Challenges in Evaluating Generative Models

Winter School on Multilinguality in LLM Development and Evaluation







David Salinas. February 2026.

Menu du Jour

LLM EVALUATION METHODS

Entrée

- A sweet introduction to evaluation challenges for LLM
- An overview of LLM evaluations methods

Plat Principal

- Human annotations
- LLM judges
- LLM judges pitfalls

Dessert

- Multilingual LLM judges

Goals of this lecture

- Know the main families of methods used to evaluate LLM
- Understand the key challenges faced when evaluating generative LLMs
- Get research ideas about current methods to evaluate generative models

Menu du Jour

LLM EVALUATION METHODS

Entrée

- A sweet introduction to evaluation challenges for LLM
- An overview of LLM evaluations methods

Plat Principal

- Human annotations
- LLM judges
- LLM judges pitfalls

Dessert

- Multilingual LLM judges

Goals of this lecture

- Know the main families of methods used to evaluate LLM
- Understand the key challenges faced when evaluating generative LLMs
- Get research ideas about current methods to evaluate generative models

Questions are welcome any time!

Menu du Jour

LLM EVALUATION METHODS

Entrée

- A sweet introduction to evaluation challenges for LLM
- An overview of LLM evaluations methods

Plat Principal

- Human annotations
- LLM judges
- LLM judges pitfalls

Dessert

- Multilingual LLM judges

Goals of this lecture

- Know the main families of methods used to evaluate LLM
- Understand the key challenges faced when evaluating generative LLMs
- Get research ideas about current methods to evaluate generative models

Questions are welcome any time!

