杨紫童

Zitong Yang

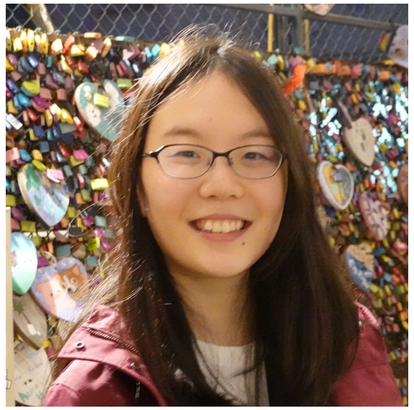
BAAI 智源Talk,
2025年2月18日

https://arxiv.org/abs/2501.19393

合作者

Niklas Muennighoff*

Weijia Shi*



Xiang Lisa Li*

Li Fei-Fei

Hannaneh Hajishirzi

Luke Zettlemoyer

Percy Liang

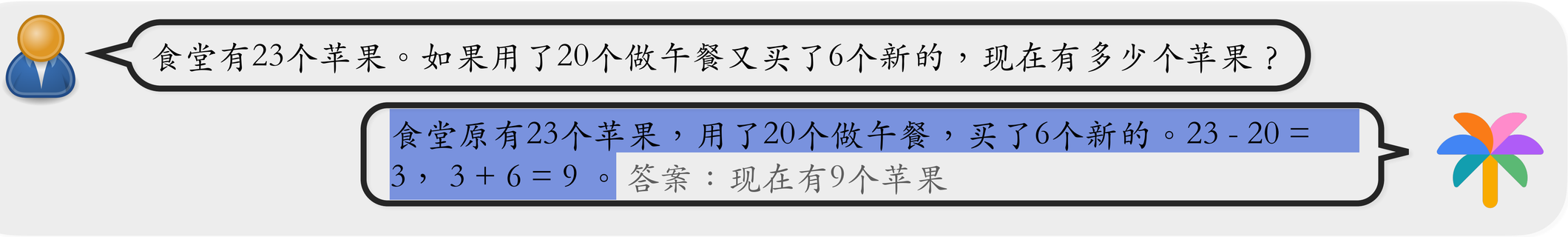
Emmanuel Candès

Tatsunori Hashimoto

** Equal contribution*

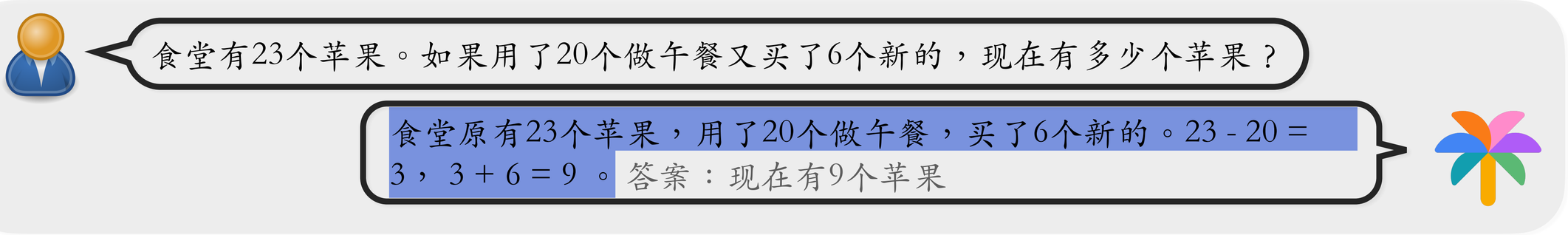
关于Test-compute scaling的早期探索（o1之前）

- 让模型在回答前“打草稿”： Chain of Thought (Nye et al., 2021; Wei et al., 2022), STaR (Zelikman et al, 2022)

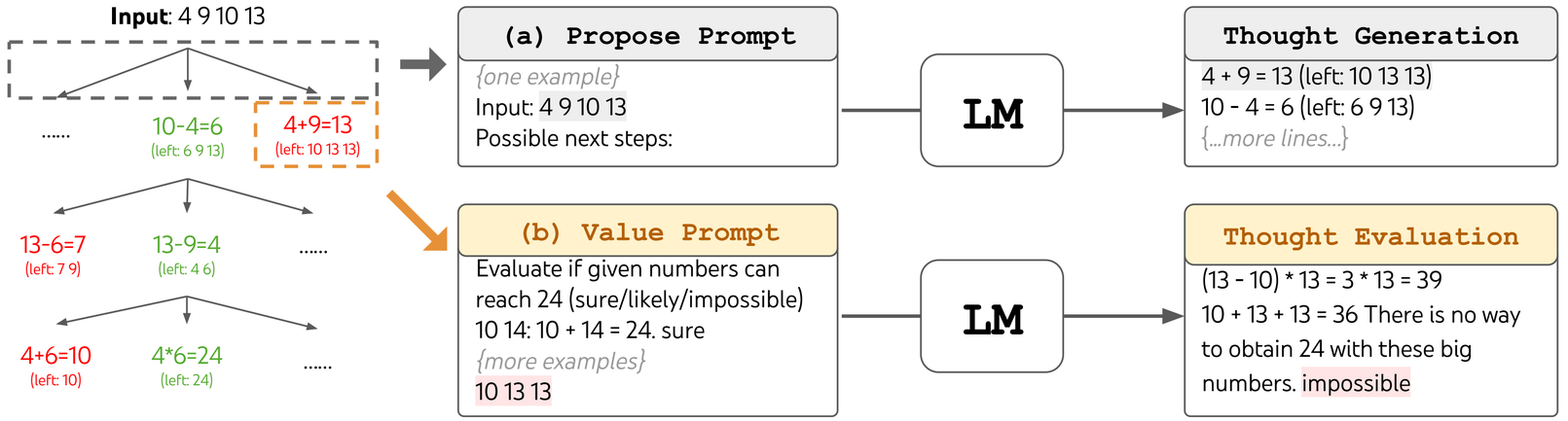


关于Test-compute scaling的早期探索（o1之前）

- 让模型在回答前“打草稿”： Chain of Thought (Nye et al., 2021; Wei et al., 2022), STaR (Zelikman et al, 2022)

