Kimi k1.5

KIMI K1.5
深度解读

ZOMI

视频目录大纲

1. KIMI K1.5 技术文章解读
2. KIMI1.5 和 DeepSeek-R1 比较
3. KIMI1.5 & DeepSeek-R1 在线测试
4. 对产业的思考与小结

01

KIMI K1.5

技术文章解读

数据准备

- 在强化学习阶段，Kimi k1.5的数据准备主要包括编写代码测试用例、数学问题奖励建模和处理多模态视觉数据。
- 编程问题：使用CYaRon库自动生成测试用例，并通过运行这些测试用例来验证模型生成的代码。
- 数学问题：采用基于链式推理的奖励模型，生成逐步的推理过程，并以JSON格式提供最终的正确性判断，提供更强大和可解释的奖励信号，以更准确地评估答案的正确性。
- 视觉数据：包括真实世界数据、合成视觉推理数据和文本渲染数据，以增强模型的视觉推理能力。

Long-CoT to Shot-COT

- 模型合并：
  - 之前都是通过模型合并来提高模型的泛化性，k1.5发现long-cot模型和short-cot模型也可以合并，从而提高输出效率，中和输出内容，并且无需训练。
- 最短拒绝采样：
  - 对于模型输出结果进行n次采样（实验中n=8），选择最短的正确结果进行模型微调。

Long-CoT to Shot-COT

- DPO：
  - 与最短拒绝采样类似，利用 long-cot 模型生成多个输出结果，将最短的正确输出作为正样本，而较长的响应作为负样本，通过构造的正负样本进行DPO偏好学习。
- Long2Short 强化：
  - 在标准的强化学习训练阶段之后，选择一个在性能和输出效率之间达到最佳平衡的模型作为基础模型，并进行单独的long-cot到short-cot的强化学习训练阶段。在这一阶段，采用长度惩罚，进一步惩罚超出期望长度，但保证模型仍然可能正确的输出答案。

高质量 RL 提示数据三要素

- 覆盖范围-广：
  - 提示数据应涵盖广泛的学科领域，如科学、技术、工程和数学、代码和一般推理，增强模型在不同领域的普适性。开发了一个标签系统，对提示按照领域和学科进行分类，确保不同学科领域数据平衡。
- 难度分布-均：
  - 提示数据应包含易、中、难不同难度级别的问题，让模型逐步学习，防止模型过拟合到一些特定复杂的问题上。这里k1.5通过模型自身的推理能力，来评估每个prompt的难度，就是对相同的prompt利用相对较高温度生成10次答案，然后计算答案的通过率，通过率越低，代表prompt难度越高。
- 可评估性-准：
  - 提示数据应允许验证器进行客观且可靠的评估，确保模型结果是基于正确的推理过程，而不是简单模式或随机猜测。这里k1.5利用没有任何链式推理步骤的情况下预测可能的答案，如果在N次尝试内，均预测正确答案，认为该prompt容易产生reward hacking。

核心技术

- 传统的 RL 训练中，对于长文本序列，模型通常需要一次性生成整个轨迹（例如，完整的 CoT 推理过程）。这直接导致计算开销大、内存占用高、训练不稳定，以及数据效率低这些问题。
- kimi团队在扩展上下文长度的情况下，采用了一个关键的提高训练效率的技术——partial rollouts。它的核心思想是将长轨迹的生成过程分割成多个迭代步骤，避免一次性生成整个轨迹，从而提高训练效率并节省计算资源。
- Partial rollouts 显著提升了长上下文任务的处理能力，优化了计算资源，系统能够生成更长的响应，还在不牺牲输出质量的前提下加速了模型训练过程。

Partial rollouts 的工作方式

1. 固定输出 token 预算。定义一个固定输出 token 数量，作为每次 rollouts 的长度上限（例如，每次生成 500 个 token）。
2. 分段生成。在每次训练迭代中，模型并不生成完整的轨迹，而是只生成部分轨迹，即在 token 数量达到预算上限时停止。
3. 保存中间状态。将生成的中间轨迹片段及其对应的模型状态保存到 replay buffer 中。
4. 多次迭代完成长轨迹。在后续的训练迭代中，模型可以从 replay buffer 中读取之前保存的中间轨迹片段，并在此基础上继续生成新的轨迹片段。通过多次迭代，最终完成整个轨迹的生成。
5. 选择性计算 loss。在计算 loss 时，可以选择只计算当前迭代生成的部分轨迹片段的 损失，也可以计算整个轨迹的 loss，具体策略取决于实际情况。

