为 q 和 k。现将它们的位置同时增加同一偏移。忽略舍入误差，以下哪项必然成立?

A. 即使只移动查询、保持键的位置不变，点积也必然不变

B. $(Rq)^T(Rk) = q^Tk$，即该二维子空间对注意力点积的贡献不变

C. 共同平移后 $Rq = q$ 且 $Rk = k$，即向量的每个分量均不变

D. $(Rq)^T(Rk) = q^Tk/2$，即共同平移使点积减半

答案： B

解析： 由 $(Rq)^T(Rk) = q^TR^TRk = q^Tk$ 可知共同旋转保持点积。它不要求各分量不变，也不能推出仅改变一方位置时点积仍不变。

第 28 题｜LoRA 初始化与梯度传播

LoRA 冻结基础权重 W，训练两个较小矩阵 A、B，以 $\Delta W = BA$ 作为权重增量。本题省略常数缩放因子。记 G 为损失对 ΔW 的梯度，可直接使用 $\partial L/\partial A = B^{T}G$、$\partial L/\partial B = GA^{T}$ 判断两个分支能否开始学习，无需自行求导。

希望初始化时 $\Delta W = 0$，同时至少一个可训练分支在一般非退化输入与梯度下可以获得非零梯度并开始更新。除选项另有说明外，A、B 均参与训练。以下哪种初始化与训练设置满足要求？

A. A 随机初始化为非零矩阵，B 初始化为零矩阵

B. A、B 均初始化为零矩阵，只要学习率非零，第一步两个梯度就必然非零

C. A 随机初始化为非零矩阵，B 初始化为零矩阵；但整个训练中只更新 A，始终将 B 固定为零

D. A、B 均独立随机初始化为非零矩阵，并断言 BA 必为零

答案：A

解析： $B = 0$ 保证 $BA = 0$；A 非零时 GA^{T} 可以非零，B 因而能够开始学习。若 A、B 同时为零，两者梯度都为零；若始终固定 $B = 0$、只更新 A，则 A 的梯度 $B^{T}G$ 始终为零，无法开始学习。这里说的是一般非退化情况下可以获得非零梯度，不是对任意 G 的保证。

第 29 题｜MoE 路由容量与实际激活参数

MoE 将部分计算交给路由选中的专家。本题规定：每个专家独立按 token 编号从小到大接受请求，最多接受3个；超出的该次专家调用被丢弃，不补发、不重路由，其他专家对同一 token 的接受结果不受影响。所有 token 都执行共享部分，即使两次专家调用都被丢弃。每个 token 的实际激活量为“共享参数 + 实际调用的专家参数之和”；平均值以全部 token 为分母。$1B = 10^{9}$ 。

一个简化MoE模块有0.5B共享参数和A、B、C三个专家，每个专家含0.25B参数。6个token编号为1～6，最初各选两个不同专家，路由对依次为(A,B)、(A,C)、(A,B)、(A,C)、(B,C)、(B,C)。仍保留两个专家调用的token有几个？平均每token实际激活参数量是多少？

A. 4个；1.3125B

B. 5个；0.875B

C. 4个；0.875B

D. 4个；1.000B

答案：C

解析： A接受token 1、2、3；B接受1、3、5；C接受2、4、5。故1、2、3、5保留两个专家调用，共4个；token 4只剩C，token 6无专家调用。实际专家调用共9次。对全部6个token计算平均激活量：$(6\times0.5 + 9\times0.25)/6 = 0.875B$。不能仍按名义top-2计12次调用，也不能只用4个双专家token作平均的分母。

第 30 题｜GQA 与 KV 缓存

KV 缓存保存已处理 token 的键 K 和值 V，供后续生成使用。GQA 让多个查询头共享一组 KV 头。单请求缓存字节数为“2（K 和 V）×层数×KV 头数×每头维度×缓存 token 数×每元素字节数”。本题为输入和预留输出一次性分配缓存，每个请求的合计 token 数向上取整到 256 的整数倍；请求间不共享缓存。1 KiB = 1024 字节，1 MiB = 1024 KiB。

某 GQA 模型有 8 层、8 个查询头、2 个 KV 头，每头维度 32；K 和 V 的每个元素各占 2 字节。模型权重和其他开销已另外预留，系统可用于 KV 缓存的预算为 12 MiB。每个请求有 1000 个输入 token，并预留 200 个新增 token。无其他开销。单请求占用多少 MiB，最多同时容纳多少个这样的请求？

