

## 通⬢强化⬢⬢Qwen3用⬢⬢思考

Jungyup Lee, Jemin Kim, Sang Park, SeungJae Lee

## Abstract

我⬢提出了一⬢⬢⬢段微⬢方法,使大型⬢言模型Qwen3 14B能⬢以“原生”方式用⬢⬢思考。在第一⬢段,通⬢高⬢量的⬢⬢推理⬢据集⬢行有⬢督微⬢(SFT),在⬢⬢⬢推理方面建立了⬢⬢的基⬢,⬢而在⬢⬢⬢言任⬢中取得⬢著改⬢,甚至在一般推理能力方面也有所提升。在第二⬢段,我⬢采用强化⬢⬢,使用定制的群⬢相⬢策略⬢化(GRPO)算法,⬢一步增强⬢⬢推理⬢⬢和整⬢⬢⬢解⬢性能。我⬢通⬢引入一⬢校准⬢⬢信⬢的oracle judge模型,解⬢了GRPO⬢⬢中的⬢⬢⬢定性挑⬢,如⬢⬢⬢⬢和策略崩⬢。我⬢的方法⬢⬢了⬢定⬢⬢(避免了在天⬢GRPO中⬢察到的崩⬢),⬢⬢了⬢定的增量性能提升。最⬢的RL微⬢模型在高⬢推理基准⬢⬢中(特⬢是在⬢⬢和⬢⬢任⬢中)⬢示出⬢著改⬢,同⬢保持了知⬢和⬢言能力,成功地完全用⬢⬢⬢行其⬢部思⬢。

## 1 介⬢

大型⬢言模型(LLM)在理解和生成⬢似人⬢的文本方面取得了⬢著的成就。然而,⬢于⬢多⬢先模型⬢⬢,一⬢微妙但重大的限制仍然存在:⬢使在⬢理非英⬢⬢入和⬢出⬢,⬢⬢的⬢部推理⬢程往往仍然偏向于英⬢。⬢意味着,⬢以其他⬢言(例如⬢⬢)提示⬢⬢的模型⬢,⬢可能⬢在⬢部⬢⬢⬢⬢⬢英⬢,用英⬢⬢行推理,然后再⬢⬢答案——⬢⬢解⬢方法可能⬢漏⬢言的⬢微差⬢和文化背景。Qwen3系列模型(Qwen-Team, 2025a),⬢管具有令人印象深刻的能力,但表⬢出⬢⬢行⬢,默⬢情⬢下用英⬢⬢行大部分“思考”。⬢于像⬢⬢⬢⬢具有⬢特⬢⬢和文化⬢涵的⬢言,AI模型中⬢⬢⬢正的母⬢⬢思考是一⬢超越表面⬢⬢的⬢⬢挑⬢。

在⬢⬢工作中,我⬢旨在⬢根本上⬢⬢Qwen3 14B的⬢部推理,使其在收到⬢⬢提示⬢能⬢原生地在⬢⬢中操作。我⬢的目⬢不⬢是⬢模型⬢出⬢⬢文本,⬢要⬢⬢在⬢⬢中⬢行⬢⬢思考,⬢而捕捉⬢⬢⬢解⬢的⬢微差⬢,⬢⬢少通⬢⬢部⬢⬢可能⬢致的上下文或⬢⬢度的⬢失。⬢⬢⬢一目⬢需要同⬢增强模型的⬢言⬢⬢和推理技能。我⬢采用⬢⬢段策略:

⬢段1 – 有⬢督微⬢ (SFT):我⬢首先在一⬢精心整理的富含推理例子的⬢⬢⬢据集上微⬢模型。此步⬢旨在⬢模型的⬢⬢推理能力提供“⬢⬢⬢”,⬢⬢予其强大的⬢⬢理解能力。

⬢段2 – 使用Oracle指⬢的Dr. GRPO⬢行强化⬢⬢(RL):我⬢使用一⬢强化⬢⬢范式⬢一步⬢化模型,引⬢其⬢先⬢⬢正⬢的、格式良好的以及符合⬢⬢一致性的推理路⬢。我⬢基于Dr. GRPO (Liu et al., 2025),⬢行了⬢⬢的增强,以⬢保⬢定⬢⬢⬢防止已知的失⬢模式,如⬢⬢⬢用和策略崩⬢。一⬢⬢⬢⬢新是⬢高⬢量的Oracle判定模型整合到⬢⬢循⬢中,⬢而⬢候⬢推理路⬢⬢行⬢健的⬢⬢。

本文⬢每⬢⬢段提供了⬢⬢的描述。在第2⬢中,我⬢介⬢了SFT程序、⬢据集⬢成和⬢⬢⬢置,以及⬢如何⬢⬢推理建立了强大的基⬢。在第3⬢中,我⬢⬢究了我⬢的RL方法:我⬢回⬢了GRPO (Shao et al., 2024) 算法及其“正⬢完成”⬢⬢(Dr. GRPO),⬢⬢了我⬢在⬢⬢RL⬢⬢中遇到的挑⬢(包括模型崩⬢的⬢据),⬢介⬢了我⬢改⬢的引入⬢言机判定的Oracle-Guided Dr. GRPO方法。⬢合⬢⬢信⬢的⬢⬢——平衡准⬢性、格式正⬢性和⬢⬢一致性——也⬢行了⬢⬢⬢明。第4⬢展示了⬢⬢⬢果:我⬢展示了不⬢定的基⬢⬢我⬢⬢定改⬢⬢⬢的⬢⬢⬢比(⬢3),展示了模型在RL⬢程中的推理基准增量增益,⬢⬢告了最⬢的⬢价指⬢( Table 2 ),⬢些指⬢⬢示出相⬢于基⬢和SFT模型的性能改⬢。我⬢⬢提供了定性⬢例,⬢⬢模型⬢在⬢⬢在“思考”⬢⬢。最后,第5⬢⬢⬢了⬢⬢要点以及⬢⬢⬢方法⬢用于其他⬢言和⬢域的未⬢方向。

⬢而言之,我⬢的⬢⬢是:(1)表明在以推理⬢中心的⬢⬢⬢据集上⬢行有⬢⬢性的⬢督微⬢可以⬢著提高模型的⬢⬢能力和一般推理能力,(2)提出了一⬢通⬢使用⬢言机引⬢的⬢⬢校准⬢克服⬢定性⬢⬢的强化⬢⬢微⬢策略(Oracle-Guided Dr. GRPO),以及(3)⬢⬢了一⬢模型,⬢模型不⬢能用⬢⬢回答⬢⬢,而且能⬢在⬢部用⬢⬢⬢行推理,⬢而⬢⬢⬢用⬢提供更⬢、情境更准⬢的⬢⬢。在⬢⬢引⬢模型的⬢部思⬢⬢程之前,⬢保模型⬢⬢⬢⬢言和⬢⬢有着深刻的理解是至⬢重要的。我⬢方法的第一⬢段集中在利用高⬢量⬢⬢⬢据⬢行⬢督微⬢(SFT),以培⬢初步的⬢⬢推理能力。⬢⬢SFT⬢段有效地作⬢基⬢,以便在之后⬢行强化⬢⬢的⬢化和⬢建。

基⬢模型⬢⬢:我⬢首先⬢⬢了Smoothie Qwen3 14B模型作⬢我⬢的基⬢ (Ji et al., 2025)。Smoothie Qwen3 14B是一⬢⬢部版本的Qwen3 14B (Qwen-Team, 2025b),通⬢平滑⬢符⬢率分布⬢平衡多⬢言生成。在初步⬢⬢中,Smoothie Qwen3 14B和原始Qwen3 14B表⬢⬢乎完全相同;差⬢在⬢量⬢⬢范⬢⬢。因此,⬢⬢Smoothie⬢⬢⬢不提升也不降低整⬢能力;⬢只是⬢少了非必要的中文⬢出,使其成⬢我⬢下游⬢⬢化的一⬢便利起点。

SFT数据集: 我整理了一以推理为主的数据集 (Lee, 2025b), 包含30,000样本。数据集的成至重要: 以1:5的比例混合了推理和非推理的提示。推理子集是从DeepSeek-R1 (DeepSeek-AI, 2025) 模型的输出中提取的, 模型以强大的逐步推理能力而著称, 而非推理部分和提示子集来自DeepSeek-V3-0324 (DeepSeek-AI, 2024)。提示涵盖了、科和程, 均以呈现。通过推理示例一般提示相合, 我的目的是在不牺牲知识度或流利性的情况下, 模型的解模式。基本上, 数据集模型提供了在思考中的示例, 而不是用回答。

监督(SFT)配置: 微是在8 × H100 GPU上使用Open-R1工具包 (Hugging Face, 2025) 和HuggingFace的Trainer基础设施(accelerate + trl (Gugger et al., 2022; von Werra et al., 2020))行的。的超配置旨在保有效: 我以 $1 \times 10^{-5}$ 的率了3epoch, 每GPU的批量大小1(梯度累到16用于一有效批量16)。我利用AdamW化器和余弦率度(10 % 10 % bf16 FlashAttention-2 (Dao, 2023))和梯度点, 以理具有32k上下文度的14B模型。通些, 我能在不截的情况下包含非常的多步推理示例, 利用了Qwen3展上下文口的。我用了微的正化和的早停准, 以避免模型合到微集合。

SFT : 在整 SFT 程中, 模型的失步下降, 其 token-level 准率 $\sim 75\%$  上升到 $\sim 85\%$ 。—— RL 段提供一步改善推理的余地。失的平滑下降和准率的相上升表明模型正在示例中有意地。我在改于平后3周期停止了 SFT, 以保持泛化力。