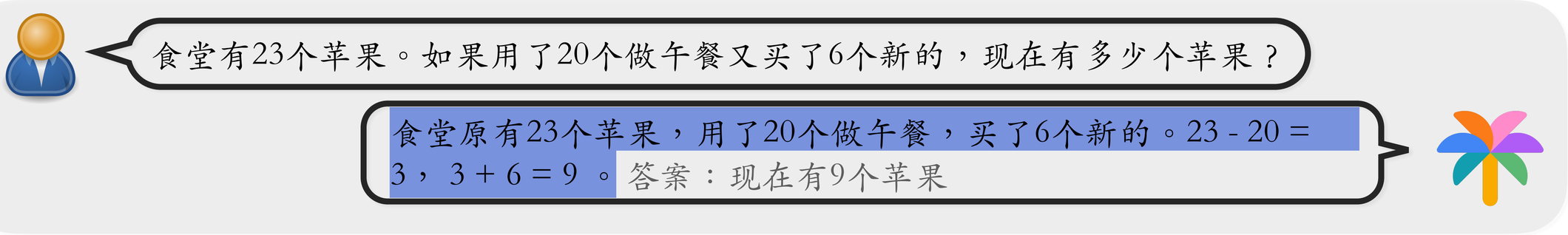


- 验证器+搜索: Tree of thoughts (Yao et al., 2023), Self-critic (2022+)

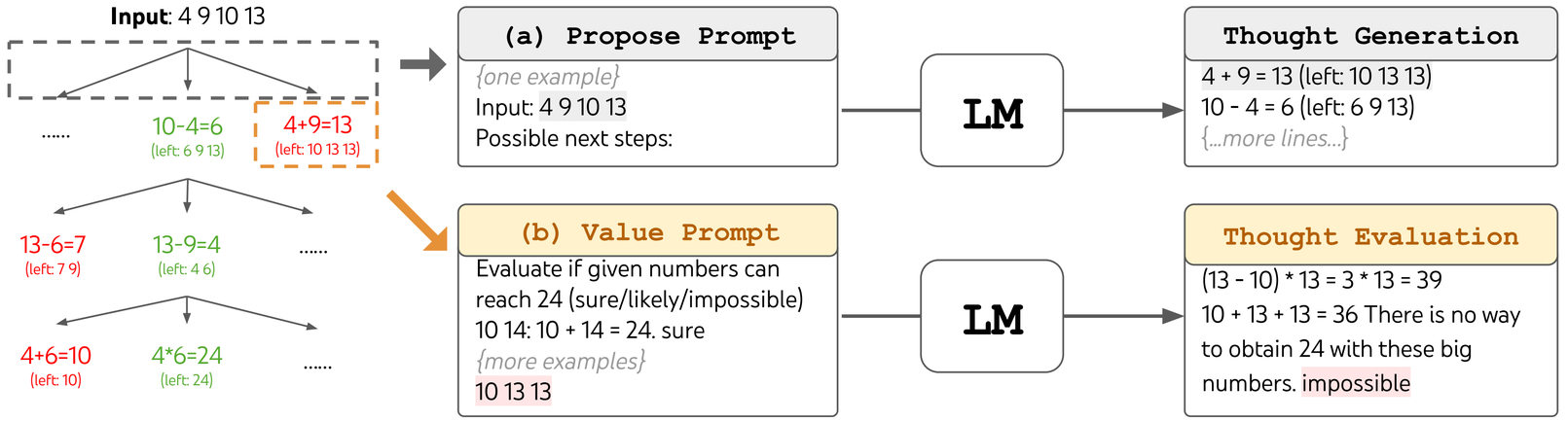


关于Test-compute scaling的早期探索（o1之前）

- 让模型在回答前“打草稿”： Chain of Thought (Nye et al., 2021; Wei et al., 2022), STaR (Zelikman et al, 2022)



- 验证器+搜索: Tree of thoughts (Yao et al., 2023), Self-critic (2022+)



- 过程监督（process supervision）：PRM800K (Lightman et al., 2023)

OpenAI o1-preview

- 2024年9月12号，OpenAI发布了o1-preview

September 12, 2024 Product

Introducing OpenAI o1-preview

A new series of reasoning models for solving hard problems. Available now.