RL阶段

- 训练阶段：
  - Megatron 和 vLLM 分别在独立的容器中运行，容器称为 checkpoint-engine 的外壳进程封装。Megatron 首先启动训练过程，训练完成后，Megatron 会释放 GPU 内存，并准备将当前权重传递给 vLLM。
- 推理阶段：
  - 在 Megatron 释放内存后，vLLM 以虚拟模型权重启动，并通过 Mooncake 从 Megatron 接收最新的权重更新。完成回放后，checkpoint-engine 会停止所有 vLLM 进程。
- 后续训练阶段：
  - 释放 vLLM 所占用的内存后，Megatron 重新加载内存并开始下一轮训练。

02

KIMI1.5 vs DeepSeek-R1

相似

相似	区别
抛弃 MCTS 复杂树搜索，用 CoT data 作为 SFT 或者 Post-Training 阶段	
减少 RL 额外部署一个 Value Model 来提供 Value Funcion	
减少 RL 过程对 Reward 的复杂奖励，通过最终结果引导模型学习	

区别

DeepSeek-R1	KIMI K1.5
DeepSeek-R1 通过 RL 自学习，直接冷启动，然后 CoT 微调迭代	KIMI K1.5 通过 复杂提示工程 Prompt Engine 来构建 CoT 然后使用 SFT 预热
DeepSeek-R1 已经权重开源，并且可在 chat DeepSeek 中使用，遵循 MIT 许可协议；	KIMI 目前还在灰度上线，短期内不会开源，走闭源的路线，提供线上服务；
RL 过程通过惩罚函数实现 Long-CoT 的 hark reward	使用Long-COT向Short-CoT的知识迁移
聚焦 NLP 领域，通过蒸馏成小模型，提升小模型效果	具备多模态能力，在 MathVista 基准可测评
技术文章内容纯粹，但是隐藏了细节	在数据准备和宏观概念上技术细节充分，但是缺乏实操性