A. 2.5 MiB；4 个

B. 2 MiB；6 个

C. 10 MiB；1 个

D. 2.5 MiB；5 个

答案：A

解析： 每请求需预留 1000 + 200 = 1200 个 token，向上取整为 5×256 = 1280。每个 token 的 KV 占 2×8×2×32×2 = 2048 字节，即 2 KiB；每块 256 个 token 占 512 KiB，即 0.5 MiB。每请求占 5×0.5 = 2.5 MiB。4 个请求占 10 MiB，5 个占 12.5 MiB，超过 12 MiB 预算，因此最多 4 个。2 MiB 漏掉输出预留；10 MiB 误用查询头数代替 KV 头数；5 个请求则错误地将可容纳数量向上取整。

二、材料题（共 3 题、10 个小题，共 40 分）

材料题一（4 小题，共 16 分）

下面程序为 SFT 训练准备数据。按原顺序处理记录：先去除 instruction 和 output 首尾空白，再删除 source_id 属于预留来源集合 heldout_sources 的记录，以及清洗后任一字段为空的记录；按清洗后的 (instruction, output) 完整二元组去重，只保留首个合格记录，并保存清洗后的文本。随后按原顺序贪心分组，每组 token 总数不超过 max_tokens，恰好等于上限也允许。这里仅生成样本 ID 分组，不实现 token 拼接或组内 attention mask。

heldout_sources 是预留给验证或测试的 source_id 集合。每条记录有唯一整数 id、字符串 source_id/instruction/output、正整数 n_tokens。max_tokens 为正整数，所有记录的 n_tokens⩽max_tokens，长度已由题目给定，无需重新分词。同一清洗后二元组的长度一致；不改变大小写及内部空格。输入可以为空。第 31～33 题分别补全空 1～3，每个空只替换一个表达式，其余代码保持原样；第 34 题根据示例判断程序返回结果。

```
def pack_training(rows, heldout_sources, max_tokens):
    kept, seen = [], set()
    for row in rows:
        instruction = row["instruction"].strip()
        output = row["output"].strip()
        if __1__:
            continue
        key = __2__
        if key in seen:
            continue
        seen.add(key)
        cleaned_row = dict(row)
        cleaned_row["instruction"] = instruction
        cleaned_row["output"] = output
        kept.append(cleaned_row)

    batches, current, used = [], [], 0
    for row in kept:
        n = row["n_tokens"]
        if current and (__3__):
            batches.append((current, used))
            current, used = [], 0
        current.append(row["id"])
        used = used + n
    if current:
        batches.append((current, used))
    return batches
```

令 heldout_sources = {"test"}，max_tokens = 7。字符串两侧的引号用于显示首尾空格，不属于内容。输入记录如下。

id	source_id	instruction	output	n_tokens
1	test	"A"	"x"	3
2	train	" A "	"x"	3
3	train	"A"	"x "	3
4	train	"B"	" "	1
5	train	"A"	"y"	4
6	train	"C"	"z"	2
7	train	"D"	"w"	5
8	train	"E"	"v"	1
9	train	" "	"q"	1

第 31 题（空 1）

填入空 1，正确排除预留来源的记录，以及清洗后 instruction 或 output 为空的记录。

A. `row["source_id"] in heldout_sources or (not instruction and not output)`

B. `row["source_id"] in heldout_sources or not instruction or not output`

C. `row["source_id"] in heldout_sources and (not instruction or not output)`

D. `row["source_id"] not in heldout_sources or not instruction or not output`

答案： B

解析： 三种剔除条件之间为“或”。示例中剔除 1、4、9，过滤后还有 6 条（含尚未去重的 2、3）。必须先过滤再登记 seen，否则测试来源的重复文本可能错误挤掉训练记录。

第 32 题（空 2）

填入空 2，使去重严格遵守清洗后的完整二元组。

A. `instruction`

B. `output`

C. `(len(instruction), len(output))`

D. `(instruction, output)`

答案： D

解析： 2 和 3 清洗后相同，保留 2。5 与 2 的 instruction 相同但 output 不同，仍须保留。去重后 ID 为 [2,5,6,7,8]，共 5 条，不能只用一侧文本或长度作为去重键。