OpenAI o1-preview

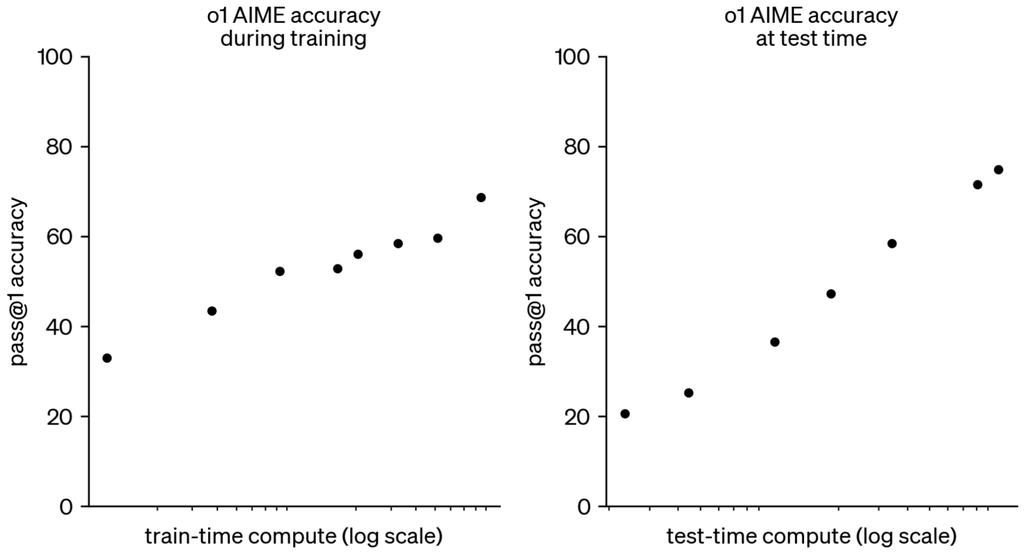
- 2024年9月12号，OpenAI发布了o1-preview

September 12, 2024 Product

Introducing OpenAI o1-preview

A new series of reasoning models for solving hard problems. Available now.

- o1-preview让人们开始关注test-compute scaling这个概念



OpenAI o1-preview

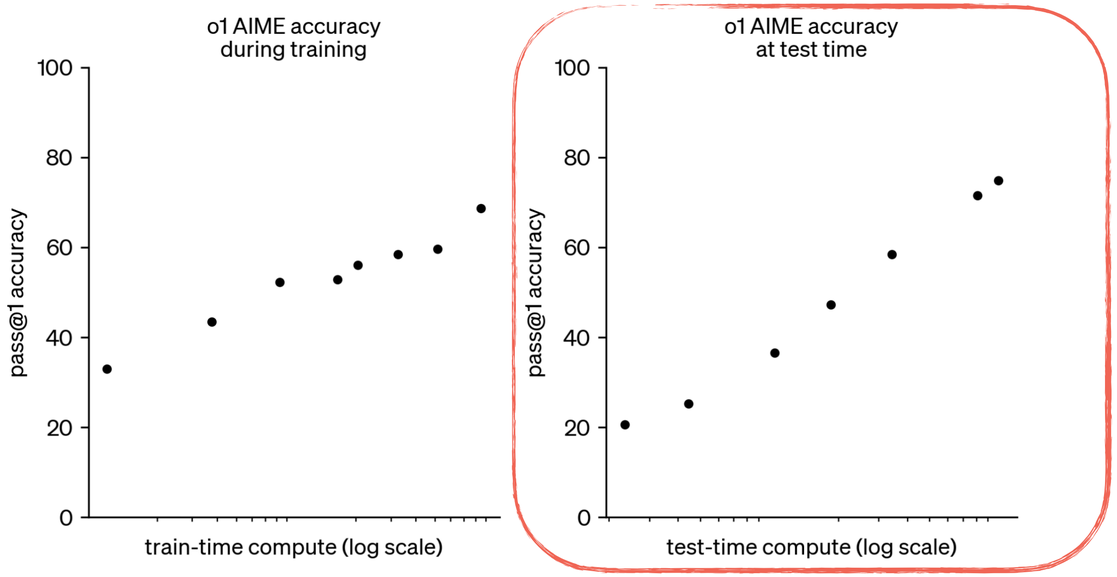
- 2024年9月12号，OpenAI发布了o1-preview

September 12, 2024 Product

Introducing OpenAI o1-preview

A new series of reasoning models for solving hard problems. Available now.

- o1-preview让人们开始关注test-compute scaling这个概念



- 对于复杂的问题，模型通过思考更长的时间来达到更好的效果。
- 类似于认知心理学里“快思考，慢思考”的概念。

为何test-compute scaling的关注度如此之高？

- 传统的data-scaling走到了尽头。Ilya Sutskever:“ We have but one internet”

Pre-training as we know it will end

Compute is growing:

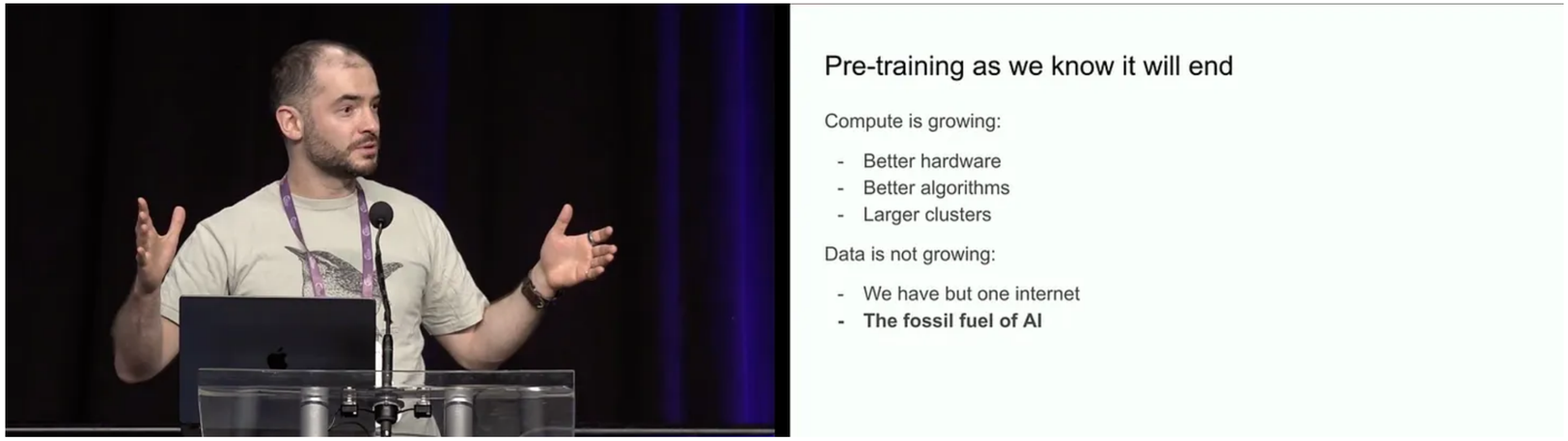
- Better hardware
- Better algorithms
- Larger clusters