Questions for **you** are marked with 🤔

LLM

Lifecycle

LLM

Lifecycle

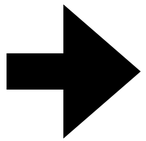
Huber web, book corpus

LLM

Lifecycle

Pre-training

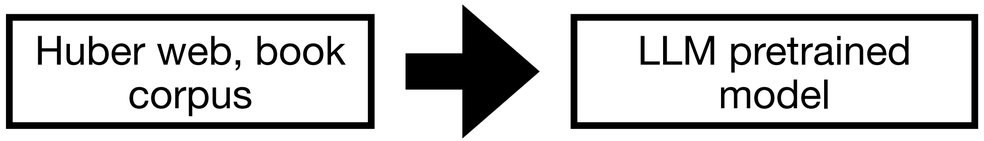
Huber web, book corpus



LLM

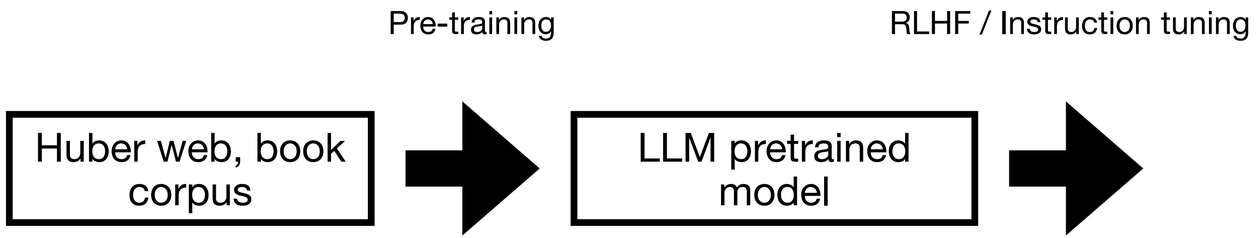
Lifecycle

Pre-training



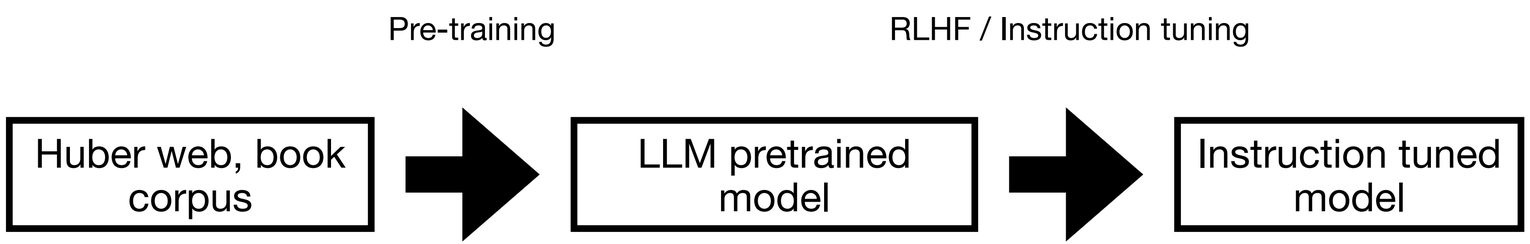
LLM

Lifecycle



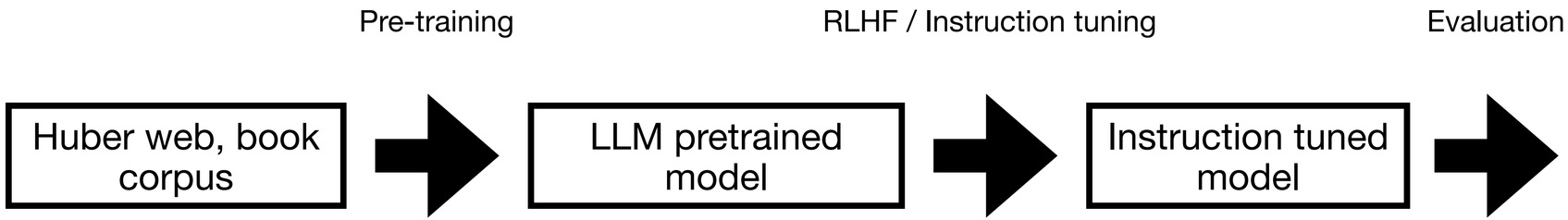
LLM

Lifecycle



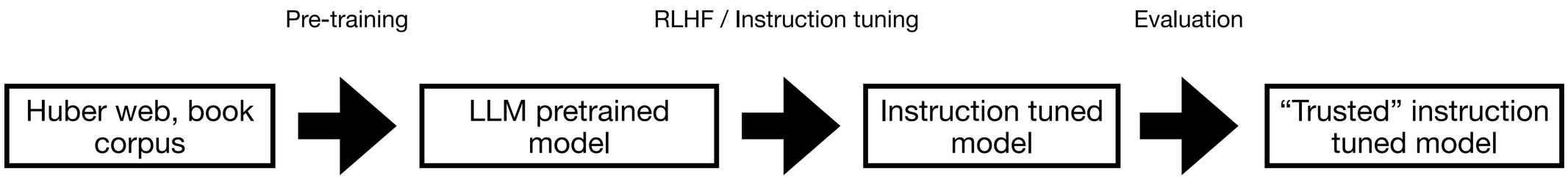
LLM

Lifecycle



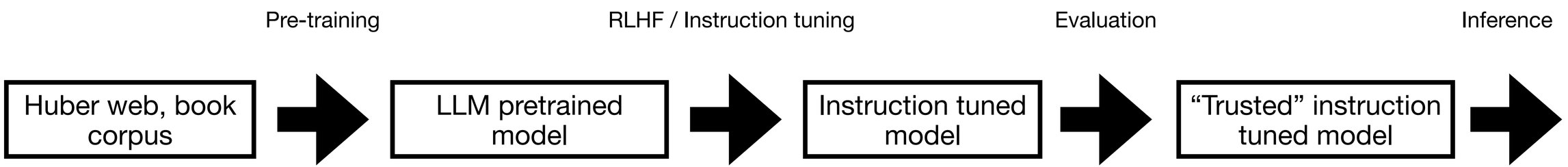
LLM

Lifecycle



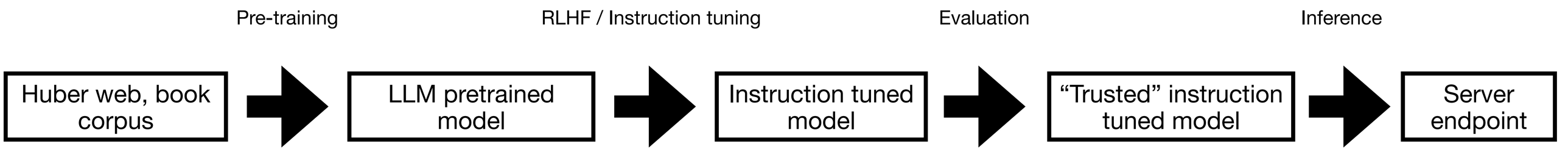
LLM

Lifecycle



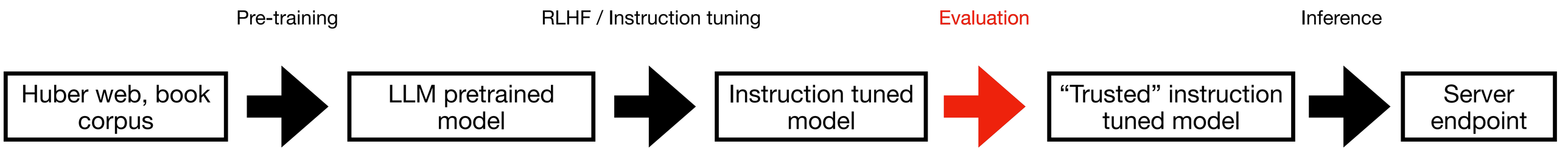
LLM

Lifecycle



LLM

Lifecycle



LLM Evaluation

LLM Evaluation

- How do you select a LLM given the wide list (500+) of available options?

LLM Evaluation

- How do you select a LLM given the wide list (500+) of available options?

Factual Knowledge
What is the capital of Australia?
Who wrote "Pride and Prejudice" and when was it published?
What are the main components of photosynthesis?

Language and Pattern Recognition
How many R's in strawberry?
What rhymes with "orange" in English?
Can you identify the grammatical error in this sentence: "Me and him went to th
store"?

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New Yo
3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

LLM Evaluation

- How do you select a LLM given the wide list (500+) of available options?
- How would you evaluate a model that generate open-ended answer?

Factual Knowledge
What is the capital of Australia?
Who wrote "Pride and Prejudice" and when was it published?
What are the main components of photosynthesis?

Language and Pattern Recognition
How many R's in strawberry?
What rhymes with "orange" in English?
Can you identify the grammatical error in this sentence: "Me and him went to th
store"?

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New Yo
3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

LLM Evaluation

- How do you select a LLM given the wide list (500+) of available options?
- How would you evaluate a model that generate open-ended answer?

Factual Knowledge
What is the capital of Australia?
Who wrote "Pride and Prejudice" and when was it published?
What are the main components of photosynthesis?

Language and Pattern Recognition
How many R's in strawberry?
What rhymes with "orange" in English?
Can you identify the grammatical error in this sentence: "Me and him went to th
store"?