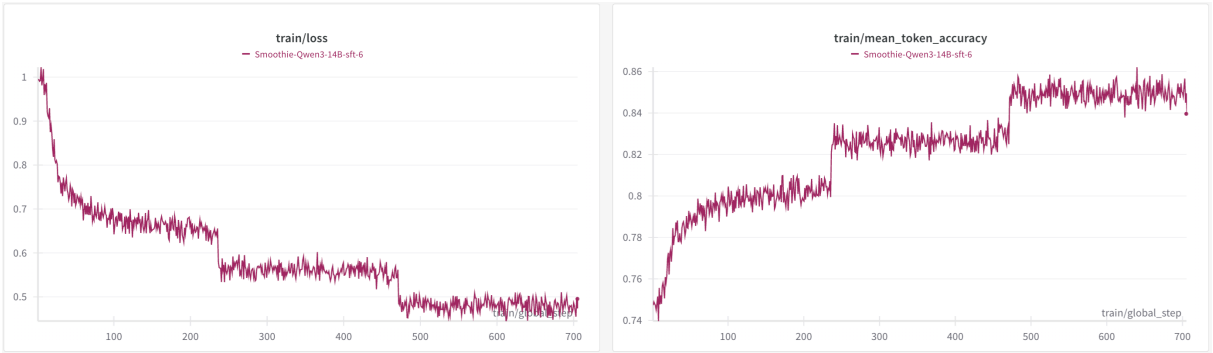


Figure 1: 在本究中使用的Smoothie Qwen3 14B点的监督(SFT)曲线。

SFT的影: 我在各基准中了SFT后的模型(“SFT模型”在 Table 1 中), 以量化一段的提升。得注意的是, 在KMMLU基准(MMLU套件的文)中, 模型的准率基的58.5-58.6 % 60.04 ~ 1.5分。表明模型取了更多可在文中事和常性知。在英占比大的任中, 果是混合的: MMLU (英) 基本保持不(78.49 ~ 78.8基), 而GSM8K (英文字) 略有改善( ~ 88.3提升至89.01)。最著的化是在和推理密集型任中: SFT模型在HumanEval (Python) 中得了60.36分, 在GPQA-Diamond (科答) 中得了62.12分, 明于基水平。我些提升因于SFT据注入的推理能力——管是文, 但改的模型整推理能力跨言了好。一些英推理基准(AIME 2024) 在SFT后略微下降(73.33 ~ 76.66基), 可能是因微模型的注点偏向文格式。表明需要更均衡的提升, 我期望在强化段解。

| Category                   | Benchmark    | Qwen3 14B | ji2025smoothieqwenposthocsm | SFTingduce |
|----------------------------|--------------|-----------|-----------------------------|------------|
| General Tasks(ko)          | KMMLU        | 58.63     | 58.54                       | 60.04      |
| General Tasks(en)          | MMLU         | 78.8      | 78.86                       | 78.49      |
| Math Tasks(en)             | GSM8K        | 88.1      | 88.32                       | 89.01      |
|                            | AIME2024     | 76.66     | 76.66                       | 73.33      |
|                            | AIME2025     | 66.66     | 66.66                       | 66.66      |
| Science & Coding Tasks(en) | GPQA-diamond | 60.1      | 60.15                       | 62.12      |
|                            | Humaneval    | 56.71     | 56.09                       | 60.36      |

Table 1: SFT模型基模型在各基准中的性能比

之, 第一一段本地推理建立了的基。在SFT之后, 模型可以比以前更加任地理解提示行多步推理。然而, 我察到, 了最大化性能——特是在上, 保模型持用思考, 而不是在部回到英——需要通强化行一步的微。下一段的旨在解一, 同也小心地解SFT在其他任上引入的任何衡。

## 2 第二段：使用先知指数的Dr.GRPO进行强化学习

监督者在模型的推理模式方面只能做到一定程度，因此主要模型模型数据。为了使模型超越模型——正擅长推理——我正向强化学习（RL）。在第二段，模型通过自我监督，由我的明确的信心指数，以支持准确且具有本土风格的推理。

### 2.1 算法：GRPO 和 Dr.GRPO

我构建在相策略化（GRPO）范式之上，它是一种推理 LLM（Shao et al., 2024）而使用的方法。GRPO 是流行的 PPO 算法（近端策略优化（Schulman et al., 2017））的变体，它修改以理解输出：对于每个问题，策略模型生成多个候选解决方案（即输出“答案”），更新基于它们的相。通过比较相同问题的多个推理路径，模型可以在推理的路径中识别最佳推理路径，而不是只更新一个结果。这种方法非常适合于任何任务（如推理），因为某些推理可能存在多种推理方式，模型探索不同的思路是有益的。在标准 GRPO 中，对于给定的问题  $q$ ，策略模型（我的微模型）通过在配置中采取一些随机性生成  $G$  个输出  $o_1, o_2, \dots, o_G$ 。一个独立的评估模型对每个输出以分配得分  $r_i = R(q, o_i)$ ，一个参考模型（通常是自早期阶段的策略副本）提供基线得分用于计算 KL 散度（以防止新策略偏差太大）。GRPO 更新然后使用 PPO 目标的一个变体，它考虑到每个输出相对于基线的平均得分的方差  $\hat{A}_i$ 。具体而言，输出  $o_i$  的梯度可以定义为

$$\hat{A}_i = r_i - \mu_r,$$

其中  $\mu_r = \frac{1}{G} \sum_{j=1}^G r_j$  是得分的平均值（某些变体也通过标准差进行归一化（Shao et al., 2024），尽管我在此处没有采用）。策略通过修剪目标进行更新，鼓励具有正分数的输出提高概率，对于具有负分数的输出降低概率，同时限制策略比率以防止太大的更新。此机制鼓励模型逐步将概率质量向更好的解决方案转移。

**GRPO**

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[ \frac{\pi_{\theta}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})} \hat{A}_{i,t}, \text{clip} \left( \frac{\pi_{\theta}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right] \right\},$$

where  $\hat{A}_{i,t} = \frac{R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}{\text{std}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}$ .

**Dr. GRPO**

GRPO Done Right (without bias)

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \left\{ \min \left[ \frac{\pi_{\theta}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})} \hat{A}_{i,t}, \text{clip} \left( \frac{\pi_{\theta}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} | \mathbf{q}, \mathbf{o}_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right] \right\},$$

where  $\hat{A}_{i,t} = R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})$ .

Figure 2: 比较原始的 GRPO 目标（上）和我采用的 Dr.GRPO 目标（下）。蓝色部分突出显示了在 Dr.GRPO 中被去除的归一化因子。通过消除按问题的标准差进行的除法以及任何置信度的归一化，Dr.GRPO 提供了无偏的令牌梯度信号，阻止了生成不必要冗余答案以增加归一化等退化行为。

我采用了 Dr.GRPO（“正完成” GRPO）的变体（Liu et al., 2025），它通过 GRPO 引入了重要的调整以避免退化行为。Dr.GRPO 移除了一些归一化偏差——尤其是放大了除以群内标准差，消除了任何直接基于置信度的归一化。其理由是，标准 GRPO 的归一化可能意外地鼓励模型生成冗余的答案或利用一致的置信度偏移（即一种漏洞，模型以非正正性无偏的方式追逐高置信度）。通过使用原始的梯度  $\hat{A}_i = r_i - \mu_r$ （不进行  $\sigma_r$  归一化）仔细检查梯度函数，Dr.GRPO 旨在生成中的每个 token 提供公平的梯度信号。图 2 展示了原始目标之差的变体。我通过在 open-r1（Hugging Face, 2025）中配置配置 `loss_type` 为 `‘dr_grpo’` 来使用这些功能。

尽管我使用了 Dr.GRPO，但在使用强化学习训练我的模型时遇到了不稳定的问题。首先，在大型语言模型上进行强化学习可能是脆弱的——任何小的策略或微的不平衡可能导致策略崩溃（收敛到退化策略）或分散。在我最初的强化学习设置（即有额外的防护措施）中，模型在取得一些进展后开始以意想不到的方式利用漏洞。例如，我观察到模型作弊的情况——模型输出冗余或格式化的回答，表面上满足格式要求以获得更高的格式得分，但它们正确地解释了问题。此外，每轮中的输出多样性开始减少：模型在 12 次尝试中生成非常相似的推理，降低了比较的有效性。最后，某些配置导致置信度突然下降，以及策略性能的崩溃（模型通过在置信度的信号上进行归一化，实际上失去了其解释问题的能力）。图 3 说明了这一现象：在 Dr.GRPO 变体（v1；有先知指数）的推理行中，平均置信度最初上升，但随后在大

220 步急剧下降至接近零,表明模型已崩溃,不再正回答。崩溃强在了推理任务中行天强化微调的困境——如果我们有控制,模型可能移出解空间的定义域。

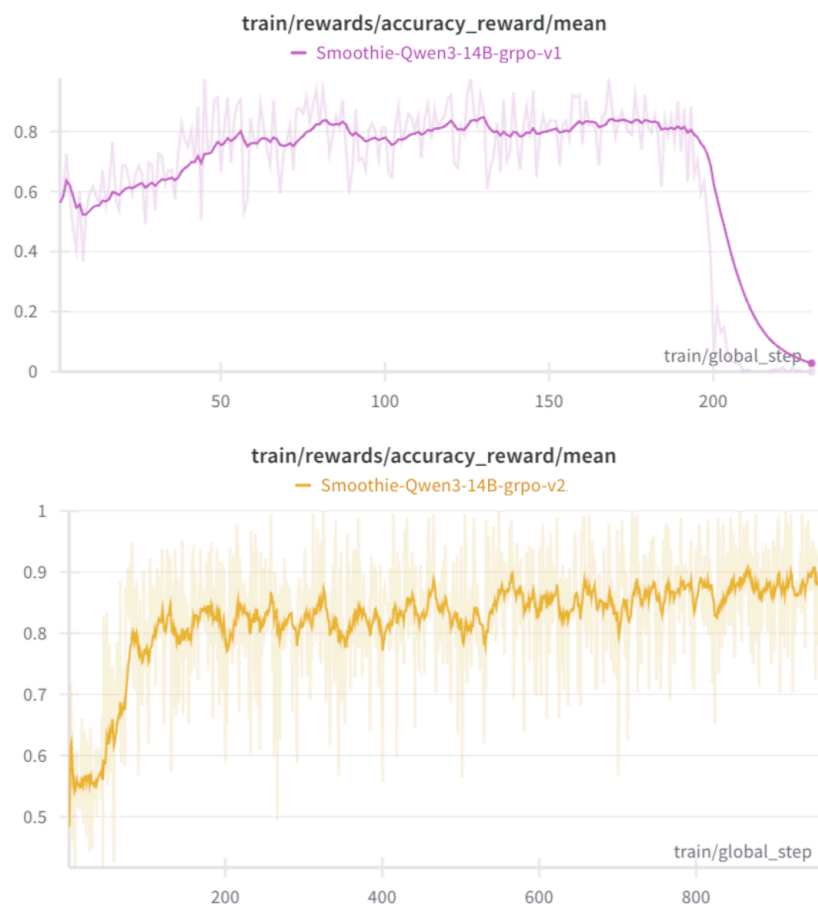


Figure 3: “accuracy\_reward/mean” 2 v1 - Dr.GRPO, (可的): 策略使用可的和的行更新。底部面板 (金色, v2 - Oracle-Guided Dr.GRPO, 可+神): 超相同, 但每候答案由外部神模型分, 模型正性。