Data is not growing:

- We have but one internet
- **The fossil fuel of AI**

为何test-compute scaling的关注度如此之高？

- 传统的data-scaling走到了尽头。Ilya Sutskever:“ We have but one internet”



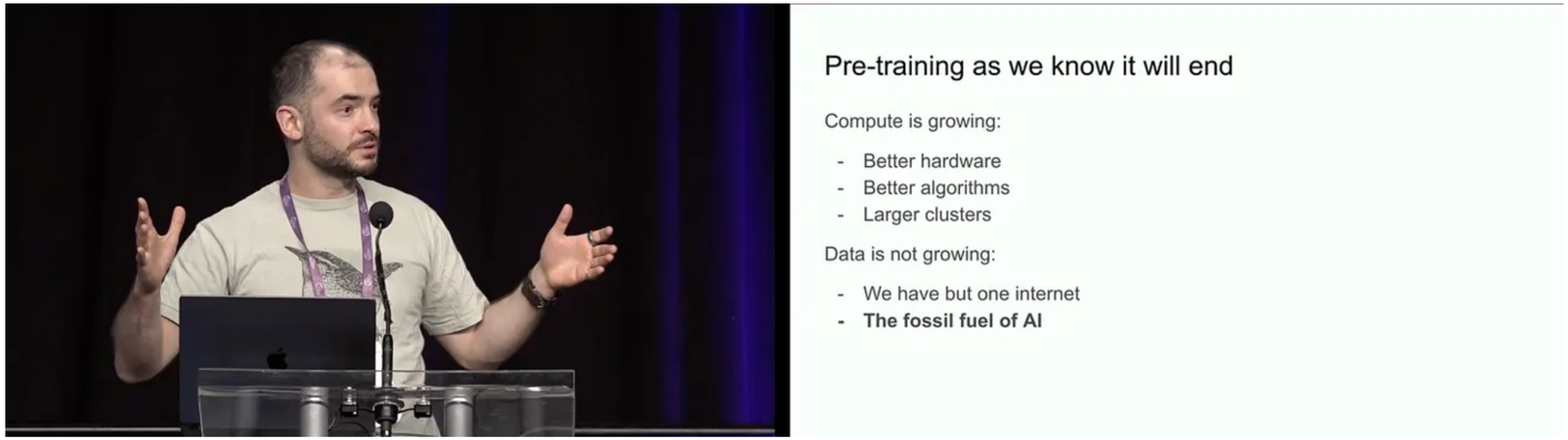
- o1模型在特定benchmark中产生了非常夸张的提升

On the 2024 AIME exams, GPT-4o only solved on average **12%** (1.8/15) of problems. o1 averaged **74%** (11.1/15) with a single sample per problem…

与base model相比，o1在数学竞赛上提升了62% （MMLU上只提升了2.8%）。

为何test-compute scaling的关注度如此之高？

- 传统的data-scaling走到了尽头。Ilya Sutskever:“ We have but one internet”



- o1模型在特定benchmark中产生了非常夸张的提升

On the 2024 AIME exams, GPT-4o only solved on average **12%** (1.8/15) of problems. o1 averaged **74%** (11.1/15) with a single sample per problem…

与base model相比，o1在数学竞赛上提升了62% （MMLU上只提升了2.8%）。

- o1的思考过程非常有趣：自我纠错，试错检验，等等。很像人类的思路。

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

Our **large-scale reinforcement learning** algorithm teaches the model how to think productively using its chain of thought in a highly **data-efficient** training process.

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

Our **large-scale reinforcement learning** algorithm teaches the model how to think productively using its chain of thought in a highly **data-efficient** training process.