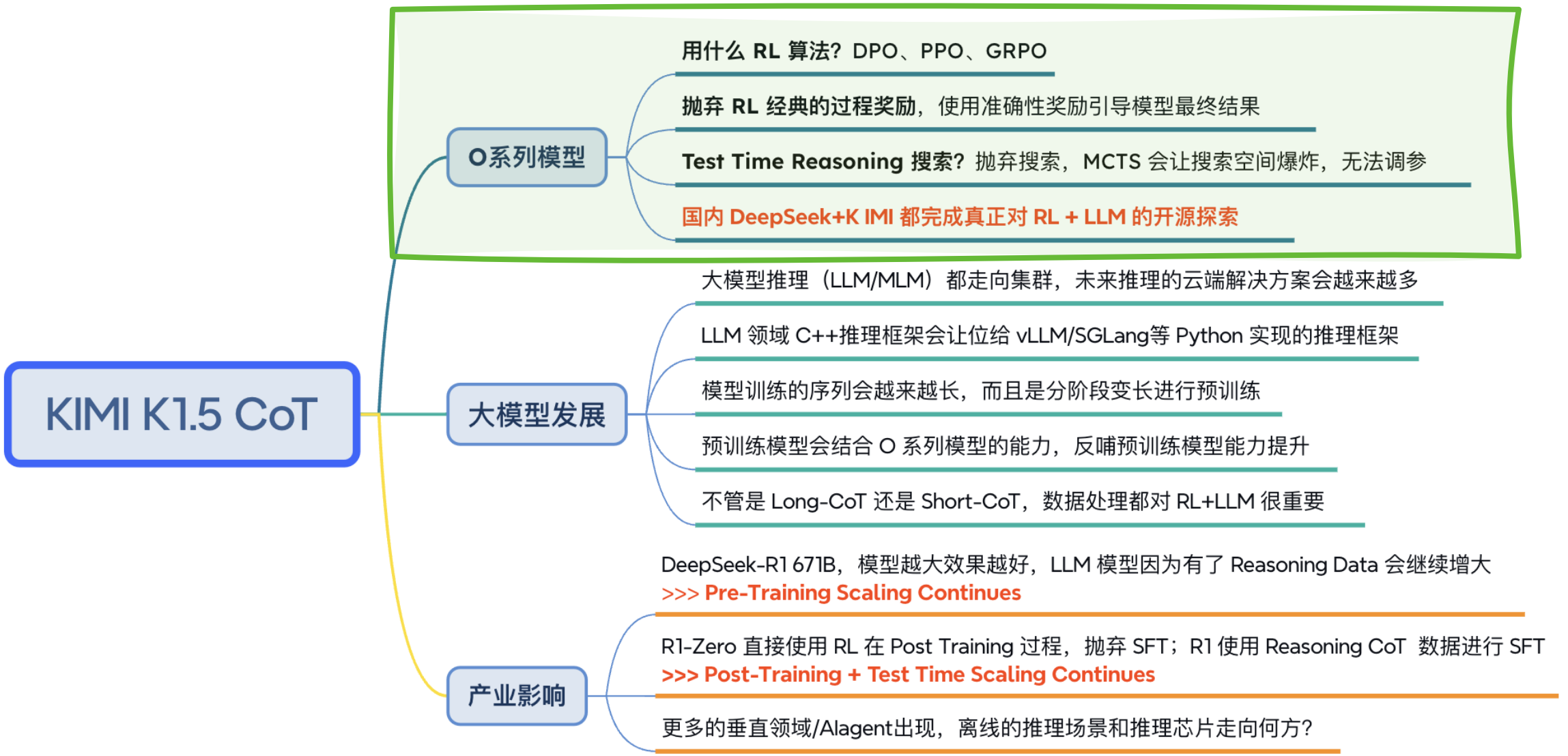
03

实验小案例

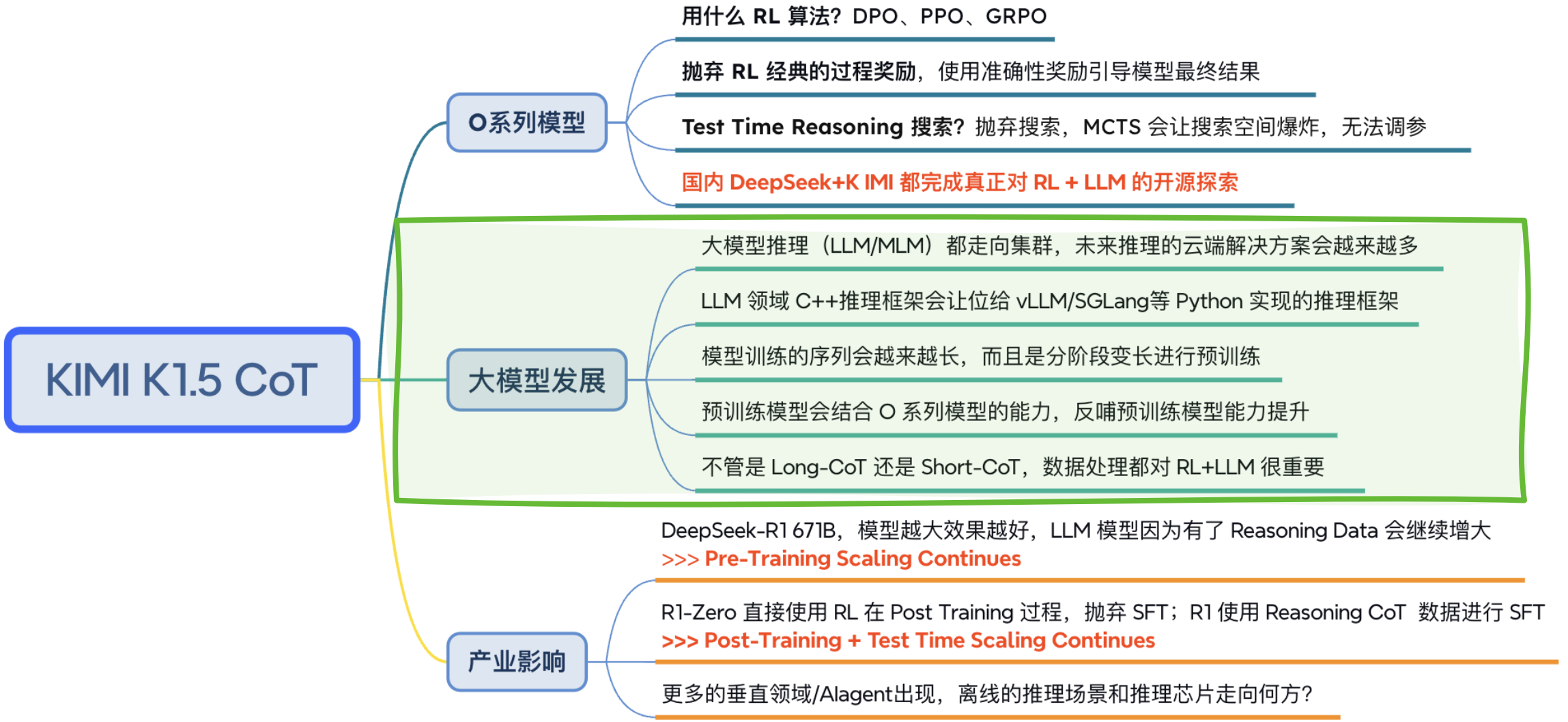
04

对产业思考与小结

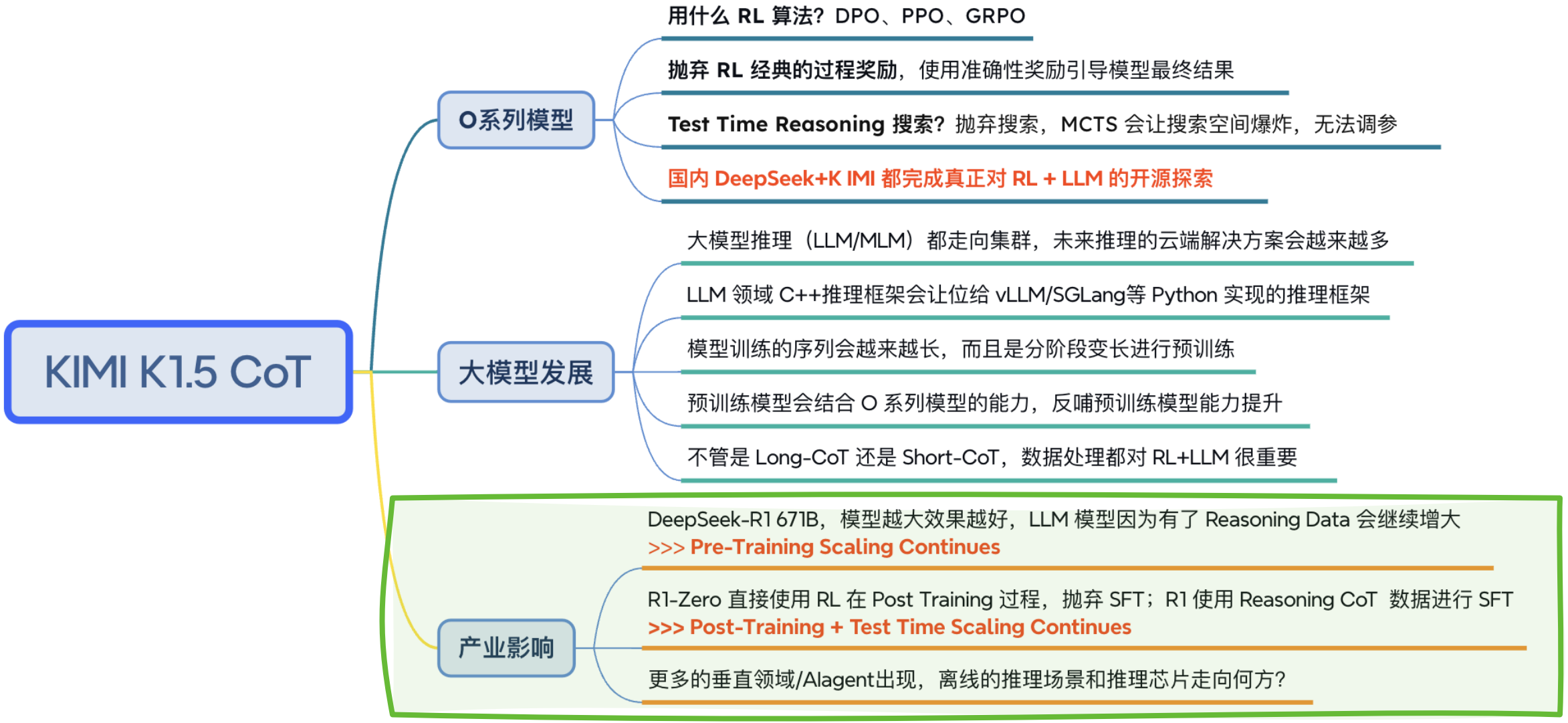
总结与思考


KIMI K1.5 CoT
O系列模型
用什么 RL 算法？DPO、PPO、GRPO
抛弃 RL 经典的过程奖励，使用准确性奖励引导模型最终结果
Test Time Reasoning 搜索？抛弃搜索，MCTS 会让搜索空间爆炸，无法调参
国内 DeepSeek+K IMI 都完成真正对 RL + LLM 的开源探索
大模型发展
大模型推理（LLM/MLM）都走向集群，未来推理的云端解决方案会越来越多
LLM 领域 C++推理框架会让位给 vLLM/SGLang等 Python 实现的推理框架
模型训练的序列会越来越长，而且是分阶段变长进行预训练
预训练模型会结合 O 系列模型的能力，反哺预训练模型能力提升
不管是 Long-CoT 还是 Short-CoT，数据处理都对 RL+LLM 很重要
产业影响
DeepSeek-R1 671B，模型越大效果越好，LLM 模型因为有了 Reasoning Data 会继续增大
>>> Pre-Training Scaling Continues
R1-Zero 直接使用 RL 在 Post Training 过程，抛弃 SFT；R1 使用 Reasoning CoT 数据进行 SFT
>>> Post-Training + Test Time Scaling Continues
更多的垂直领域/Alagent出现，离线的推理场景和推理芯片走向何方？

总结与思考


KIMI K1.5 CoT
O系列模型
用什么 RL 算法？DPO、PPO、GRPO
抛弃 RL 经典的过程奖励，使用准确性奖励引导模型最终结果