第 33 题（空 3）

填入空 3，使得只有放入当前记录会超过上限时，才先结束已有分组。

A. `used > max_tokens`

B. `n > max_tokens`

C. `used + n >= max_tokens`

D. `used + n > max_tokens`

答案： D

解析： 相等时应继续放入，所以使用严格大于。示例的 3 + 4 = 7、2 + 5 = 7 都可以同组；最终分组为 [2,5]、[6,7]、[8]。

第 34 题

假设空 1～3 均已正确填写，以表中的全部记录按原顺序作为 rows，执行 pack_training(rows, {"test"}, 7)。函数的返回结果是什么？每个二元组依次表示（组内记录 ID 列表，组内 token 总数）。

A. `[([2], 3), ([5, 6], 6), ([7, 8], 6)]`

B. `[([2, 6], 5), ([7, 8], 6)]`

C. `[([2, 5], 7), ([6, 7], 7), ([8], 1)]`

D. `[([2, 5], 7), ([6, 7], 7)]`

答案： C

解析： 先排除 1、4、9，再去除与 2 清洗后重复的 3，保留 ID 为 2、5、6、7、8，对应长度为 3、4、2、5、1。3 + 4 = 7，记录 2、5 同组；下一条会超过上限，另开一组。2 + 5 = 7，记录 6、7 同组；最后记录 8 单独成组，循环结束后仍须保存。返回 [([2, 5], 7), ([6, 7], 7), ([8], 1)]。加入当前记录前，若把 `used + n > max_tokens` 写成 `used + n >= max_tokens`，恰好能装满当前组的记录就会被提前划入下一组，本例得到 A。若先加入记录，恰好满 7 后立即封组，本例仍得到 C。只按 instruction 去重会误删记录 5；省略循环末尾的保存会漏掉最后一组。

考点小结： 过滤逻辑；去重单位；分组边界；程序执行与分组结果。

材料题二（3 小题，共 12 分）

以下程序先对每个查询的有效 token 做平均池化，再对查询向量和文档向量做 L2 归一化，以点积计算余弦相似度。只允许检索“全局有效且当前查询有访问权限”的文档，按得分从高到低返回每个查询的前 k 个文档 ID 及得分。文档 ID 是 docs 中从 0 开始的行号；返回的 indices、values 形状均为 [B,k]。

hidden 的形状为 [B,L,H]，token_mask 为 [B,L]，docs 为 [N,H]，doc_valid 为 [N]，allowed 为 [B,N]；B、L、H、N 均为正整数。三个掩码张量均为 bool：token_mask[b,l] 为 True 表示第 b 个查询的第 l 个 token 有效；doc_valid[j] 为 True 表示文档 j 全局有效；allowed[b,j] 为 True 表示第 b 个查询有权访问文档 j。所有张量位于同一设备，hidden 与 docs 使用相同的 float32 或 float64 类型，计算过程不使用自动混合精度。所有浮点输入均为有限数，并保证中间计算不发生溢出或下溢；忽略舍入差异。每个查询至少有一个有效 token，池化向量与每个文档向量的 L2 范数均大于 10^{-6} 。每个查询至少有 k 个可检索文档，k 为整数且 $1 \leqslant k \leqslant N$；同一查询的可检索文档之间无同分。dim 从 0 开始编号。

可使用以下接口说明：`unsqueeze(d)` 在指定位置插入长度为 1 的维度；`sum(d)` 沿指定维度求和，`keepdim=True` 保留该维度；`F.normalize(x, p=2, dim=d)` 沿指定维度做 L2 归一化。布尔张量的 `&`、`|`、`~` 分别表示逐元素与、或、非；`masked_fill(mask, value)` 在 mask 为 True 的位置填入 value。`torch.topk(x, k, dim=d)` 返回该维度上最大的 k 项及其索引，默认按值降序排列；`largest=False` 改为取最小的 k 项。`torch.sort` 返回该维度上的全部排序项及索引。广播按末尾维度对齐，相应维度相等或其中一个为 1 时可进行逐元素运算。