OpenAI的官方介绍讲o1使用了大规模强化学习方法，非常高效的利用了数据。但到底多少计算资源才算是大规模？用多少数据才算是高效？

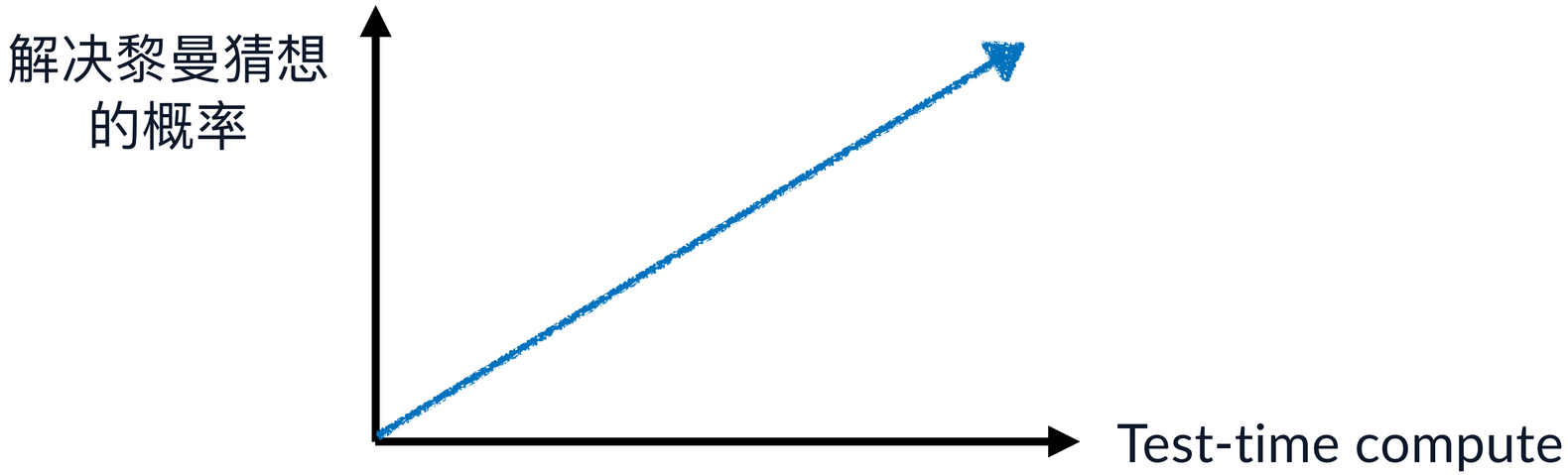
从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

Our **large-scale reinforcement learning** algorithm teaches the model how to think productively using its chain of thought in a highly **data-efficient** training process.

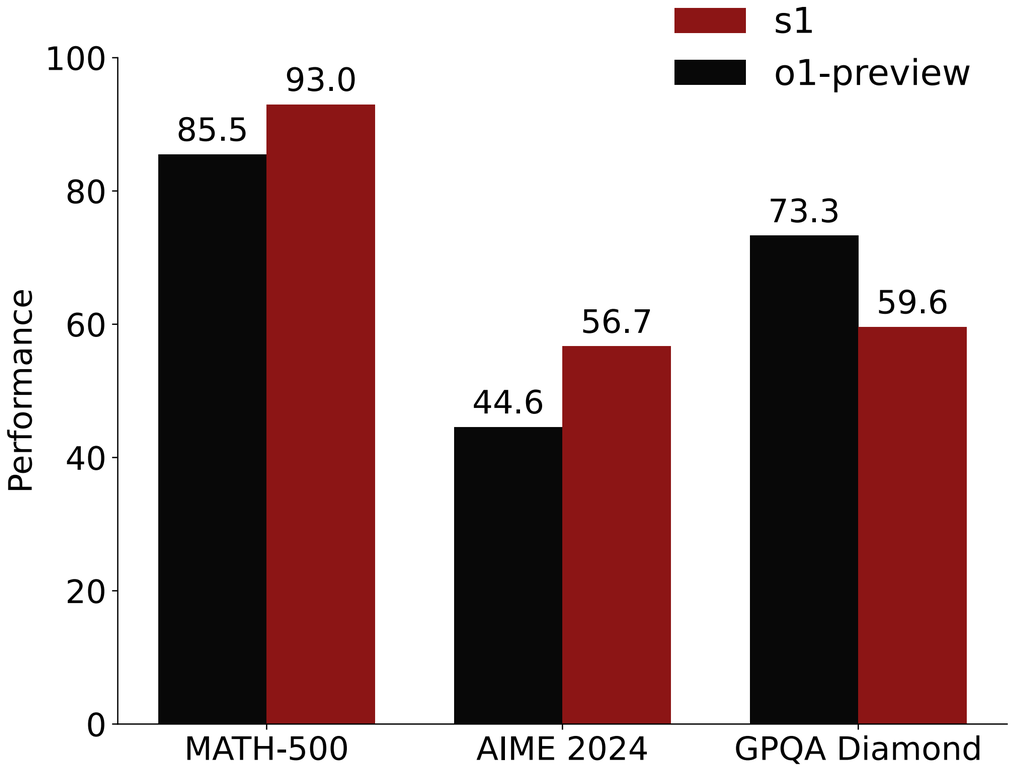
OpenAI的官方介绍讲o1使用了大规模强化学习方法，非常高效的利用了数据。但到底多少计算资源才算是大规模？用多少数据才算是高效？