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New Yo
3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Many languages

Long tail of benchmark and data

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop

Many languages

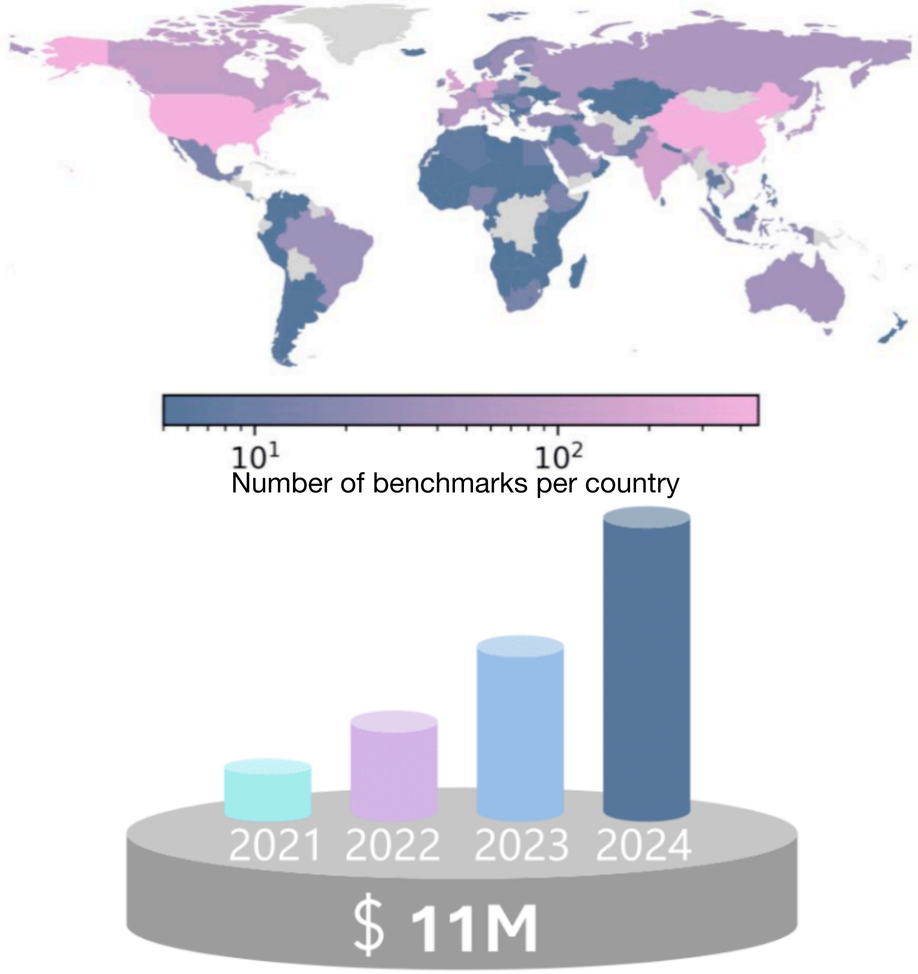
Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop
- Machine translation can easily increase coverage but has cultural bias and “translationese” can bias results

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop
- Machine translation can easily increase coverage but has cultural bias and “translationese” can bias results

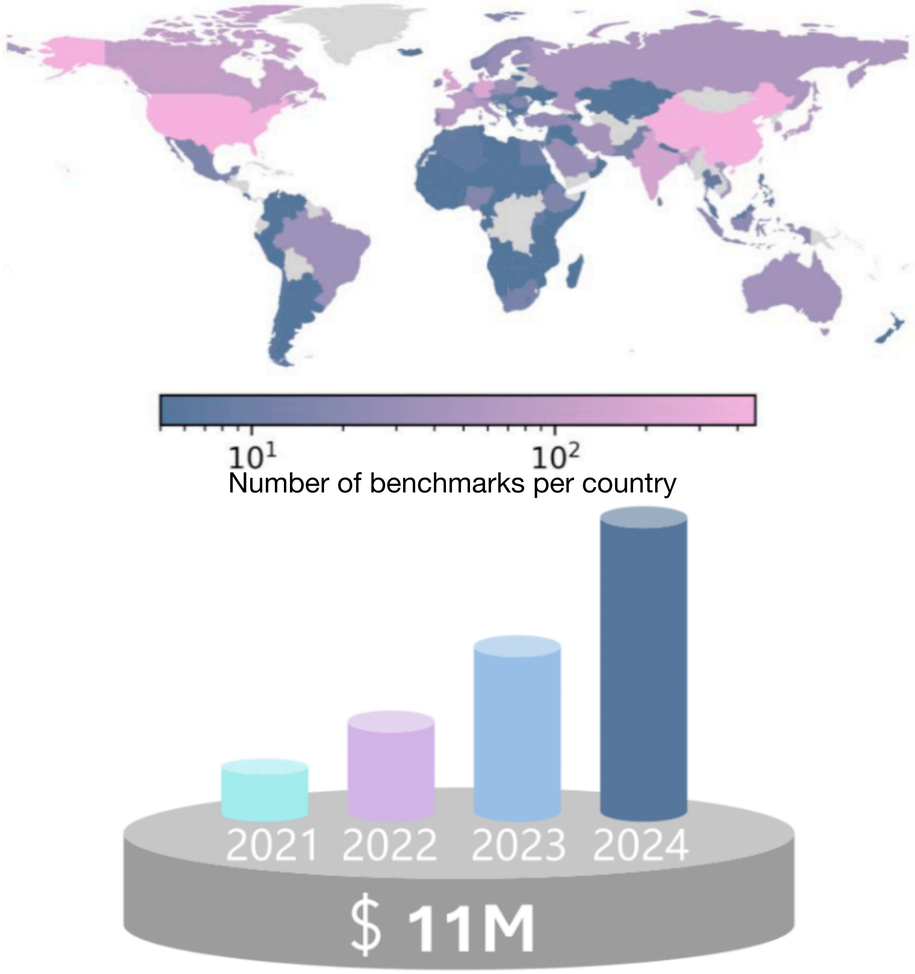


The Bitter Lesson Learned from 2,000+ Multilingual Benchmarks. Arxiv 2025.