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

def retrieve(hidden, token_mask, docs,
             doc_valid, allowed, k):
    mask = token_mask.to(hidden.dtype)
    pooled = __1__
    queries = F.normalize(pooled, p=2, dim=-1)
    documents = F.normalize(docs, p=2, dim=-1)
    scores = queries @ documents.T
    scores = __2__
    values, indices = __3__
    return indices, values
```

示例设 B = 2、L = 3、H = 2、N = 5、k = 2。查询 1 的 hidden 为 [(2,0),(0,2),(8,8)]，查询 2 为 [(-2,0),(0,2),(-9,5)]，两行 token_mask 均为 [1,1,0]。文档 0～4 依次为 [(2,1),(1,0),(-1,0),(-1,-1),(0,1)]。doc_valid = [1,1,1,1,0]；allowed 两行分别为 [0,1,1,1,1] 和 [1,0,1,1,1]。示例中的 1、0 分别代表 True、False。

第 35 题（空 1）

填入空 1，仅对各查询的有效 token 求平均。

A. `(hidden * mask.unsqueeze(-1)).sum(1) / mask.sum(1, keepdim=True)`

B. `(hidden * mask.unsqueeze(-1)).sum(2) / mask.sum(1, keepdim=True)`

C. `hidden.mean(dim=1)`

D. `hidden[:, 0, :]`

答案：A

解析： 在 token 维（dim=1）加总，并除以各行有效 token 数；unsqueeze(-1) 使标记能广播到 H 维。示例池化后为 (1,1)、(−1,1)。填充向量值不是零，不能直接平均全部位置。

第 36 题（空 2）

填入空 2，使不可检索文档无法挤入结果，即使合法文档相似度为负也应正确。

A. `scores.masked_fill(~(doc_valid.unsqueeze(0) | allowed), float("-inf"))`

B. `scores * (doc_valid.unsqueeze(0) & allowed).to(scores.dtype)`

C. `scores.masked_fill(~(doc_valid.unsqueeze(0) & allowed), float("-inf"))`

D. `scores.masked_fill(~doc_valid.unsqueeze(0), float("-inf"))`

答案：C

解析： 合法集合是全局有效和当前有权限的交集，非法项填负无穷。乘以零会让非法文档得分变为 0，反而可能超过合法的负分文档。只看 doc_valid 会漏掉逐查询权限。

第 37 题（空 3）

填入空 3，沿文档维取每行最大的 k 个相似度。

A. `torch.topk(scores, k=k, dim=1, largest=False)`

B. `torch.sort(scores, dim=1, descending=True)`

C. `torch.topk(scores, k=k, dim=0)`

D. `torch.topk(scores, k=k, dim=1)`

答案：D

解析： scores 为 [B,N]，文档维是 1，topk 默认取最大值并排序。示例 ID 为 [[1,2],[2,3]]，分数为 [[1/√2,−1/√2],[1/√2,0]]。sort 会返回 N 项，不能在一般 k<N 时替代 topk。

考点小结： 掩码平均池化；全局有效性与查询权限；按查询选取前 k 项。

材料题三（3 小题，共 12 分）

多 Agent 的反思，什么时候值得写入记忆?