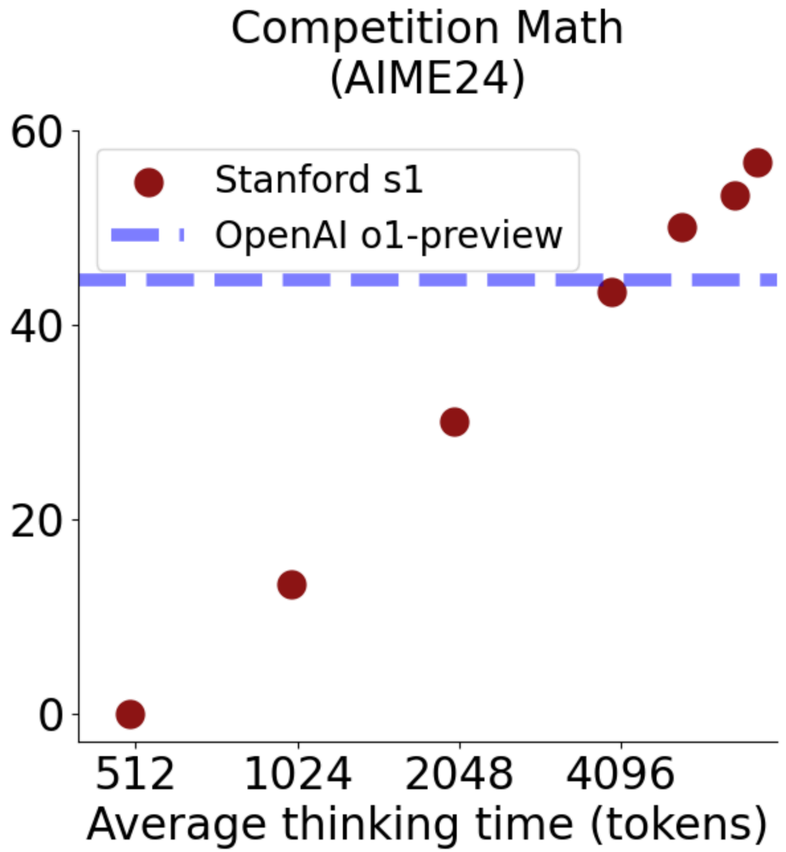
- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？



s1模型

在1000个高质量思考样本上进行监督微调就可以得到o1-preivew级别的能力。





s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

question	gemini_thoughts
..for martingale..	..use Doob’s..
..triangle ABC..	..AB is parallel..
……	……
..potential wall..	..engien-state..

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）

question	gemini_thoughts	is_qwen32b_correct	gemini_length	domain
..for martingale..	..use Doob’s..	No	8257	probability
..triangle ABC..	..AB is parallel..	Yes	4320	geometry
……	……	……	……	……
..potential wall..	..engien-state..	Yes	5697	physics

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。
2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）
3. 首先删除简单的问题。

question	gemini_thoughts	is_qwen32b_correct	gemini_length	domain
..for martingale..	..use Doob’s..	No	8257	probability
..triangle ABC..	..AB is parallel..	Yes	4320	geometry
……	……	……	……	……
..potential wall..	..engien-state..	Yes	5697	physics

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。
2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）
3. 首先删除简单的问题。

question	gemini_thoughts	is_qwen32b_correct	gemini_length	domain
..triangle ABC..	..AB is parallel..	Yes	4320	geometry
……	……	……	……	……
..potential wall..	..engien-state..	Yes	5697	physics

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。
2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）
3. 首先删除简单的问题。

- 之后会finetune Qwen2.5-32B-Instruct模型。如果一个题已经可以被Qwen解决，那么在这个题上训练的意义不如训练更难的题。

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）

3. 首先删除简单的问题。接着（a）先uniform sample一个domain，（b）再根据Gemini思路长度的power law来sample这个daomin下的一个问题。

- 之后会finetune Qwen2.5-32B-Instruct模型。如果一个题已经可以被Qwen解决，那么在这个题上训练的意义不如训练更难的题。

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）

3. 首先删除简单的问题。接着（a）先uniform sample一个domain，（b）再根据Gemini思路长度的power law来sample这个daomin下的一个问题。

- 之后会finetune Qwen2.5-32B-Instruct模型。如果一个题已经可以被Qwen解决，那么在这个题上训练的意义不如训练更难的题。
- 我们想用一千个问题涵盖尽可能广的领域，比如如果已经有了900个几何题，那么再采样一个几何题可能就不如选择其他领域。

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）

3. 首先删除简单的问题。接着（a）先uniform sample一个domain，（b）再根据Gemini思路长度的power law来sample这个daomin下的一个问题。

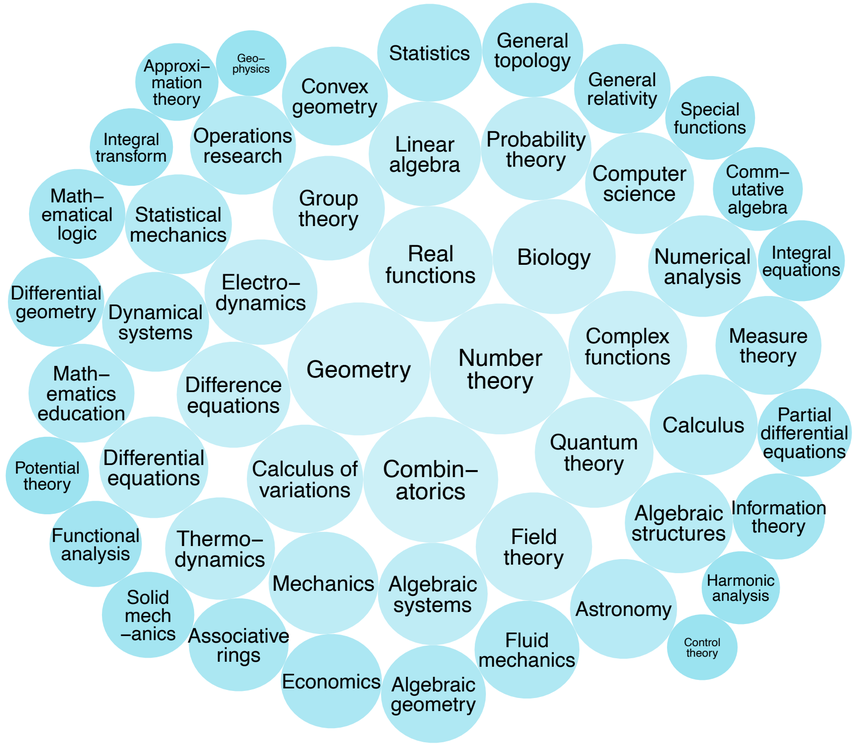
- 之后会finetune Qwen2.5-32B-Instruct模型。如果一个题已经可以被Qwen解决，那么在这个题上训练的意义不如训练更难的题。
- 我们想用一千个问题涵盖尽可能广的领域，比如如果已经有了900个几何题，那么再采样一个几何题可能就不如选择其他领域。
- 思考过程越长的问题往往越复杂，涵盖了更复杂的解题思路，所以更有教育意义。