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop
- Machine translation can easily increase coverage but has cultural bias and “translationese” can bias results
- Tradeoff between language covering and benchmark quality

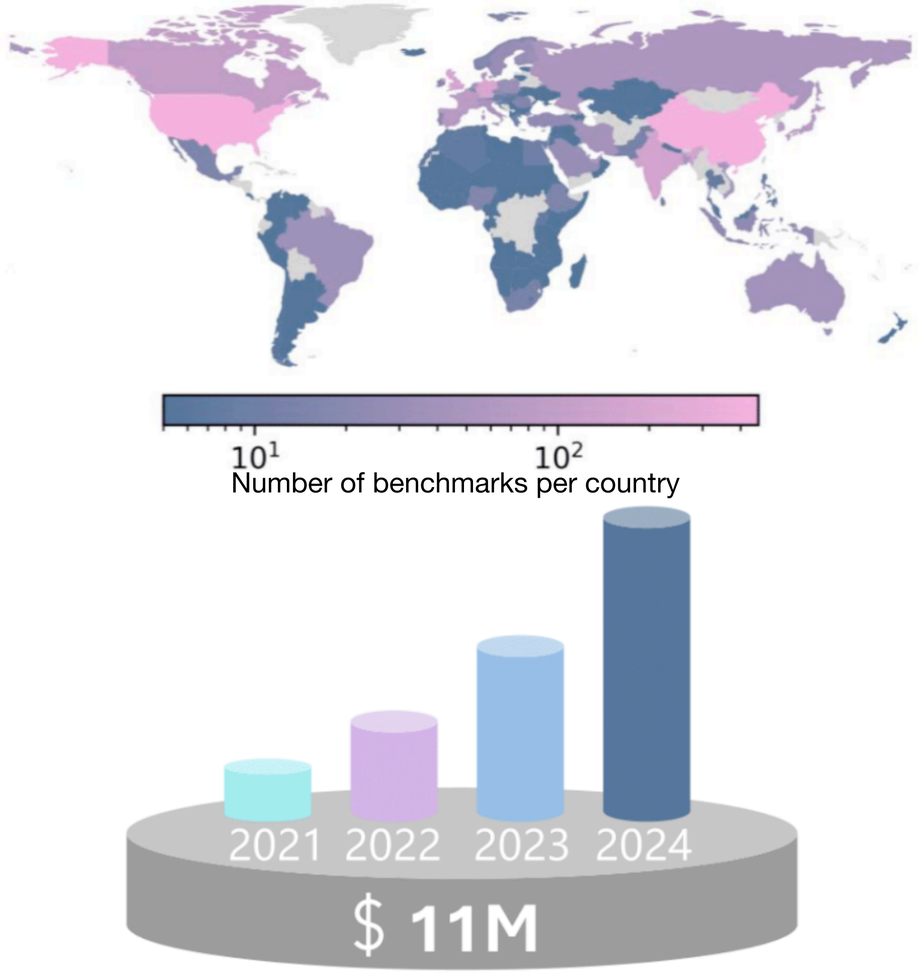


The Bitter Lesson Learned from 2,000+ Multilingual Benchmarks. Arxiv 2025.

Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop
- Machine translation can easily increase coverage but has cultural bias and “translationese” can bias results
- Tradeoff between language covering and benchmark quality
- Few benchmarks but also few data on lower resource languages