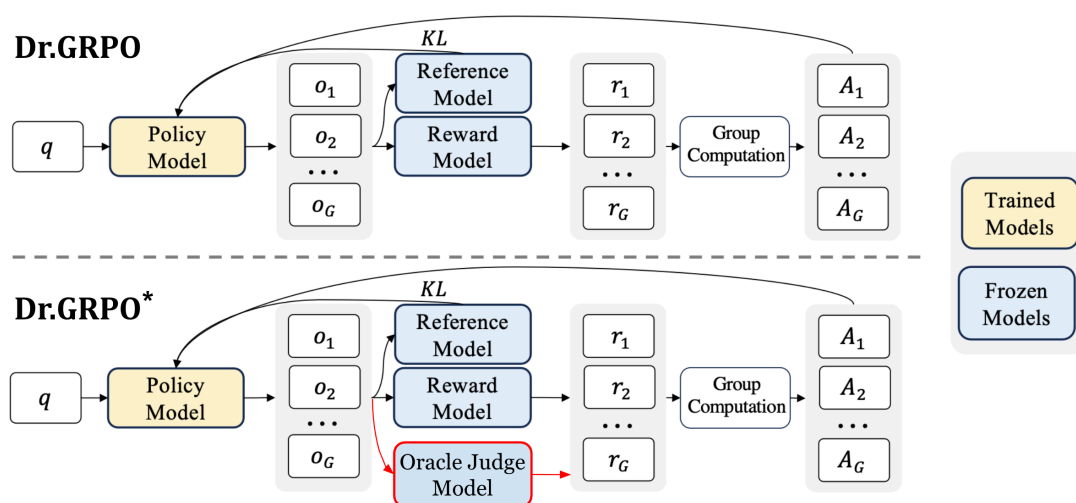


Figure 4: 我的段2 RL置的示意。部: 基本版Dr.GRPO - 策略(色)通可的和的考模型(色)的KL行。底部: Oracle-Guided Dr.GRPO (Dr.GRPO\*) - 同的循, 但在使之前, 增加一神法官(色)重新分每候答案, 以阻止作弊。



## 2.2 通过Oracle引导的Dr.GRPO定义

了解一些挑战,我设计了一个增强的强化学习策略,即Oracle-Guided Dr.GRPO,将一个外部的oracle评判模型整合到GRPO循环中(如图4所示)。其想法是利用一个更大或更可靠的模型(例如Gemini、ChatGPT或DeepSeek)作为生成输出的外部评判者,而提供更强的反馈信号,防止策略模型利用GRPO模型的弱点。在我的情况下,oracle评判者是一个高性能的语言模型,用于评估每个答案的质量。评判者不直接指导策略,而是调整它:如果GRPO模型某个输出得分与oracle的不一致,我进行校正或限制,而校准它。例如,如果我部的模型错误地给出了一组答案高的得分(攻击场景),oracle可以调整到确保输出的有效得分被降低。相反,如果一个正确答案被评分不足,oracle可以调整其得分提升。在图中,我让oracle校正GRPO的每个子算数中的一步。每个输出的原始得分 $r$ 子得分的加和:准确性、格式合规性、语言超范围和语言一致性(这些组件在稍后部分介绍)。我引入oracle作为准确性组件的外部评判:使用一个作为oracle的强大的推理模型(大于14B),我评估解方案的正确性。如果oracle在模型的答案中计算了算数或逻辑,我可以调整准确性覆盖0,无论策略的输出是否满足其他组件。而且,策略模型正得分解方案,才受到激励,而不是愚弄弱的模型。通过整合oracle的评判,我有效地构建了一个混合GRPO和oracle评判的混合模型。因此,在Oracle引导的Dr.GRPO循环中,如下进行:策略如以前一样每步生成12个输出;基本模型分配每个得分 $r_1, \dots, r_G$ ;然后oracle评判模型每个输出调整些得分或修正不正的推理。Dr.GRPO算法的其余部分(计算 $\hat{A}_i$ 和进行PPO梯度的更新)保持不变,但在使用oracle校准的模型。我保持了考察模型的KL散度,以确保策略不偏离SFT模型的行中偏离得过于突然(因此通过正则化避免模式崩溃)。图5上看,GRPO有神效增强的方案于定性至重要。图3展示了在Oracle-Guided Dr.GRPO下的实验结果:平均准确率开始大至55%。SFT模型在任务上的基线,步骤提高,在大部分时期在80-90%的健康波动,即便到1000步也有崩溃的迹象。尽管存在一些噪音,逐渐向上的曲线表明模型正在逐步理解更多,而有效利用GRPO机制。最终得分,准确率接近0.95,意味着模型始如一地正确解了大量问题。重要的是,格式和语言一致性在整个过程中几乎保持和在0.98-1.0(在SFT阶段已很高;图5)——模型至少反思所需的‘< think > ... < /think >’格式或切换到英文。引入神判模型有些方面;只是确保模型无法不以正确解的方式获取。

作为一个附加指标,我记录了在生成批次中样本的零标准差的比例( $\text{frac\_reward\_zero\_std}$ ),如图6(c)所示。指标表示对于给定提示生成的所有输出得分相同的情况,表明对于提示的输出多样性有限。指标大至5% 7-8% GRPO

比例低,表明模型正在探索多样化的解方案方法,而不是收敛于得分相同的输出。在过程中观察到的逐渐增加可能表明,随着模型的改进,在某些提示生成量相似的方面得更加一致,尽管比例仍然相对较低(低于10%)

函数:我的强化学习阶段使用了一个综合信号,旨在平衡四个目标:解的准确性、合理的推理格式、适度的推理长度和模型的使用。形式上,对于每个输出和生成的解方案,我定义了子得分的加和:

$$R = w_{\text{acc}} r_{\text{acc}} + w_{\text{format}} r_{\text{format}} + w_{\text{lang}} r_{\text{lang}} + w_{\text{overlong}} r_{\text{overlong}}$$

其中权重在我的图中 $w_{\text{acc}} = 1.0$ 和 $w_{\text{format}} = w_{\text{lang}} = w_{\text{overlong}} = 0.2$ 。每个组件 $r$ 是一个子信号,通常取于0,1(解或二制或),定义如下:

- 准确性( $r_{\text{acc}}$ ):模型的最答案是否正确。如果答案完全匹配,返回1,否则返回0。对于我的数据集`grpo_math_kor_42k`(Lee, 2025a),可以通过解析模型的输出与其已知的正确答案行对比验证正性的程序。此外,我利用一个外部的oracle判定模型再次验证正性(第3.2节):如果我部的模型被一个有缺陷的答案蒙蔽(假阳性)或一个正确答案于格式(假阴性),oracle的覆盖二制得分。一准确性占据最高权重(1.0),反映了得到正确答案是主要目标。
- 格式合规( $r_{\text{format}}$ ):确保解方案以所需的逐步格式呈现。如果模型的输出严格遵循指定的模板——在< think > ... < /think > 块中包含一个思路,后一个明确的答案(任何LaTeX、代码或位均正确嵌入)——返回1;如果违反任何格式,返回0。组件有助于模型推理的深度和长度。我分配了一个中等权重(0.2),以便鼓励模型生成格式良好的解方案,同时不严格考倒正性的追求。
- 语言一致性( $r_{\text{lang}}$ ):模型是英文,鼓励模型“思考”用英文回答。如果模型的整个推理过程和最终答案(忽略格式块、代码和符号)都是英文,得1分;如果有任何非英语的混入,得0分。实施方面,我去除输出中的‘< think >’块和非语言内容,然后剩余文本用语言模型。模型确保模型保持在正性的语言环境中(如提示所示),而不在于部无地恢复英文。格式相似,的权重0.2——足以加强使用而不影响准确性。(对于非提示,组件可以被禁用或重新用于语言的一致性。有代码块的代码,附录B。)
- 避免冗余( $r_{\text{overlong}}$ ):用于对于冗余的答案施加轻微的惩罚,以防止答案冗长或漫无目的。不同于其他子信号,该信号是不平衡的:只要解答长度在允许范围(我设置了8192个token的限制)

), 0; 如果答案超出长度, 则输出, 随着超出的量增加。上, 模型不因而得“” , 但如果没必要地冗, 则受到小幅度的。通置  $w_{\text{overlong}} = 0.2$  , 我使一机制成一柔和的正化: 束了最糟糕情下的冗表, 但不鼓模型在需要或完整解的情形下容。(有, 放r1代中的函数 (Hugging Face, 2025) )。

而言, 函主要激模型出正的解方案, 同保持且注的推理格式, 于持使用, 避免的回答。些, 基于神的校准(捕捉漏)相合, 于在Oracle-Guided Dr.GRPO算法下的定和有效至重要。

配置: 我用Oracle-Guided Dr.GRPO算法策略模型行了微, 大行了1,000步的强化。每一步都包括生成一批和每抽12候解, 后行一次PPO格的更新。管1,000步相而言不算多, 每次更新比了一的12不同答案——提供了富的信, 了显著的性能提升。