s1K 数据集的设计思路

1. 收集大量的数学、科学、编程题目，并用Gemini Thinking生成思考过程。

2. 设计关于每个问题的“ feature”：问题的难度（Gemini思考的长度，非reason model是否能做对）以及问题的领域（几何，代数，等等）

3. 首先删除简单的问题。接着（a）先uniform sample一个domain，（b）再根据Gemini思路长度的power law来sample这个daomin下的一个问题。



- 之后会finetune Qwen2.5-32B-Instruct模型。如果一个题已经可以被Qwen解决，那么在这个题上训练的意义不如训练更难的题。
- 我们想用一千个问题涵盖尽可能广的领域，比如如果已经有了900个几何题，那么再采样一个几何题可能就不如选择其他领域。
- 思考过程越长的问题往往越复杂，涵盖了更复杂的解题思路，所以更有教育意义。

关于s1K的ablation studies（消融实验）

实验	定义	AIME 2024	MATH 500	GPQA Diamond
1K-random	随机采样一千个问题	36.7%	90.6%	52.0%
1K-diverse	随机采样一个领域	26.7%	91.2%	54.6%
1K-longest	最长的一千个问题	33.3%	90.4%	59.6%
59K-Full	所有的数据	53.3%	92.8%	58.1%
s1K	最终的s1K数据集	50.0%	93.0%	57.6%

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token
  “ 开始思考”和“ 结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...` `</thinking> Final answer:`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...` `</thinking> Final answer:`

- 强制思考至少1000个token。

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...` `</thinking> Final answer:`

- 强制思考至少1000个token。

不使用 BF `<thinking>` `...first 529 tokens...`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...` `</thinking> Final answer:`

- 强制思考至少1000个token。

不使用 BF `<thinking>` `...first 529 tokens...` `</thinking>...`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

- 强制思考至多100个token。

不使用 BF `<thinking>` `...first 100 tokens...` `the 101-th token thinking`

使用 BF `<thinking>` `...first 100 tokens...` `</thinking> Final answer:`

- 强制思考至少1000个token。

不使用 BF `<thinking>` `...first 529 tokens...` `</thinking>...`

With BF `<thinking>` `...first 529 tokens...`

Budget forcing （强制思考时间）

- s1微调的template: 增加两个special token “开始思考”和“结束思考”。

```
<user>
How many r in raspberry?
</user>
<assistant>
<thinking>
Let’s analyze the problem...
...
Therefore...
</thinking>
The final answer is…
</assistant>
```

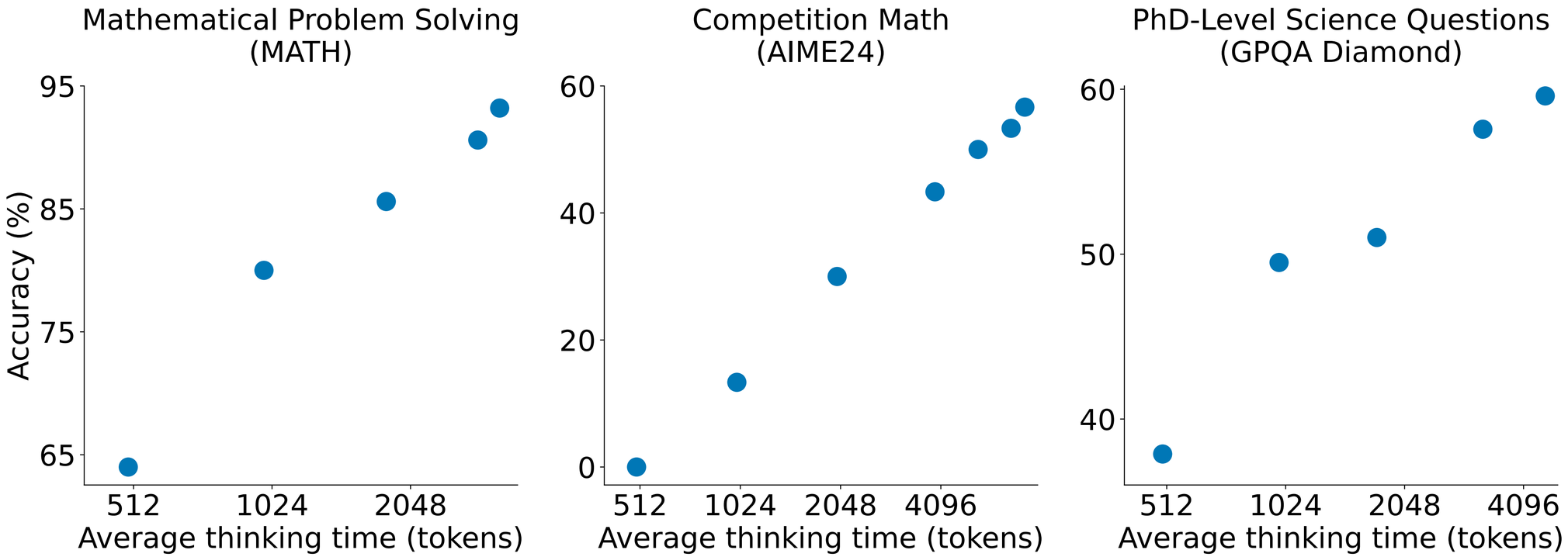
- 强制思考至多100个token。

不使用 BF	<thinking>	...first 100 tokens...	the 101-th token thinking
使用 BF	<thinking>	...first 100 tokens...	</thinking> Final answer:

- 强制思考至少1000个token。

不使用 BF	<thinking>	...first 529 tokens...	</thinking>...
With BF	<thinking>	...first 529 tokens...	Wait, ...continues...

在s1使用Budget forcing观察到了test-compute scaling



思考长度的泛化: 在AIME24上，s1正常decoding可以达到50%，而使用budget forcing后迫使模型思考更长时间后，准确率可以达到57%。

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

```
100%|IIIIIIIIIIIIIIIIIIII| 315/315 [26:46<00:00,  5.10s/it]
```

答案: 26 mins on 16 H100 （建立在已经收集好合适数据的前提上）.

- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

```
100%|IIIIIIIIIIIIIIIIIIII| 315/315 [26:46<00:00,  5.10s/it]
```

答案: 26 mins on 16 H100 （建立在已经收集好合适数据的前提上）.

启发: 类似o1的reasoning能力已经存在于一个足够好的pretraining模型之中了。

- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？

从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

```
100%|IIIIIIIIIIIIIIIIIIII| 315/315 [26:46<00:00,  5.10s/it]
```

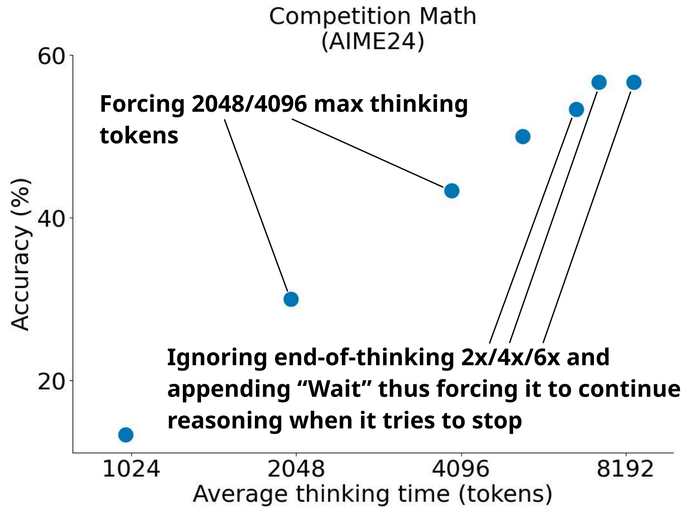
答案: 26 mins on 16 H100 （建立在已经收集好合适数据的前提上）.

启发: 类似o1的reasoning能力已经存在于一个足够好的pretraining模型之中了。

- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？

答案：没有定论。

  - 如果过度强制模型思考越来越长的时间，performance最终会减弱。
  - 这至少说明类似BF的方法不能无止境的通过scale test-time compute来提升能力。



从技术的角度来讲，o1引出了哪些有趣的科学问题？

- 训练一个具有类似o1思考能力的模型需要多少数据与计算资源？

```
100%|IIIIIIIIIIIIIIIIIIII| 315/315 [26:46<00:00,  5.10s/it]
```

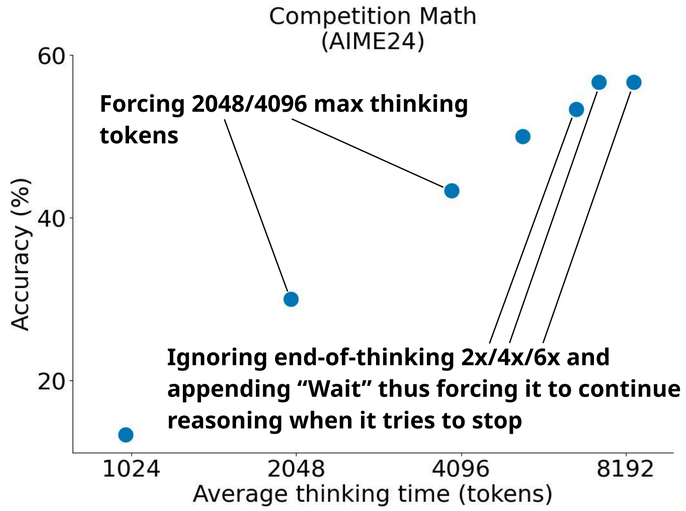
答案: 26 mins on 16 H100 （建立在已经收集好合适数据的前提上）.

启发: 类似o1的reasoning能力已经存在于一个足够好的pretraining模型之中了。

- 我们能无止境的扩大test-time compute的规模来解决越来越难的问题吗？

答案：没有定论。

- 如果过度强制模型思考越来越长的时间，performance最终会减弱。
- 这至少说明类似BF的方法不能无止境的通过scale test-time compute来提升能力。



启发: 要谨慎理解test-compute scaling相关的结论。对于传统的pretraining data scaling，更多的数据总可以提升模型的能力，test-comute scaling可能不同。