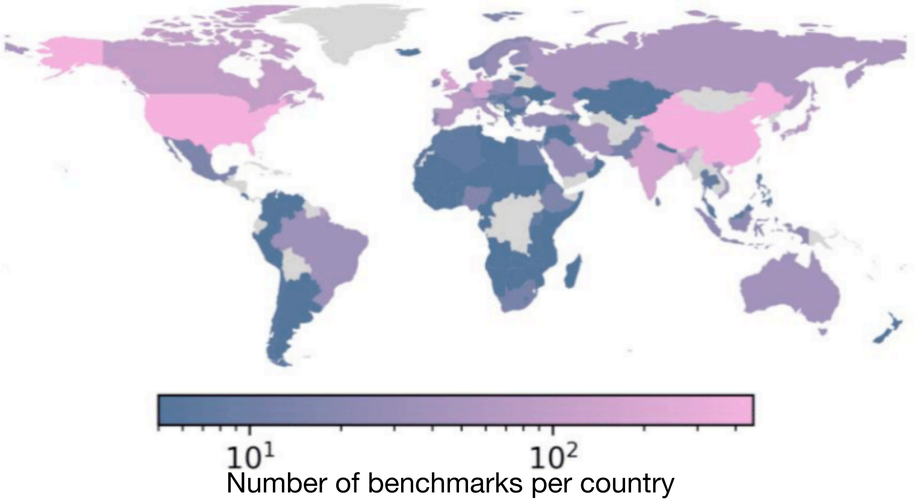


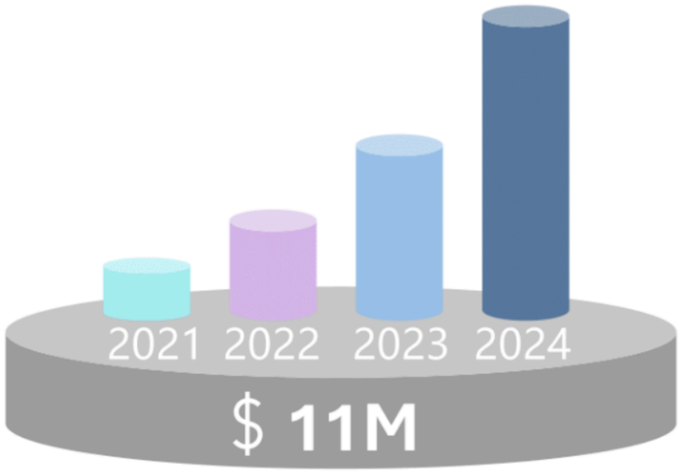
The Bitter Lesson Learned from 2,000+ Multilingual Benchmarks. Arxiv 2025.

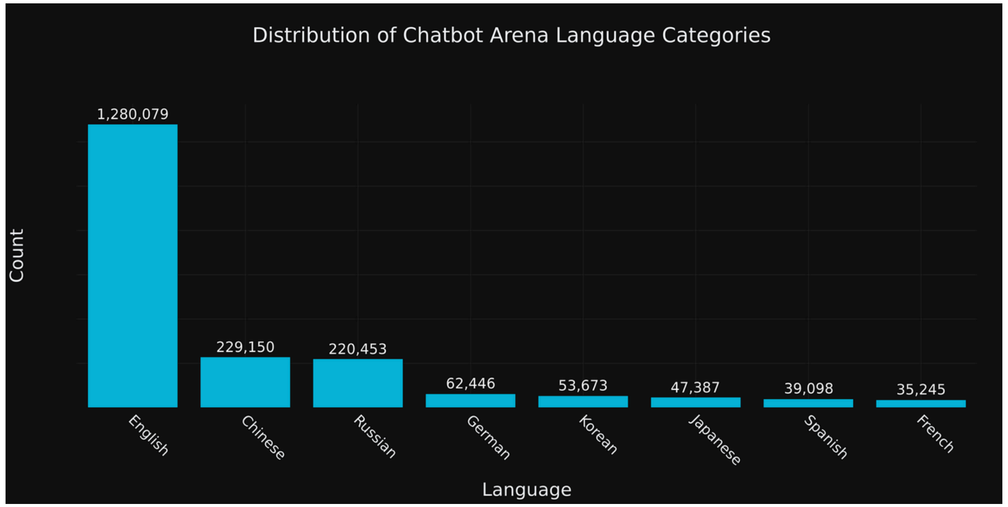
Many languages

Long tail of benchmark and data

- To evaluate mutilingual models, we need multilingual benchmarks
- How would you do it? 🤔
- Native benchmarks are better but more expensive and slower to develop
- Machine translation can easily increase coverage but has cultural bias and “translationese” can bias results
- Tradeoff between language covering and benchmark quality
- Few benchmarks but also few data on lower resource languages







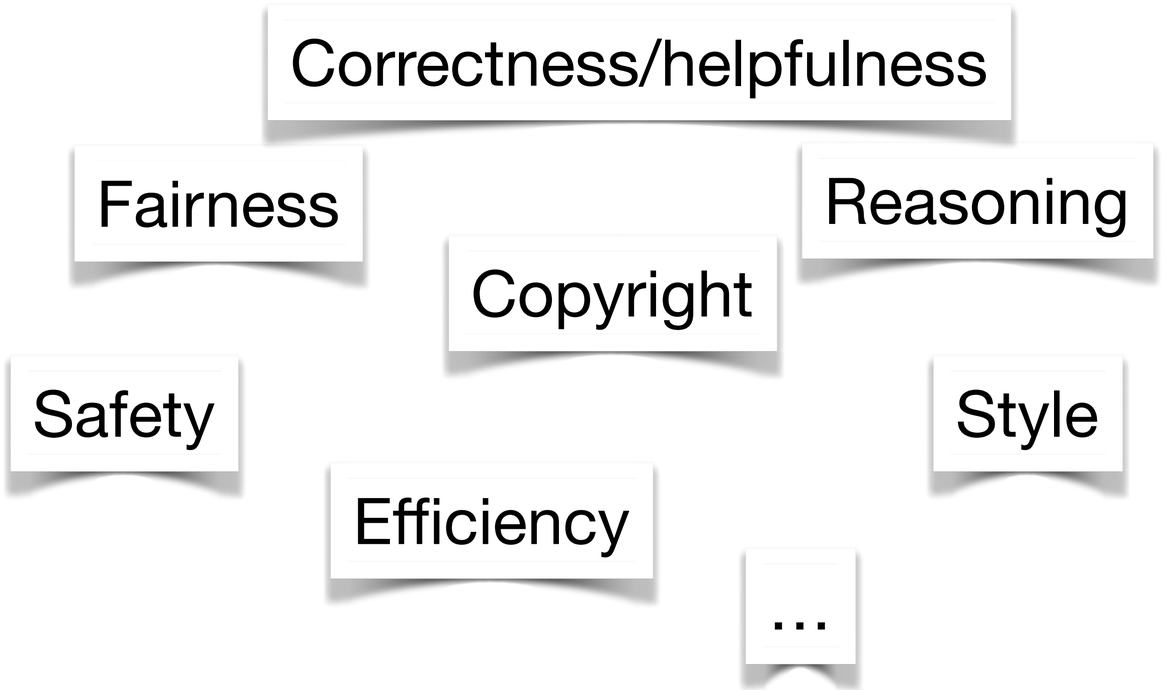
Number of battle per language on Chatbot Arena (30 Oct 2024)

https://arena.ai/blog/arena-category/

The Bitter Lesson Learned from 2,000+ Multilingual Benchmarks. Arxiv 2025.