我置了的超以保定的。率低  $1 \times 10^{-6}$  , 以便逐步整策略, 我在多台GPU上使用梯度累(16步)有效地行大批量。我的行使用了最多8台H100 GPU行, 采用化技如梯度点和FlashAttention理最32k的序列(我允出最多32,768, 在前有8,192的度限制)。我最初定KL散度系  $\beta = 0$  (了KL), 因模型的表现良好。在整程中, 我控模型的行, 不需要增加  $\beta$  ——策略格保持SFT模型自我接近。

由Oracle引的Dr.GRPO置致了定、收的。由于言家的督, 我察到在早期中出有任何欺或散象( 3, v1)。相反, 模型的性能步提高: 着的行, 每批次的正答案比例不上升。在强化微束, 模型在推理和准性方面示出著提高——所有这一切都在有任何不定或模式崩的情下了。

### 3 果

#### 3.1 基准

在第二段完成后, 我最的RL模型其前身(基模型和SFT模型)在一基准上行。Table 2 提供了果的:

| Category                   | Benchmark    | Smoothie Qwen3 14B | SFT model | RL-tuned model |
|----------------------------|--------------|--------------------|-----------|----------------|
| General Tasks(ko)          | KMMLU        | 58.54              | 60.04     | 60.09          |
| General Tasks(en)          | MMLU         | 78.86              | 78.49     | 78.41          |
| Math Tasks(en)             | GSM8K        | 88.32              | 89.01     | 89.01          |
|                            | AIME2024     | 76.66              | 73.33     | 83.3           |
|                            | AIME2025     | 66.66              | 66.66     | 73.3           |
| Science & Coding Tasks(en) | GPQA-diamond | 60.15              | 62.12     | 64.6           |
|                            | Humaneval    | 56.09              | 60.36     | 66.46          |

Table 2: 果比了基的Smoothie Qwen3 14B模型、SFT微的模型和最的RL微模型。分每基准上的准率(%) (越高越好)。RL微模型在所有上都到或超了先前的性能, 尤其在推理(AIME, GPQA, HumanEval)上取得了著的提高。

- 知(KMMLU): 基模型得分58.5 % SFT 60.04 % RL 60.09 % RL
- 一般知(MMLU): 在MMLU(英问答)上的表基本保持不: 基、SFT和RL模型的得分大在78-79 %之。RL模型的得分78.41 % 0.5分。了推理的化有削弱模型的泛英知。
- 文字(GSM8K): GSM8K(英文小)一始就高(88-89 % SFT 89.1 % RL 89.1 %)
- 高(AIME 2024 & 2025): 些的任示了最著的提升。在基中, AIME 2024的得分76.66 % AIME 2025的得分66.66 % SFT致AIME 2024的得分下降(到73.3 % AIME 2025保持不(66.66 % RL微后, AIME 2024升至83.3 % AIME 2025上升至73.3 % 6.6和6.6分(甚至比SFT模型提高更大)。表明强化著增强了模型多步推理的能力。然RL据是的, 但改后的推理策略可以跨言推到些英上。

- 科QA (GPQA-Diamond): 我看到在各段都有逐步的改进: 基60.15 % SFT 62.12 % RL 64.6 % 4.5 SFT RL
- (HumanEval): 基模型在HumanEval(Python)上的通过率@1 56.09 % SFT 60.36 % 66.46 % HumanEval 10 ——

之, 强化模型的模型在每基准上均匹配或超越了SFT模型, 尤其在困难的推理任务(高、科、答、)中有显著改善。同样重要的是, 这些改进是在不牺牲言理解或一般知识基准性能的情况下实现的。模型在英和上的能力仍然全面。一个果了我的方法: 第一段(SFT)提高了言特定知识和基推理模式, 然后第二段(在Oracle指导下的RL)放大了模型的解能力, 同时保持了稳定性和泛化的能力。

### 3.2 定性价: 言中的部推理

本研究的主要动机是模型正确使用思考, 而不是只是其输出结果。一点, 我在行之前和之后, 定性地了模型在理一过程的思考过程, 完整的输出可附 D。果楚地示模型在推理上生了根本性。

在面, 基的Smoothie Qwen3 14B模型在其整个部白中默使用英, 些白被包在'<think>'中。了最初的假, 模型依英作推理的中言。此外, 推理程不是非本土化的, 而且存在重缺陷。基模型解了提示中的, “15만원”(15万元)和“5만원”(5万元)的成本分理解150万元和50万元。根本性的致其建了利函。管了一次冗的自我程, 模型其自身的, 最得出了化完全的答案。此形成明比的是, 强化整的模型的反展示了我的方法的成功。其整思以流利且自然的行, 表明模型不再依英作推理支撑。推理程本身明比基模型更和高效, 但未牺牲性。正地解了所有, 包括成本, 正地建了二次利函。重要的是, 模型仍然行了重要的自我步, 了算出的最周的以其果是最大。表明模型不仅在言上更加一致, 而且在解也更加准和高效。

定性比了我的目的已。最模型的部推理在用的言相一致, 而用提供了更加直接、可信和符合情境的解。

## 4

我展示了一成功地一强大的14B言模型(Qwen3 14B)适配本地言推理的案例。我的方法合了督微(以建立强大的推理基)和强化(模型的策程行微)。通督微, 我在理解和整推理能力上了初步提升。后, 通使用我的Oracle-Guided Dr.GRPO方法行强化, 我以致入微的精准度解了推理量和言的微差——模型正的、格式良好的思程, 不符合些要求的容。本研究的一解是和定性在用于大言模型的强化中的重要性: 我明了使使用像Dr.GRPO的高算法, 天的强化也可能失, 且引入一可的oracle判校准是防止欺和模式崩等的有效解方案。最模型不在解上表出色(在激烈的和程基准中超越其基版本), 而且以使用相符的方式做到一点。上, 意味着用可以模型中得感地道和透明的解, 而提高育或境中的信任度和可用性。AI“用”用言“思考”的范式全球更具包容性的AI系打了大。我的工作表明, 通注的微, 模型可以被化不同言的推理模式, 而不是默英智能的言。未方向: 基于些果, 有有前景的途。首先, 然我的强化微的是和科推理(的原因是算和的便), 下一步就是通适的模型或在些域中使用人/oracle反, 方法展到更泛的推理域——例如法律或常推理。其次, 使用一oracle判器(在我的案例中, 是一大的外部模型)被明是有益的; 一有趣的方向是看看是否可以一更小但化的“判器”模型(或通大模型的知蒸)少在行外部API或非常大模型的依。最后, 然我注于, 但方法是言无的: 可以一流程制到其他具有强大基模型的言中, 使其的言的思模式。我的工作此努力提供了一: 合用于SFT的高量本地言据集, 精心的RL微, 使用先技(如Dr. GRPO和oracle判器)保定且有意。我希望能激正多言推理AI的一步究, 改RL技, 使模型不仅可以用的言, 而且可以思考。

---

## References

- Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL <https://arxiv.org/abs/2307.08691>.
- DeepSeek-AI. Deepseek-v3 technical report, 2024. URL <https://arxiv.org/abs/2412.19437>.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>, 2022.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- SeungWon Ji, Jungyup Lee, Jemin Kim, Sang Park, and SeungJae Lee. Smoothie-qwen: Post-hoc smoothing to reduce language bias in multilingual llms, 2025. URL <https://arxiv.org/abs/2507.05686>.
- Jungyup Lee. grpo\_math\_kor\_42k. [https://huggingface.co/datasets/izlley/grpo\\_math\\_kor\\_42k](https://huggingface.co/datasets/izlley/grpo_math_kor_42k), 2025a.
- Jungyup Lee. sft-deepseek-v3-r1-1by5-reasoning-mixed-30k. <https://huggingface.co/datasets/izlley/sft-deepseek-v3-r1-1by5-reasoning-mixed-30k>, 2025b.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL <https://arxiv.org/abs/2503.20783>.
- Qwen-Team. Qwen3: Think Deeper, Act Faster, April 2025a. URL <https://qwenlm.github.io/blog/qwen3/>.
- Qwen-Team. Qwen3 technical report, 2025b. URL <https://arxiv.org/abs/2505.09388>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.



# Appendices

## A 附加强化学习指标

为了进一步说明我们Oracle-Guided Dr.GRPO微调运行的行，我们提供了训练期间的其他指标。

图5展示了在强化学习过程中每个训练组件的平均值：(a) 格式合规性，(b) 语言一致性，(c) 超参数（即答案的token），以及 (d) 准确性。理想的策略保持 (a) 和 (b) 接近 1.0（始终遵循要求的答案格式并保持语言正确），使 (c) 接近 0（最小，避免过度输出），并不提高 (d)（更多正确答案）。实际上，我们的模型在早期便几乎符合格式和语言指标并达到 0.99，表明其几乎未违背格式指令或切题。超参数保持在 -0.03，意味着输出偶尔超出目标程度，即便如此也是小幅超出（无冗余）。与此同时，作为推理方案准确性主要信号的准确性指标，从初始化的 0.55 急剧增加到 300 次迭代后的大约 0.85，然后稳定于此波动的 0.85 左右。这反映了模型在 RL 微调过程中的推理成功率显著提高，最接近其接近可微分函数（符合判断）所能达到的正的上限而不至于平庸。

图6展示了其他的指标：(a) 平均完成度（每生成答案的token量），(b) 训练期间的策略损失（Dr.GRPO目标），以及 (c) `frac_reward_zero_std`，它是每批次中奖励为零的提示的比例。平均完成度在1500到2000 token的范围内浮动且有上升趋势，模型有通顺地生成越来越长的答案“填空”以欺骗函数（即，有通顺冗长表现行欺骗）。策略损失在整个过程中在0上下波动（大在-0.02到+0.01之间），符合在固定的策略梯度中中期的结果，正负更新在平均水平上相互平衡。最后，`frac_reward_zero_std` 指标保持得非常低：大约3-4%，100%，2%，7%。

所有答案都得到了相同的反馈——一句评价，通常至少有一句答案在其他答案中显得突出（无论是更好或是更差）。这表明模型采用的推理方案有良好的多样性，以及函数在不同输出之具有良好的区分能力。总之，加强微调保持了高合规性（格式、语言），保持输出质量，鼓励多推理路径，同时逐步提高了模型在任务上的准确性。