以下内容选译并改写自 Yihang Chen、Yuxiang Chen、Yuxuan Huang、Meng Fang、Weilin Luo、Jun Wang 的论文 Bilevel Coordinated Reflection: A Game-Theoretic Approach to Multi-Agent LLM Systems（《双层协调反思：面向多 Agent 大语言模型系统的博弈论方法》），采用 2026 年 9 月 2 日发布的 arXiv v1。节选范围为引言、第 3.2～3.4 节、第 4 节的相关实验与局限性讨论，省略证明并补充术语解释；不是全文逐字翻译。原文采用 CC BY 4.0 许可。论文原文

在许多多 Agent 系统中，一个协调者负责拆分任务，多个工作 Agent 分别完成子任务，再把结果交给协调者整合。团队还会对执行过程进行“反思”：把发现的问题、提出的假设和总结的经验写成文本，存入后续执行会读取的记忆。本研究中的模型权重在执行期间保持不变，因此，修改文本记忆是系统适应任务的重要途径。协调者的策略记忆侧重任务分工，工作 Agent 的执行记忆侧重如何完成分配到的任务。

然而，写入更多反思并不自然等于获得更多有效经验。一次反思可能修正错误，也可能把未经证实的猜测变成后续 Agent 共同依赖的前提。作者因此区分了两个过程：**提出一条反思建议**，以及**决定是否把它正式写入记忆**。如果每条建议都被接收，纠正信息和错误信息就可能一起积累。论文并没有断言所有未经筛选的反思都会失败；其理论结论依赖持续引入有害信息等具体条件。

筛选建议的一种方法，是让另一个语言模型阅读执行记录和反思文本，判断建议是否合理，本文称之为“纯文本门控”。另一种方法，则让候选记忆参与实际执行，通过环境返回、模拟器状态、测试结果或形式化检查来评价它，本文称之为“基于外部验证的门控”。两者的区别在于能否获得依赖外部环境的证据，而不在于评审 Agent 数量多少、解释长短或语气是否自信。

为什么只读文本会有局限？作者考虑了这样一类情形：两个环境对系统呈现完全相同的生成文本，同一条反思却在一个环境中有益、在另一个环境中有害。只观察这些文本的评审者没有区分两个环境的信息，因此即使文本推理能力很强，也无法保证在两个环境中都作出有益的更新。这一结论不意味着文本评审毫无价值。如果文本本身包含可完整核验的证明，它仍可支持判断；困难在于结论依赖文本没有揭示的外部状态。

论文提出了 SRMA（随机反思记忆提升）方法。系统先生成候选记忆，再用同一套评估规则比较当前记忆与候选记忆。记忆的“验证风险”记为 R，它是验证器对使用该记忆执行任务所得结果的评分，数值越小越好。在固定、确定性的评估条件下，只有候选记忆的验证风险**严格小于**当前记忆，才接收更新；否则保留当前记忆。被拒绝的候选不会因为已经生成，就自动进入下一轮记忆。

如果评估存在随机波动，仅比较两次评分的高低可能误把偶然波动当作进步。因此，论文还提出带置信控制的门控。设当前记忆和候选记忆的风险估计分别为 $\widehat{R}_{\mathsf{cur}}$、$\widehat{R}_{\mathsf{new}}$，两者的误差半径均为 a。接收条件为：

$$\widehat{R}_{\mathsf{new}} + a < \widehat{R}_{\mathsf{cur}} - a$$

也就是候选风险的区间上界必须严格低于当前风险的区间下界。区间相交或端点相等，都不满足这一接收条件。这里的 a 用来控制评估不确定性，不是额外的任务奖励；满足条件所提供的可靠性仍取决于评估样本及误差控制的假设。

作者在资源分配、双 Agent 协作游戏 Overcooked 和软件修复任务中开展实验。为了解门控到底接收了什么，论文还用**独立于门控评分的任务风险评估**判断候选建议有益还是有害：若接收建议会使这一任务风险上升，就将其标为有害。下表摘录原文表 3 的三种反思方式，数值为 5 个随机种子实验的均值 ± 标准差。

反思方式	有害候选接收率 ↓	有益候选接收率 ↑	最终任务风险 ↓
全部接收（Free-form）	100.0% ± 0.0%	100.0% ± 0.0%	0.42 ± 0.12
纯文本门控（Self-gate）	34.5% ± 4.2%	72.8% ± 5.1%	0.28 ± 0.08
外部验证门控（Grounded SRMA）	6.2% ± 1.8%	85.4% ± 3.6%	0.14 ± 0.03

“有害候选接收率”的分母是全部被判为有害的候选建议，分子是其中被门控接收的数量；“有益候选接收率”同理。它们都不是“已接收建议中某类建议所占的比例”。因此，这张表既反映能否挡住有害建议，也反映能否保留有益建议。表中列出的标准差用于描述实验波动，不直接等同于置信区间。

值得注意的是，外部验证门控仍接收了少量按独立任务评价判定为有害的建议。这与“仅接收验证风险下降的更新”并不矛盾，因为**门控验证风险与实际任务风险可能不完全一致**。例如，不完整的测试只覆盖部分行为，某条建议可能改善所测项目，却损害未测项目。论文明确指出：只有在验证器与任务目标之间满足相应校准条件时，验证风险的保证才能进一步转化为任务表现的保证。环境发生变化时，旧评估结果也可能失效，需要在新环境中重新评估当前记忆和候选记忆。