Test Time Reasoning 搜索？抛弃搜索，MCTS 会让搜索空间爆炸，无法调参
国内 DeepSeek+K IMI 都完成真正对 RL + LLM 的开源探索
大模型发展
大模型推理（LLM/MLM）都走向集群，未来推理的云端解决方案会越来越多
LLM 领域 C++推理框架会让位给 vLLM/SGLang等 Python 实现的推理框架
模型训练的序列会越来越长，而且是分阶段变长进行预训练
预训练模型会结合 O 系列模型的能力，反哺预训练模型能力提升
不管是 Long-CoT 还是 Short-CoT，数据处理都对 RL+LLM 很重要
产业影响
DeepSeek-R1 671B，模型越大效果越好，LLM 模型因为有了 Reasoning Data 会继续增大
>>> Pre-Training Scaling Continues
R1-Zero 直接使用 RL 在 Post Training 过程，抛弃 SFT；R1 使用 Reasoning CoT 数据进行 SFT
>>> Post-Training + Test Time Scaling Continues
更多的垂直领域/Alagent出现，离线的推理场景和推理芯片走向何方？

总结与思考


KIMI K1.5 CoT
O系列模型
用什么 RL 算法？DPO、PPO、GRPO
抛弃 RL 经典的过程奖励，使用准确性奖励引导模型最终结果
Test Time Reasoning 搜索？抛弃搜索，MCTS 会让搜索空间爆炸，无法调参
国内 DeepSeek+K IMI 都完成真正对 RL + LLM 的开源探索
大模型发展
大模型推理（LLM/MLM）都走向集群，未来推理的云端解决方案会越来越多
LLM 领域 C++推理框架会让位给 vLLM/SGLang等 Python 实现的推理框架
模型训练的序列会越来越长，而且是分阶段变长进行预训练