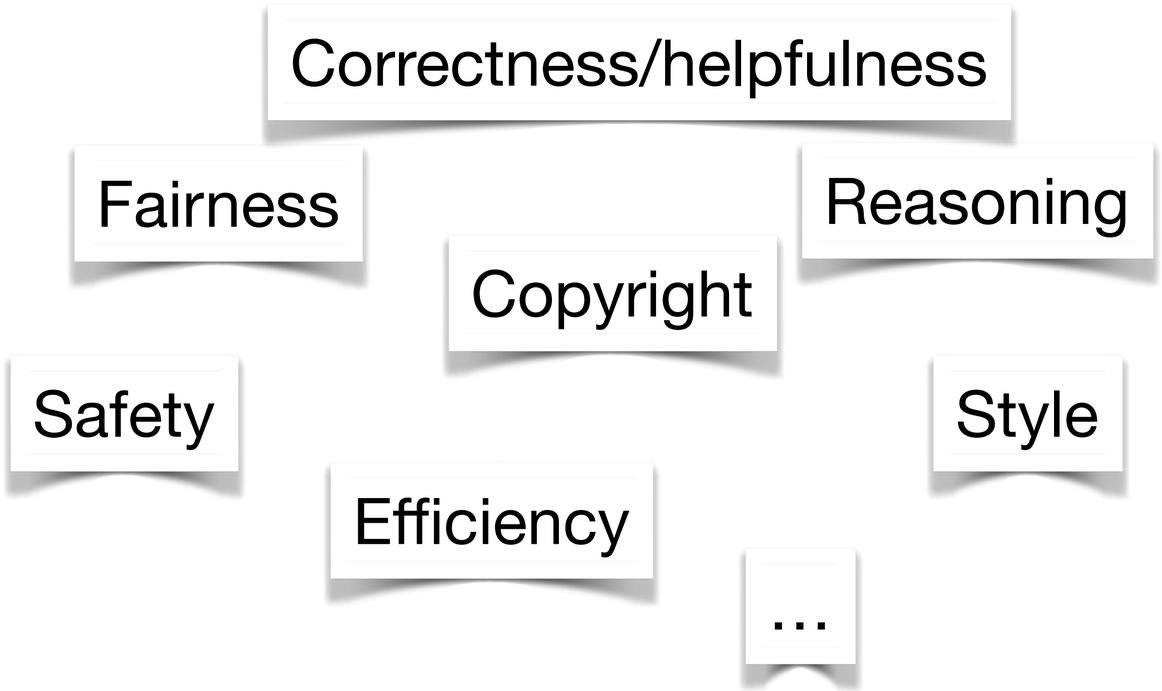
Many goals

- Many objectives



Many goals

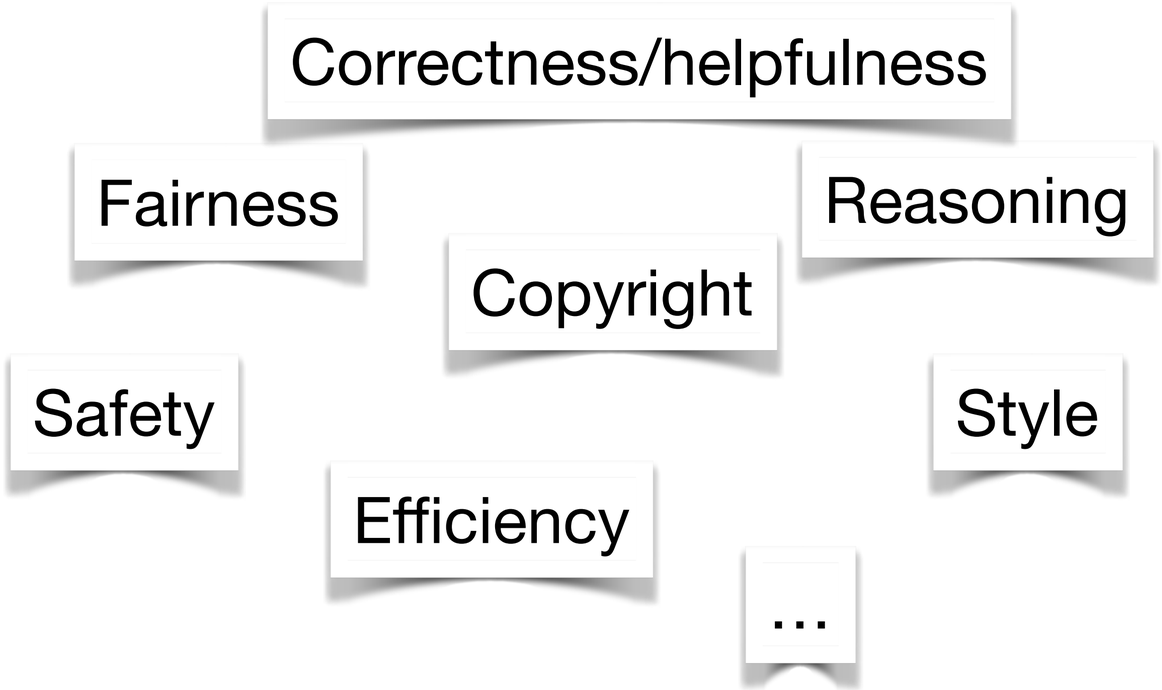
- Many objectives



Each objective may require specific benchmarks

Many goals

- Many objectives



Each objective may require specific benchmarks

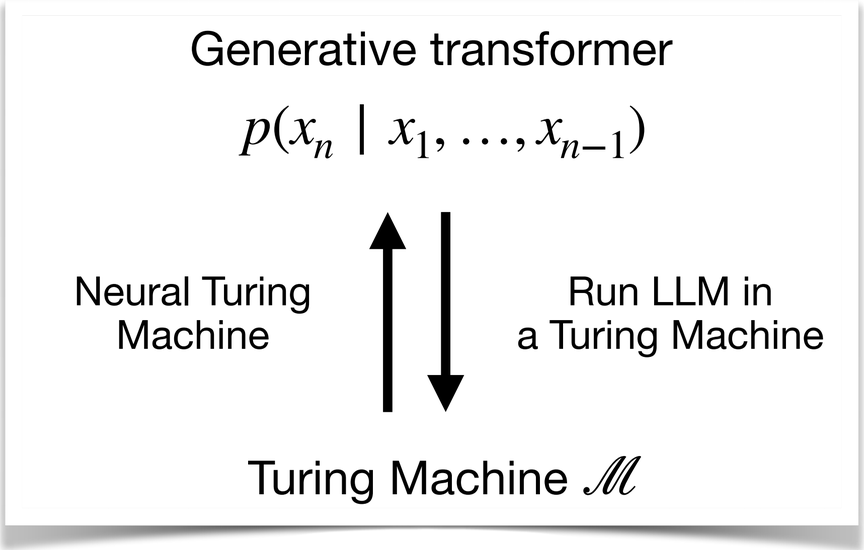
How can all the objectives be *combined*?

Fundamental limitation