下列问题中的具体场景和逐轮风险数据为命题设定；上表为论文原始实验数据。

第 38 题

一个多 Agent 团队要决定是否把“统一采用方案 X”写入共享记忆。它可能被部署到环境甲或环境乙；两者向评审 Agent 提供的任务描述、执行记录和反思文本完全相同，但未被记录的环境状态不同，导致采用 X 在甲中降低任务风险、在乙中提高任务风险。三个只读上述文本的评审 Agent 都支持 X。

根据材料，以下哪一项最适合作为下一步处理?

A. 让协调者延长对现有文本的推理，待三个评审者都提高置信度后，再据此判断目标环境适用 X。

B. 先把 X 写入共享记忆，再让三个 Agent 根据加入 X 后的文本记录重新投票，以此验证 X 的效果。

C. 增加独立的文本评审 Agent，保持输入不变，把是否接收 X 改为由更大的多数票决定。

D. 在目标环境中用同一评估规则比较当前记忆与候选记忆，再依据验证结果决定是否接收 X。

答案： D

解析： 两个环境在评审者能看到的文本上不可区分，却对 X 有相反的效果；缺少的是区分环境的证据。实际执行并在目标环境中按同一规则比较，能够补充这种证据。增加评审数量或延长对同一文本的推理没有引入新的环境信息；先写入 X 再让评审者阅读受它影响的文本，也不能替代效果验证。这并不保证任意外部测试都充分，仍须检查验证指标是否覆盖任务目标。

第 39 题

一位工程师根据材料中的表格评价 Grounded SRMA 相对 Self-gate 的改进。为进行数值比较，本题只按表中均值计算，不据此判断统计显著性。

以下哪一项同时正确解释了数值变化和实验结论?

A. 最终任务风险相对下降 50%，接收后的建议中平均有 6.2% 有害；这两个指标分别衡量总体效果与接收集合的质量。

B. 有害候选接收率下降 28.3 个百分点，最终任务风险相对下降 50%；仍存在有害候选被接收，说明外部验证与任务目标的一致性仍需检查。

C. 有害候选接收率下降约 82.0 个百分点，最终任务风险相对下降 50%；该结果支持外部验证在这组实验中改善建议筛选。

D. 有害候选接收率下降 28.3 个百分点，但仍接收有害候选；这足以认定实验中的门控没有遵守“验证风险严格下降才接收”的规则。

答案： B

解析： 34.5%−6.2% = 28.3 个百分点；最终风险的相对降幅为 (0.28−0.14)/0.28 = 50%。约 82.0% 是有害候选接收率的相对降幅，不是下降的百分点数。6.2% 的分母是全部有害候选，不是已接收候选。独立任务风险与门控使用的验证风险可能不一致，因此仍接收有害候选不能单独证明门控违反自己的接收规则。上述结论只解释所给实验，不把均值差当作显著性检验。

第 40 题

某多 Agent 系统在同一环境中连续尝试 4 次记忆更新，采用材料中的置信门控规则，当前记忆的初始风险估计为 0.30。每轮都会重新评估当前记忆和候选记忆；为便于计算，本题假定同一记忆的各次风险估计恰好相同，且两者的误差半径均为 $\alpha = 0.02$。各轮候选均基于该轮实际保留的记忆生成，拒绝候选时保留原记忆，接收后下一轮改以新记忆为当前记忆。表中小数按精确值计算。

轮次	候选记忆的风险估计
1	0.24
2	0.20
3	0.21
4	0.18

依次执行 4 轮后，哪些轮次的候选会被接收，最终保留记忆的风险估计是多少?

A. 接收第 1、2、4 轮；最终为 0.18。

B. 接收第 1、2、3、4 轮；最终为 0.18。

C. 接收第 1、2 轮；最终为 0.20。

D. 接收第 1、4 轮；最终为 0.18。

答案： D

解析： 第 1 轮：$0.24 + 0.02 = 0.26 < 0.30 - 0.02 = 0.28$，接收，当前值变为 0.24。第 2 轮：$0.20 + 0.02 = 0.22$，与 $0.24 - 0.02 = 0.22$ 相等，不满足严格小于，拒绝。第 3 轮：$0.21 + 0.02 = 0.23$，不小于当前下界 0.22，拒绝。第 4 轮：$0.18 + 0.02 = 0.20 < 0.22$，接收，最终为 0.18。接收 1、2 轮对应把严格小于误作小于等于；接收 1、2、4 轮对应只比较均值；四轮全接收对应一直与初始 0.30 比较。

考点小结： 外部证据与多 Agent 文本共识的边界；门控实验的统计分母、改善幅度与结论边界；置信门控的严格边界与记忆状态更新。