预训练模型会结合 O 系列模型的能力，反哺预训练模型能力提升
不管是 Long-CoT 还是 Short-CoT，数据处理都对 RL+LLM 很重要
产业影响
DeepSeek-R1 671B，模型越大效果越好，LLM 模型因为有了 Reasoning Data 会继续增大
>>> Pre-Training Scaling Continues
R1-Zero 直接使用 RL 在 Post Training 过程，抛弃 SFT；R1 使用 Reasoning CoT 数据进行 SFT
>>> Post-Training + Test Time Scaling Continues
更多的垂直领域/Alagent出现，离线的推理场景和推理芯片走向何方？

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

GitHub https://github.com/chenzomi12/AIFoundation

留给读者思考

1. KIMI K1.5 模型相比之前的模型有哪些创新点和改进之处?
2. 文中提到的链式思维（CoT）是如何应用在模型训练中的？它如何提高模型的推理能力?
3. KIMI K1.5 模型的训练和推理部署策略是如何设计的？有哪些优势和特点?
4. 链式思维和长度奖励对模型的性能有什么影响？它们在不同领域和任务中是否都适用?
5. 文章中提到了模型大小和上下文长度扩展问题，如何选择合适模型大小和上下文长度，来平衡性能和计算资源的利用?

参考与引用

- https://www.zhihu.com/question/10114790245
- https://github.com/MoonshotAI/Kimi-k1.5

- PPT 开源在：
- https://github.com/chenzomi12/AIFoundation/tree/main/09News/00Others