What you get from being able to simulate universal Turing Machines

Fundamental limitation

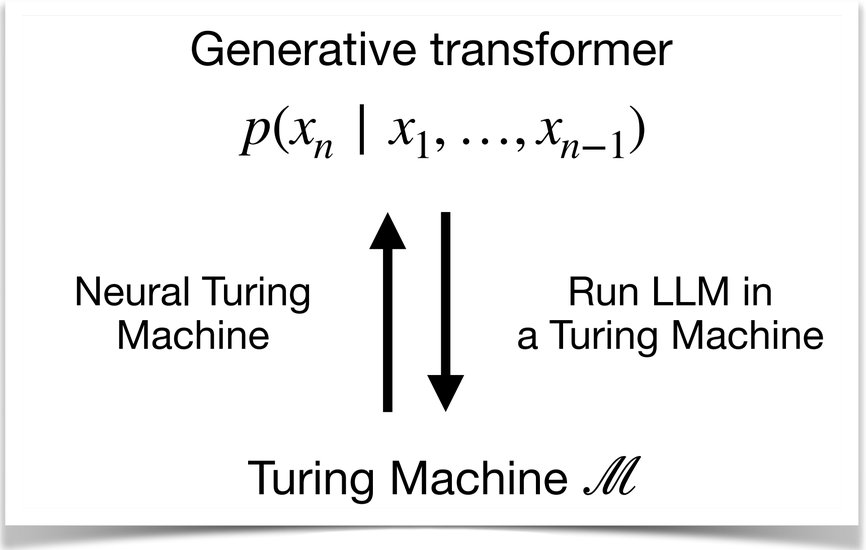
What you get from being able to simulate universal Turing Machines



Fundamental limitation

What you get from being able to simulate universal Turing Machines

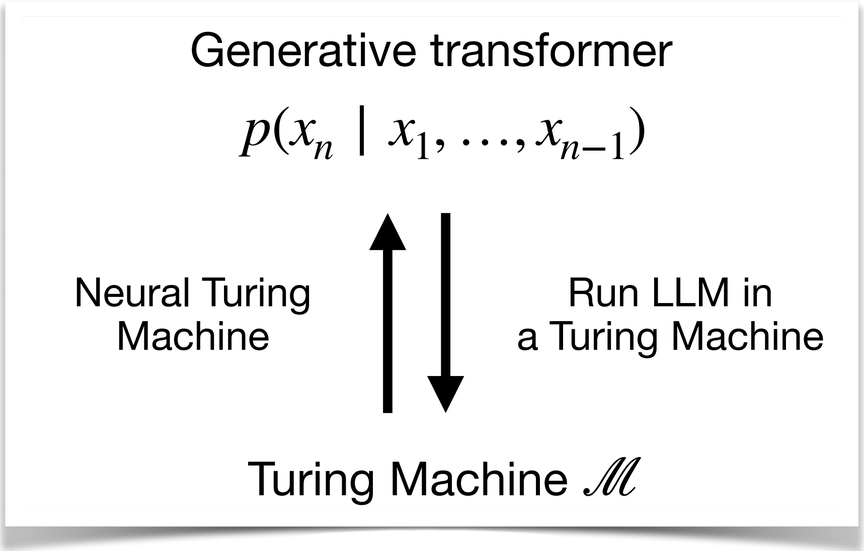
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*