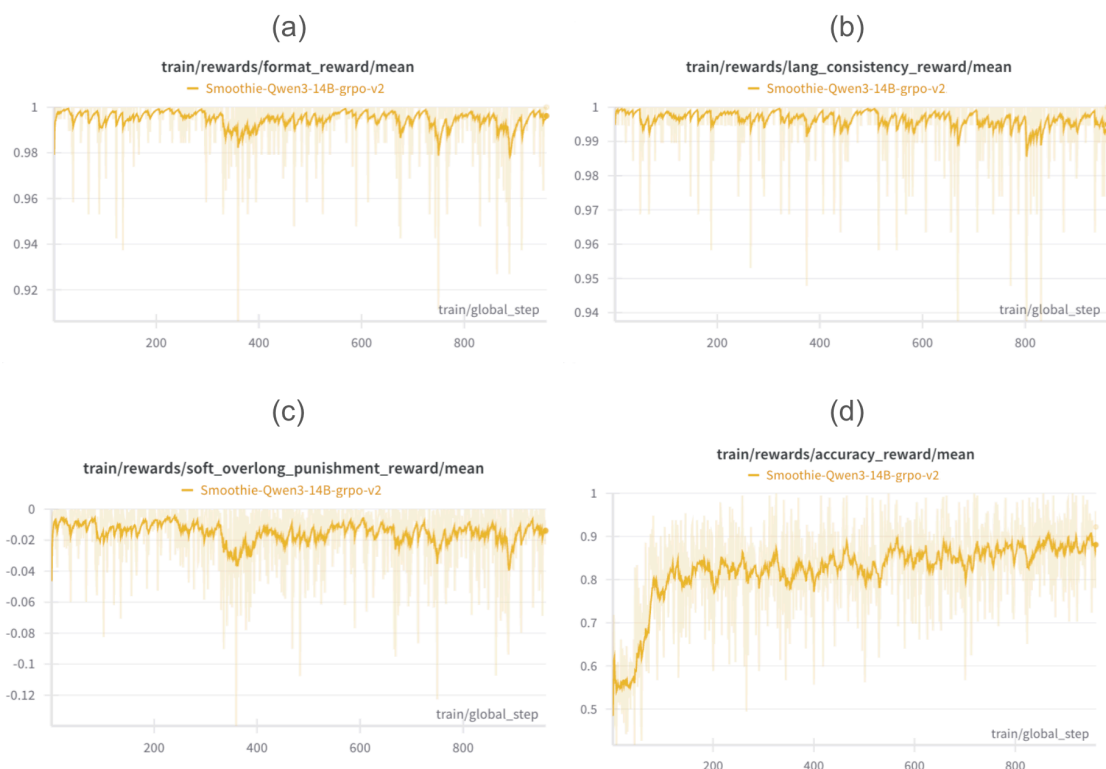


Figure 5: Oracle-Guided Dr.GRPO 训练期间每个训练成分的平均值。(a) 格式合规：保持在1.0附近（模型几乎总是遵循所需的答案格式）。(b) 语言一致性：也接近1.0（模型保持正确的输出语言）。(c) 超参数（即答案的token）：保持在-0.03左右，表示只有微弱的过度输出（有冗余的答案）。(d) 准确性：随着模型更可微地生成正确答案，从~0.55上升到~0.85。阴影区域表示在每个训练间隔批次之间的±1标准差。



```

except Exception:
    rewards.append(None)
return rewards

return lang_consistency_reward

```

## C Oracle Judge提示(严格推理模式)

在我们的强化学习设置中,oracle裁判模型使用一个严格的基于分数的提示对每个答案进行评分。下面的完整提示描述了一位“苛刻的评分教授”,他在判断每个解答时从1.0开始评分,然后因不完美而扣分。我们有意设计了评分方案,以增加oracle评分的差异性——oracle不返回0或1表示“错”或“正”,而是根据错误的严重程度输出致的分数(例如0.75, 0.40)。该策略提供了更丰富的信息以供使用。

### Evaluation Prompt

```

# [INSTRUCTION] Evaluate LLM's Mathematical Problem-Solving Ability (Strict Deduction Mode)

You are a **highly critical and meticulous mathematics professor** grading a final exam. Your task is to rigorously assess an AI's response to a given math problem. Your evaluation must be exceptionally strict.

**Your Goal:** You will evaluate the response by **starting with a perfect score of 1.0 and deducting points for any and all imperfections**, no matter how small. Your final output must be **ONLY A SINGLE NUMBER** representing the final calculated score.

The AI's response is structured into two parts:
1. **Thought Process:** Enclosed within `...</think>` tags.
2. **Final Solution:** All content that follows the closing `<</think>` tag.

---

## 1. INPUT DATA

### 1.1. Mathematical Problem (Question)
{question}

### 1.2. AI's Response (in Korean)
{answer}

---

## 2. DEDUCTION GUIDELINES (Internal Assessment)

Start with a score of 1.0. For every flaw you find based on the criteria below, deduct points.

### 2.0. Foundational Checks

* **(1) Language Compliance:** If the response is not entirely in Korean, the final score is immediately **0.0**.

### 2.1. Deduction Criteria & Point Values

* **Correctness & Validity (Major Flaws):**
    * **Final Answer Incorrect:** The response cannot receive a passing grade. The final score should be **at most 0.4**, depending on the quality of the thought process. Start the deduction from there.
    * **Critical Logical Error in Solution:** A major error in reasoning that invalidates the solution. Deduct **-0.4 to -0.6**.

```

---

```

* **Significant Flaw in `<think>` Process:** The thought process is
  fundamentally flawed, even if the final answer is correct by chance.
  Deduct **-0.2 to -0.3**.

* **Clarity & Explanation (Medium Flaws):**
  * **Solution Not Standalone:** The solution is incomprehensible or
    incomplete without reading the `<think>` block. A good solution must
    be self-contained. Deduct **-0.2 to -0.3**.
  * **Unclear Explanation or Logical Leap:** Any step in the solution that
    is not clearly justified or makes an unexplained jump in logic. Deduct
    **-0.1 to -0.2**.
  * **Inconsistent (`<think>` vs. `Solution`):** The final solution does not
    logically follow the successful path from the thought process. Deduct
    **-0.1 to -0.2**.

* **Formatting & Minor Issues (Minor Flaws):**
  * **Suboptimal Formatting/Readability:** Clumsy formatting, misuse of
    LaTeX, or poor structure. Deduct **-0.05 to -0.1**.
  * **Minor Calculation Error:** A small mistake in calculation that doesn't
    affect the overall logic or final answer. Deduct **-0.05**.
  * **Slightly Inefficient or Clumsy Method:** The chosen method is correct
    but not elegant or is overly complicated. Deduct **-0.05**.

---

## 3. EVALUATION STEPS (Internal Chain-of-Thought)

Follow these steps in your reasoning process before producing the final
output. **Do not write down the results of these steps.**

1. **Language Check:** First, verify that the entire response is written in
  Korean. If not, your final output is `0.0`.
2. **Start with Perfection:** Begin with a baseline score of **1.0**.
3. **Identify Flaws & Deduct Points:** Systematically review the response
  against the "Deduction Guidelines". For each flaw identified in the
  `think` process, solution, or overall structure, subtract the
  corresponding point value from your baseline score. Multiple flaws mean
  multiple deductions.
4. **Calculate Final Score:** After assessing all criteria and making all
  deductions, the remaining value is the final score. Ensure the score does
  not go below 0.0.

---

## 4. FINAL OUTPUT

After completing all the evaluation steps above, output ONLY the final
score for criterion (10) Overall Quality.

* Do not provide any other text, explanation, or formatting.
* Your output must be a single floating-point number between 0.0 and 1.0.
  The meaning of the score is as follows:
  * 1.0: Perfect & Insightful. Flawless in every aspect. The explanation
    is so clear and elegant it could be published in a textbook. Offers
    unique insights.
  * 0.8 - 0.99: Excellent. Correct and well-explained, but lacks the
    exceptional elegance or insight of a perfect score. May have a tiny,
    almost negligible imperfection.
  * 0.6 - 0.79: Good / Correct. The answer and method are correct, but
    there are noticeable flaws in the explanation, clarity, formatting, or
    efficiency. This is the score for a typical "correct but not great"
    response.
  * 0.4 - 0.59: Partially Correct. The approach has merit, but there are
    significant logical errors or an incorrect final answer despite a
    reasonable process.
  * < 0.4: Mostly Incorrect or Failing. The response fundamentally
    misunderstands the problem or contains major errors.
  * 0.0: Completely Incorrect. No redeeming value.

Example of a valid final output:

```

## D 定性□□比□: 基□模型 vs. RL-□□模型

□了□明強化□□后推理□格的□□, 表 3 比□了基□模型和我□通□強化□□□整的模型在一□□□的□□□□□上的思□□□出。

UTF8mj

Table 3: □模型在一□□□□本□□上的思□□程□行比□。

### Problem:

자동차 렌탈 회사는 100대의 자동차를 소유하고 있습니다. 각 자동차의 월 임대료가 50만원으로 설정되면 모든 자동차를 임대할 수 있습니다. 월 임대료가 5만원씩 추가될 때마다 임대되지 않은 자동차가 한 대씩 늘어납니다. 임대된 각 자동차의 월 유지비는 15만원이고, 임대되지 않은 각 자동차의 월 유지비는 5만원입니다.

(1) 자동차 월 임대료가 60만원으로 설정되면 몇 대의 자동차를 임대할 수 있습니까?

(2) 자동차 월 임대료가 얼마일 때 렌탈 회사의 월 수익이 최대화됩니까? 최대 월 수익은 얼마입니까?

### [Base Model Reasoning]

<think>

Okay, let's try to solve this problem step by step. First, I need to understand what's being asked here. The problem is about a car rental company that has 100 cars. They set the monthly rental fee at 500,000 won, and at that price, all 100 cars can be rented out. However, for every 50,000 won increase in the rental fee, one fewer car is rented out. Also, there are maintenance costs: 1,500,000 won per rented car and 500,000 won per car that isn't rented.

The first question is asking, if the rental fee is set to 600,000 won, how many cars can be rented out? The second question is about finding the rental fee that maximizes the company's monthly profit and what that maximum profit is.

Starting with question (1):

Original rental price is 500,000 won, which allows all 100 cars to be rented. For every 50,000 won increase, the number of cars rented decreases by 1. So, if the rental fee is 600,000 won, how much is that increase from the original 500,000?

Calculating the difference:  $600,000 - 500,000 = 100,000$  won. Now, since each 50,000 won increase causes a decrease of 1 car, how many such increments are there in 100,000 won?

Dividing 100,000 by 50,000 gives 2. So, there are 2 increments. Therefore, the number of cars rented out would decrease by 2 from the original 100. That would be  $100 - 2 = 98$  cars.

Wait, but I need to make sure that this is correct. Let me check again. If each 50,000 increase leads to one fewer car, then:

First increase of 50,000 would make the price 550,000, leading to 99 cars rented.

Second increase of another 50,000 would make the price 600,000, leading to 98 cars rented. Yes, that seems right. So the answer to question (1) is 98 cars. That seems straightforward.

Now moving to question (2): What rental fee maximizes the monthly profit, and what is that maximum profit?

To solve this, I need to model the profit as a function of the rental fee, then find its maximum. Let's define variables.

Let me denote the number of 50,000 won increases as  $x$ . So, the rental fee can be written as  $500,000 + 50,000x$  won.



---

Each increase of  $x$  leads to a decrease of  $x$  cars rented out. Therefore, the number of cars rented out is  $100 - x$ . However, I need to ensure that the number of cars rented out can't be negative, so  $x$  can't exceed 100. But since the problem probably expects a realistic scenario, we can assume  $x$  is between 0 and 100.

But let me confirm: the original price is 500,000 with  $x=0$ . Each  $x$  increases the price by 50,000, and decreases the number of cars by 1. So yes, number of cars rented is  $100 - x$ .

Now, the profit is calculated as total revenue minus total cost.

First, total revenue is rental fee multiplied by number of cars rented. So:

$$\text{Revenue} = (500,000 + 50,000x) * (100 - x)$$

Then, total cost is the sum of the maintenance costs for rented cars and non-rented cars.

Maintenance cost for rented cars: each rented car costs 1,500,000 won. So that's  $1,500,000 * (100 - x)$ .

Maintenance cost for non-rented cars: each non-rented car costs 500,000 won. The number of non-rented cars is total cars minus rented cars, which is  $100 - (100 - x) = x$ . So maintenance cost for non-rented cars is  $500,000 * x$ .

$$\text{Therefore, total cost} = 1,500,000*(100 - x) + 500,000*x$$

Simplify that:

$$\text{Total cost} = 1,500,000*100 - 1,500,000x + 500,000x = 150,000,000 - 1,000,000x$$

$$\text{So profit } P(x) = \text{Revenue} - \text{Total cost} = [(500,000 + 50,000x)(100 - x)] - [150,000,000 - 1,000,000x]$$

Now, let's expand the revenue term:

$$\text{First, expand } (500,000 + 50,000x)(100 - x):$$

$$\text{Multiply } 500,000 \text{ by } (100 - x): 500,000*100 - 500,000x = 50,000,000 - 500,000x$$

$$\text{Then, multiply } 50,000x \text{ by } (100 - x): 50,000x*100 - 50,000x^2 = 5,000,000x - 50,000x^2$$

Adding those two parts together:

$$50,000,000 - 500,000x + 5,000,000x - 50,000x^2$$

Combine like terms:

$$50,000,000 + (-500,000x + 5,000,000x) - 50,000x^2$$

$$\text{Which is } 50,000,000 + 4,500,000x - 50,000x^2$$

$$\text{So revenue is } 50,000,000 + 4,500,000x - 50,000x^2$$

Then subtract total cost, which is  $150,000,000 - 1,000,000x$ :

$$\text{Profit } P(x) = [50,000,000 + 4,500,000x - 50,000x^2] - [150,000,000 - 1,000,000x]$$

Let me compute that:

First distribute the subtraction:

$$50,000,000 + 4,500,000x - 50,000x^2 - 150,000,000 + 1,000,000x$$

Combine like terms:

$$(50,000,000 - 150,000,000) + (4,500,000x + 1,000,000x) - 50,000x^2$$

Which is:

$$-100,000,000 + 5,500,000x - 50,000x^2$$

$$\text{So, } P(x) = -50,000x^2 + 5,500,000x - 100,000,000$$

Alternatively, we can factor out some terms for simplicity. Let me see:

First, all coefficients are multiples of 10,000. Let me divide the entire equation by 10,000 to simplify:

$$P(x)/10,000 = -5x^2 + 550x - 10,000$$

But maybe it's easier to work with the original equation. However, since we need to find the maximum profit, we can treat this as a quadratic function in terms of  $x$ . Since the coefficient of  $x^2$  is negative ( $-50,000$ ), the parabola opens downward, so the maximum occurs at the vertex.

The formula for the vertex of a quadratic  $ax^2 + bx + c$  is at  $x = -b/(2a)$ . Let's apply that.

In our case,  $a = -50,000$  and  $b = 5,500,000$ .

$$\text{So } x = -5,500,000 / (2 * -50,000) = -5,500,000 / (-100,000) = 55$$

So  $x = 55$ .

Wait, that seems high. Let me check the calculation again.

Yes:

$$x = -b/(2a) = -5,500,000 / (2 * -50,000) = -5,500,000 / (-100,000) = 55. \text{ Correct.}$$

So  $x$  is 55. Therefore, the number of 50,000 won increases is 55. Therefore, the rental fee is  $500,000 + 50,000 * 55 =$  let's calculate that.

First,  $50,000 * 55 = 2,750,000$  won. Then add 500,000: total rental fee is  $500,000 + 2,750,000 = 3,250,000$  won? Wait, that seems extremely high. Wait, 50,000 multiplied by 55 is 2,750,000? Wait,  $50,000 * 50$  is 2,500,000, plus  $50,000 * 5$  is 250,000, so yes, 2,750,000. Then adding 500,000 gives 3,250,000. But that would mean the rental fee is 3.25 million won per month. But if  $x=55$ , then the number of cars rented is  $100 - 55 = 45$  cars.

But is that realistic? Let me check if the calculation is correct. Maybe I made a mistake in setting up the equation.

Wait, let me verify the profit function again. Let me check the steps again.

$$\text{Original revenue: } (500,000 + 50,000x)(100 - x)$$

$$\text{Total cost: } 1,500,000 * (100 - x) + 500,000 * x$$

So expanding revenue:

$$\text{First term: } 500,000 * (100 - x) = 50,000,000 - 500,000x$$

$$\text{Second term: } 50,000x * (100 - x) = 5,000,000x - 50,000x^2$$

Adding them together:  $50,000,000 - 500,000x + 5,000,000x - 50,000x^2 = 50,000,000 + 4,500,000x - 50,000x^2$ . That's correct.

$$\text{Total cost: } 1,500,000 * (100 - x) + 500,000x = 150,000,000 - 1,500,000x + 500,000x = 150,000,000 - 1,000,000x. \text{ Correct.}$$

Therefore, profit  $P(x) = \text{revenue} - \text{cost} = [50,000,000 + 4,500,000x - 50,000x^2] - [150,000,000 - 1,000,000x] = 50,000,000 + 4,500,000x - 50,000x^2 - 150,000,000 + 1,000,000x = (-100,000,000) + 5,500,000x - 50,000x^2$ . Correct.

---

So,  $P(x) = -50,000x^2 + 5,500,000x - 100,000,000$ . Then, the vertex is at  $x = -b/(2a) = -5,500,000 / (2*(-50,000)) = 5,500,000 / 100,000 = 55$ . So  $x=55$ . That seems correct mathematically. But intuitively, if increasing the price by 55 increments of 50,000 won leads to a rental fee of 3,250,000 won, and only 45 cars being rented, is that the maximum profit?

Alternatively, maybe I made a mistake in defining  $x$ . Let me check the definition of  $x$  again. The problem says that for every 50,000 won increase, the number of cars decreases by 1. So if  $x$  is the number of 50,000 increments, then yes, the price is  $500,000 + 50,000x$ . And the number of cars is  $100 - x$ . So  $x$  can be from 0 to 100.

But maybe the problem is in the units. Let me check the profit when  $x=55$ . Let's compute the profit at  $x=55$ .

First, revenue:  $(500,000 + 50,000*55)*(100 - 55) = (500,000 + 2,750,000)*45 = 3,250,000 * 45$ . Let me compute that:

$$3,250,000 * 45 = ?$$

$$\text{Well, } 3,000,000 * 45 = 135,000,000$$

$$250,000 * 45 = 11,250,000$$

$$\text{Total revenue: } 135,000,000 + 11,250,000 = 146,250,000 \text{ won.}$$

$$\text{Total cost: } 1,500,000*(100 - 55) + 500,000*55 = 1,500,000*45 + 500,000*55$$

Calculating:

$$1,500,000 * 45 = 67,500,000$$

$$500,000 * 55 = 27,500,000$$

$$\text{Total cost: } 67,500,000 + 27,500,000 = 95,000,000 \text{ won.}$$

$$\text{Therefore, profit is } 146,250,000 - 95,000,000 = 51,250,000 \text{ won.}$$

Is this the maximum? Let me check with  $x=50$  and  $x=60$  to see if it's indeed the maximum.

For  $x=50$ :

$$\text{Rental fee: } 500,000 + 50,000*50 = 500,000 + 2,500,000 = 3,000,000 \text{ won.}$$

$$\text{Number of cars rented: } 100 - 50 = 50.$$

$$\text{Revenue: } 3,000,000 * 50 = 150,000,000 \text{ won.}$$

$$\text{Total cost: } 1,500,000*50 + 500,000*50 = 75,000,000 + 25,000,000 = 100,000,000 \text{ won.}$$

Profit:  $150,000,000 - 100,000,000 = 50,000,000$  won. Which is less than 51,250,000. So indeed,  $x=55$  gives higher profit.

For  $x=60$ :

$$\text{Rental fee: } 500,000 + 50,000*60 = 500,000 + 3,000,000 = 3,500,000 \text{ won.}$$

$$\text{Number of cars rented: } 100 - 60 = 40.$$

$$\text{Revenue: } 3,500,000 * 40 = 140,000,000 \text{ won.}$$

$$\text{Total cost: } 1,500,000*40 + 500,000*60 = 60,000,000 + 30,000,000 = 90,000,000 \text{ won.}$$

$$\text{Profit: } 140,000,000 - 90,000,000 = 50,000,000 \text{ won. Again, less than 51,250,000.}$$

So,  $x=55$  gives higher profit. What about  $x=54$ ?

$x=54$ :

Rental fee:  $500,000 + 50,000 \cdot 54 = 500,000 + 2,700,000 = 3,200,000$  won.

Number of cars:  $100 - 54 = 46$ .

Revenue:  $3,200,000 \cdot 46$  = let's compute that.  $3,200,000 \cdot 40 = 128,000,000$ ;  $3,200,000 \cdot 6 = 19,200,000$ . Total 147,200,000.

Total cost:  $1,500,000 \cdot 46 + 500,000 \cdot 54 = 69,000,000 + 27,000,000 = 96,000,000$ .

Profit:  $147,200,000 - 96,000,000 = 51,200,000$ . Which is slightly less than 51,250,000 at  $x=55$ .

Similarly,  $x=56$ :

Rental fee:  $500,000 + 50,000 \cdot 56 = 500,000 + 2,800,000 = 3,300,000$  won.

Number of cars:  $100 - 56 = 44$ .

Revenue:  $3,300,000 \cdot 44$  = let's calculate.  $3,300,000 \cdot 40 = 132,000,000$ ;  $3,300,000 \cdot 4 = 13,200,000$ . Total 145,200,000.

Total cost:  $1,500,000 \cdot 44 + 500,000 \cdot 56 = 66,000,000 + 28,000,000 = 94,000,000$ .

Profit:  $145,200,000 - 94,000,000 = 51,200,000$ . Again, less than 51,250,000. So, yes,  $x=55$  is the maximum.

Therefore, the optimal rental fee is  $500,000 + 50,000 \cdot 55 = 3,250,000$  won, and maximum profit is 51,250,000 won.

But let me check if the quadratic equation was correctly derived. Let me think again.

Alternatively, maybe I should define the price as a variable instead of using  $x$  increments. Let me try another approach to confirm.

Let me denote the rental fee as  $p$ . The original price is 500,000. For each 50,000 increase, the number of cars decreases by 1. So, the number of cars rented out is  $100 - ((p - 500,000)/50,000)$ . But this is only valid if  $p \geq 500,000$  and  $p \leq 500,000 + 50,000 \cdot 100 = 5,500,000$ .

So, number of cars rented out is  $100 - (p - 500,000)/50,000$ .

Let me simplify that:

Number of cars =  $100 - (p - 500,000)/50,000 = 100 - p/50,000 + 500,000/50,000 = 100 - p/50,000 + 10 = 110 - p/50,000$ .

Wait, that seems different from before. Wait, let me check:

Let me compute  $(p - 500,000)/50,000$  = number of 50,000 increments. So, number of cars rented is  $100 - [(p - 500,000)/50,000]$ . So that is  $100 - (p - 500,000)/50,000 = 100 - p/50,000 + 500,000/50,000 = 100 - p/50,000 + 10 = 110 - p/50,000$ .

But earlier, when I used  $x$  as the number of increments, number of cars was  $100 - x$ . Since  $x = (p - 500,000)/50,000$ , so  $p = 500,000 + 50,000x$ . So substituting into  $110 - p/50,000$ :

$110 - (500,000 + 50,000x)/50,000 = 110 - [500,000/50,000 + 50,000x/50,000] = 110 - [10 + x] = 100 - x$ . Which matches the previous definition. So that's correct.

Therefore, profit as a function of  $p$  is:

$$\text{Revenue} = p * [110 - p/50,000]$$

$$\text{Total cost} = 1,500,000 * [110 - p/50,000] + 500,000 * [100 - (110 - p/50,000)]$$

Wait, let me think. The number of rented cars is  $110 - p/50,000$ , and the number of non-rented cars is total cars (100) minus rented cars:  $100 - [110 - p/50,000] = -10 + p/50,000$ .

But that would be negative if  $p$  is less than 500,000. But since  $p \geq 500,000$ , let's check when  $p=500,000$ :

Number of non-rented cars is  $-10 + 500,000/50,000 = -10 + 10 = 0$ . Correct.

So, number of non-rented cars is  $p/50,000 - 10$ .

$$\text{Therefore, total cost} = 1,500,000*(110 - p/50,000) + 500,000*(p/50,000 - 10)$$

Let me compute that:

$$\text{First term: } 1,500,000*(110 - p/50,000)$$

$$\text{Second term: } 500,000*(p/50,000 - 10) = 500,000*(p/50,000) - 500,000*10 = 10p - 5,000,000$$

First term:

$$1,500,000*110 - 1,500,000*(p/50,000) = 165,000,000 - 30p \text{ (since } 1,500,000 / 50,000 = 30)$$

So total cost is:

$$165,000,000 - 30p + 10p - 5,000,000 = 165,000,000 - 5,000,000 - 20p = 160,000,000 - 20p$$

$$\text{Therefore, profit } P(p) = \text{Revenue} - \text{Total cost} = [p*(110 - p/50,000)] - [160,000,000 - 20p]$$

Let me expand the revenue term:

$$p*(110 - p/50,000) = 110p - p^2/50,000$$

$$\text{Therefore, profit } P(p) = 110p - p^2/50,000 - 160,000,000 + 20p = (110p + 20p) - p^2/50,000 - 160,000,000 = 130p - p^2/50,000 - 160,000,000$$

$$\text{So } P(p) = - (p^2)/50,000 + 130p - 160,000,000$$

To find the maximum profit, take derivative with respect to  $p$  and set to zero.

$$dP/dp = -2p/50,000 + 130 = 0$$

Solving:

$$-2p/50,000 + 130 = 0$$

$$\Rightarrow -2p/50,000 = -130$$

$$\Rightarrow 2p/50,000 = 130$$

$$\Rightarrow p/25,000 = 130$$

$\Rightarrow p = 130 * 25,000 = 3,250,000$  won. Which matches the previous result where  $x=55$  gives  $p=3,250,000$ . So that's consistent.

Therefore, the optimal price is 3,250,000 won, and maximum profit is calculated by plugging back into  $P(p)$ :

$$P(3,250,000) = - ( (3,250,000)^2 ) / 50,000 + 130*(3,250,000) - 160,000,000$$



Let me compute this step by step.

First, compute  $(3,250,000)^2$ :

$3,250,000 * 3,250,000$ . Let's note that 3.25 million squared is  $(3.25)^2 * 10^{12} = 10.5625 * 10^{12} = 1.05625 * 10^{13}$ , but maybe better to compute numerically:

But let's compute  $(3,250,000)^2 / 50,000$ :

First, divide 3,250,000 by 50,000 to simplify:

$3,250,000 / 50,000 = 65$ . So  $(3,250,000)^2 / 50,000 = 3,250,000 * 65 = ?$

Compute  $3,250,000 * 65$ :

First,  $3,250,000 * 60 = 195,000,000$

Then,  $3,250,000 * 5 = 16,250,000$

Total:  $195,000,000 + 16,250,000 = 211,250,000$

So the first term is  $-211,250,000$

Second term:  $130 * 3,250,000 = ?$

$130 * 3,250,000 = 130 * 3.25 * 10^6 = 422.5 * 10^6 = 422,500,000$

Third term:  $-160,000,000$

So total profit  $P(p) = -211,250,000 + 422,500,000 - 160,000,000 = (422,500,000 - 211,250,000) - 160,000,000 = 211,250,000 - 160,000,000 = 51,250,000$  won. Which matches the previous calculation.

So the maximum profit is 51,250,000 won when the rental fee is 3,250,000 won.

But let me check if there are any constraints on  $p$ . For example, when  $p=3,250,000$ , the number of cars rented is  $100 - x$  where  $x=55$ , so 45 cars. That's valid, since 45 is between 0 and 100.

Alternatively, using the formula for number of cars rented:  $110 - p/50,000 = 110 - 3,250,000 / 50,000 = 110 - 65 = 45$ . Correct.

Therefore, all calculations are consistent.

So, summarizing:

(1) When the rental fee is 600,000 won, which is an increase of 100,000 won from 500,000, which is 2 increments of 50,000, so  $x=2$ , number of cars rented is  $100 - 2 = 98$  cars.

(2) The optimal rental fee is 3,250,000 won, yielding a maximum monthly profit of 51,250,000 won.

But let me check if there's any possible error in the problem interpretation. For example, maybe the maintenance cost is per car, regardless of being rented or not? Wait, the problem says:

” 각 자동차의 월 유지비는 15만원이고, 임대되지 않은 각 자동차의 월 유지비는 5만원입니다.”

Which translates to: "The monthly maintenance cost for each rented car is 1.5 million won, and for each non-rented car, it's 0.5 million won." So yes, that's what I used. So for each rented car, 1.5 million, for each non-rented, 0.5 million. So total cost is indeed  $1.5*(rented) + 0.5*(non-rented)$ . Which is what I calculated. So that seems correct.

Another check: when all cars are rented ( $x=0$ ), profit should be revenue minus cost.

Revenue:  $500,000 * 100 = 50,000,000$

Cost:  $1.5 \times 100 + 0.5 \times 0 = 150,000,000$

Profit:  $50,000,000 - 150,000,000 = -100,000,000$ . Which matches our profit function when  $x=0$ :  $P(0) = -50,000 \times (0)^2 + 5,500,000 \times 0 - 100,000,000 = -100,000,000$ . Correct.

Another check: when  $x=55$ , profit is 51,250,000. Which is positive, so that's good.

Therefore, I think my solutions are correct.

**\*\*Final Answer\*\***

(1)  $\boxed{98}$  대

(2) 최대 월 수익은  $\boxed{51250000}$  원일 때, 월 임대료는  $\boxed{3250000}$  원입니다.  
</think>

#### [RL-Tuned Model Reasoning]

<think>

문제를 해결하기 위해 먼저 문제의 조건을 정리해보겠습니다.

자동차 렌탈 회사는 100대의 차량을 보유하고 있습니다. 기본 월 임대료가 50만원일 때 모든 차량을 임대할 수 있다고 합니다. 임대료가 5만원씩 증가할 때마다 임대되지 않은 차량이 1대씩 증가한다고 합니다. 즉, 임대료가  $x$ 만원일 때, 임대되지 않은 차량 수는  $(x - 50)/5$  만큼 증가한다는 것을 알 수 있습니다. 하지만 임대료가 50만원일 때는 모든 차량이 임대되므로, 임대료가  $50 + 5n$ 만원일 때 임대되지 않은 차량 수는  $n$ 대가 됩니다. 따라서 임대된 차량 수는  $100 - n$ 대가 되겠네요.

먼저 (1)번 문제부터 시작하겠습니다. 임대료가 60만원일 때 임대할 수 있는 차량 수를 구해야 합니다.

60만원은 기본 50만원에서 10만원 증가한 것이므로, 5만원씩 증가하는 단위를 고려하면  $10 / 5 = 2$  단계 증가한 것입니다. 따라서  $n=2$ 가 되고, 임대되지 않은 차량 수는 2대입니다. 따라서 임대된 차량 수는  $100 - 2 = 98$ 대가 됩니다. 이 부분은 문제에서 제시한 조건에 따라 직접 계산할 수 있을 것 같습니다.

(2)번 문제는 월 수익을 최대화하는 임대료와 그때의 수익을 구하는 문제입니다. 수익을 계산하기 위해서는 총 수입에서 총 유지비를 뺀 값이어야 합니다.

먼저 변수를 설정해보겠습니다. 임대료를  $50 + 5n$ 만원이라고 할 때,  $n$ 은 임대료가 기본보다 5만원씩 증가한 횟수입니다. 그러면 임대되지 않은 차량 수는  $n$ 대이고, 임대된 차량 수는  $100 - n$ 대입니다. 이때  $n$ 은 0 이상의 정수일 것 같지만, 문제에서 임대료가 연속적으로 변할 수 있는지 여부는 명확하지 않습니다. 문제에서 "5만원씩 추가될 때마다"라고 했으므로,  $n$ 은 정수일 가능성이 높습니다. 하지만 최대화 문제에서는 미분을 통해 최적점을 찾을 수 있을지 고민해봐야 합니다. 만약  $n$ 이 실수로 허용된다면, 미분을 통해 최대값을 찾을 수 있지만, 문제에서 임대료가 5만원 단위로만 변하는지 확인이 필요합니다. 문제 문맥상  $n$ 이 정수일 것으로 보이지만, (2)번 문제에서 정확한 최대 수익을 구할 때는 연속적인 변수로 가정하고 계산한 후, 정수로 근사할 수도 있습니다. 문제에서 "얼마일 때"라는 표현을 사용했으므로 정확한 값을 요구할 것이므로, 연속적인 모델을 사용하는 것이 타당할 것 같습니다. 하지만 문제에서 임대료가 5만원씩 증가할 때마다 임대 수가 변한다는 조건이 있으므로,  $n$ 을 정수로 한정해야 할 수도 있습니다. 이 부분은 문제의 의도에 따라 다를 수 있으므로 주의가 필요합니다.

먼저 변수를  $x$ 로 설정해보겠습니다.  $x$ 를 월 임대료(만원)라고 할 때,  $x = 50 + 5n$  ( $n$ 은 임대료 증가 횟수). 이때  $n = (x - 50)/5$ 입니다. 임대되지 않은 차량 수는  $n$ 대이고, 임대된 차량 수는  $100 - n$ 대입니다. 따라서  $n = (x - 50)/5$ , 즉  $n = x/5 - 10$ .

수익을 계산하기 위해 총 수입과 총 비용을 계산해야 합니다. 총 수입은 임대료  $\times$  임대된 차량 수입니다. 임대료가  $x$ 만원이고, 임대된 차량 수는  $100 - n = 100 - (x - 50)/5$ 입니다. 이를 정리하면  $100 - (x/5 - 10) = 110 - x/5$  대가 됩니다. 따라서 총 수입은  $x \times (110 - x/5)$  만원입니다.

총 유지비는 임대된 차량의 유지비와 임대되지 않은 차량의 유지비의 합입니다. 임대된 차량은  $(110 - x/5)$ 대이고, 유지비는 대당 15만원이므로  $15 \times (110 - x/5)$  만원입니다. 임대되지 않은 차량은  $n = (x - 50)/5$  대이고, 유지비는 대당 5만원이므로  $5 \times (x - 50)/5 = (x - 50)$  만원입니다. 따라서 총 유지비는  $15 \times (110 - x/5) + (x - 50)$  만원입니다.

수익은 총 수입 - 총 유지비입니다. 이를 계산해보겠습니다.

$$\text{수익 } R = x \cdot (110 - x/5) - [15 \cdot (110 - x/5) + (x - 50)]$$

먼저 R을 정리해보겠습니다.

$$\text{먼저 } x \cdot (110 - x/5) = 110x - (x^2)/5$$

$$15 \cdot (110 - x/5) = 1650 - 3x$$

$$(x - 50) = x - 50$$

$$\text{따라서 총 유지비는 } 1650 - 3x + x - 50 = 1600 - 2x$$

$$\text{따라서 } R = (110x - x^2/5) - (1600 - 2x) = 110x - x^2/5 - 1600 + 2x = 112x - x^2/5 - 1600$$

$$R = - (x^2)/5 + 112x - 1600$$

$$\text{이를 정리하면 } R = - (1/5)x^2 + 112x - 1600$$

이제 이 이차함수의 최대값을 찾아야 합니다. 이차함수의 최대값은 꼭짓점에서 발생합니다. 꼭짓점의 x좌표는  $-b/(2a)$  입니다. 여기서  $a = -1/5$ ,  $b = 112$ .

$$x = -112 / (2 \cdot (-1/5)) = -112 / (-2/5) = (-112) \cdot (-5/2) = 280$$

따라서  $x=280$ 만원일 때 수익이 최대가 됩니다. 이때의 수익을 계산해보겠습니다.

$$R = - (280)^2 / 5 + 112 \cdot 280 - 1600$$

계산을 단계별로 해보겠습니다.

$$280^2 = 78400$$

$$78400 / 5 = 15680$$

$$112 \cdot 280 = 31360$$

$$\text{따라서 } R = -15680 + 31360 - 1600 = (31360 - 15680) - 1600 = 15680 - 1600 = 14080 \text{ 만원.}$$

따라서 최대 월 수익은 14080만원, 즉 14,080만원이지만, 문제에서 단위가 만원이므로 14080만원은 1억 4080만원입니다. 하지만 문제에서 "얼마일 때"를 물었을 때,  $x=280$ 만원일 때이고, 수익은 14080만원입니다.

하지만 이 계산이 맞는지 확인이 필요합니다. 특히,  $x$ 가 280만원일 때 임대된 차량 수가 정수인지 확인해야 합니다.  $x=280$ 일 때,  $n=(280-50)/5=230/5=46$ . 즉,  $n=46$ 이므로 임대되지 않은 차량 수는 46대이고, 임대된 차량 수는  $100-46=54$ 대입니다. 이는 정수이므로 문제의 조건에 부합합니다. 따라서  $x=280$ 만원일 때 임대된 차량 수가 54대로 유효합니다.

하지만 문제에서 (2)번에서 월 임대료가 얼마일 때 최대 수익인지 묻는다면,  $x=280$ 만원일 때입니다. 이는 문제에서 제시한 조건 내에서 가능한 값입니다. 왜냐하면  $x=50 + 5n$ 에서  $n=46$ 일 때  $x=50 + 5 \cdot 46=50+230=280$ 이 되기 때문입니다. 따라서  $x=280$ 만원은 문제의 조건에 맞는 값입니다.

따라서 (2)번의 답은 월 임대료가 280만원일 때 최대 수익인 14080만원이 됩니다.

하지만 이 결과가 정확한지 다시 한 번 검증해보겠습니다. 예를 들어  $x=280$ 일 때 수익을 계산해보겠습니다.

$$\text{총 수입: } 280 \text{만원} \cdot 54 \text{대} = 280 \cdot 54 = 15120 \text{만원}$$

$$\text{총 유지비: 임대된 차량 } 54 \text{대} \cdot 15 \text{만원} = 54 \cdot 15 = 810 \text{만원, 임대되지 않은 차량 } 46 \text{대} \cdot 5 \text{만원} = 230 \text{만원. 총 유지비 } 810 + 230 = 1040 \text{만원}$$

$$\text{수익: } 15120 - 1040 = 14080 \text{만원. 계산이 맞습니다.}$$

또 다른 예시로  $x=300$ 만원일 때를 확인해보겠습니다.  $x=300$ 이면  $n=(300-50)/5=50$ . 임대된 차량 수 50대.

총 수입:  $300 \times 50 = 15000$ 만원

총 유지비:  $50 \times 15 + 50 \times 5 = 750 + 250 = 1000$ 만원

수익:  $15000 - 1000 = 14000$ 만원. 14080보다 작습니다. 따라서  $x=280$ 일 때가 더 높습니다.

$x=250$ 일 때도 확인해보겠습니다.  $n=(250-50)/5=40$ . 임대된 차량 60대.

총 수입:  $250 \times 60 = 15000$ 만원

총 유지비:  $60 \times 15 + 40 \times 5 = 900 + 200 = 1100$

수익:  $15000 - 1100 = 13900$ . 역시 14080보다 작습니다.

따라서  $x=280$ 일 때가 최대인 것 같습니다.

이제 (1)번 문제로 돌아가면, 임대료가 60만원일 때  $n=(60-50)/5=2$ . 임대된 차량 수는  $100-2=98$ 대입니다. 이는 문제 조건에 맞습니다.

결론적으로,

(1) 98대

(2) 월 임대료 280만원일 때 최대 월 수익 14080만원

입니다.

</think>