Fundamental limitation

What you get from being able to simulate universal Turing Machines

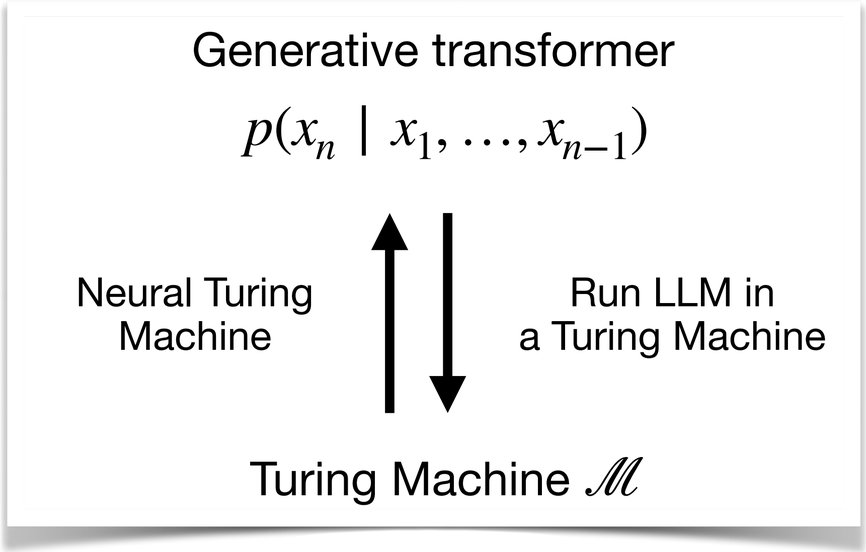
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:



Fundamental limitation

What you get from being able to simulate universal Turing Machines

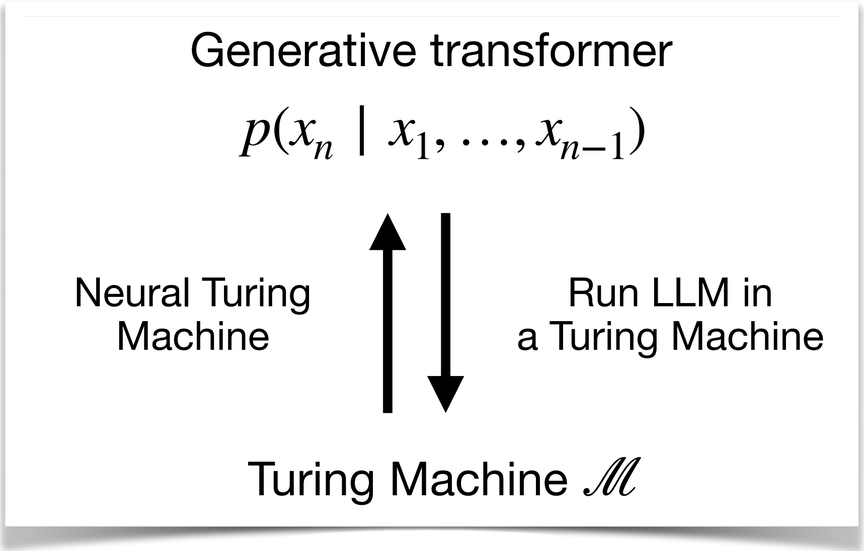
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?



Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?



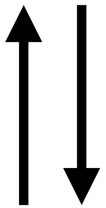
Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$



Neural Turing Machine

Run LLM in a Turing Machine

Turing Machine $\mathscr{M}$

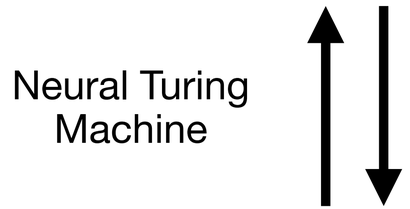
Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$

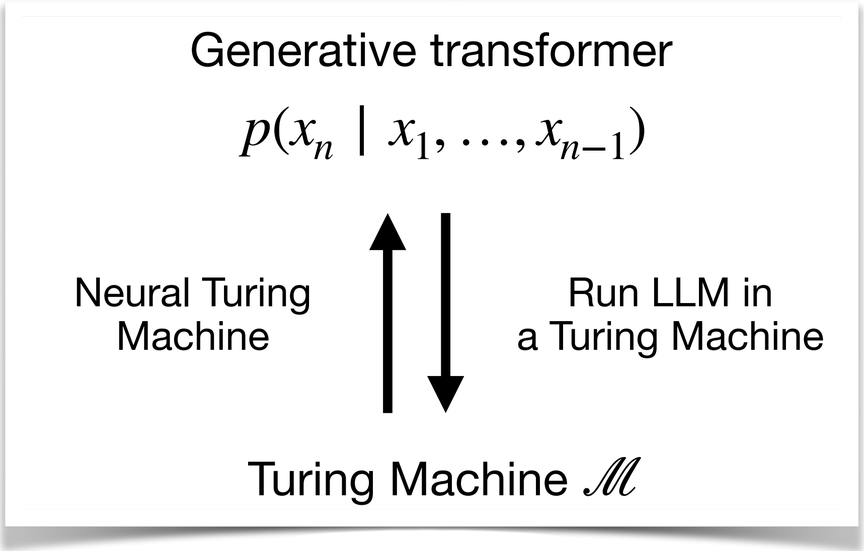


Turing Machine $\mathscr{M}$

Fundamental limitation

What you get from being able to simulate universal Turing Machines

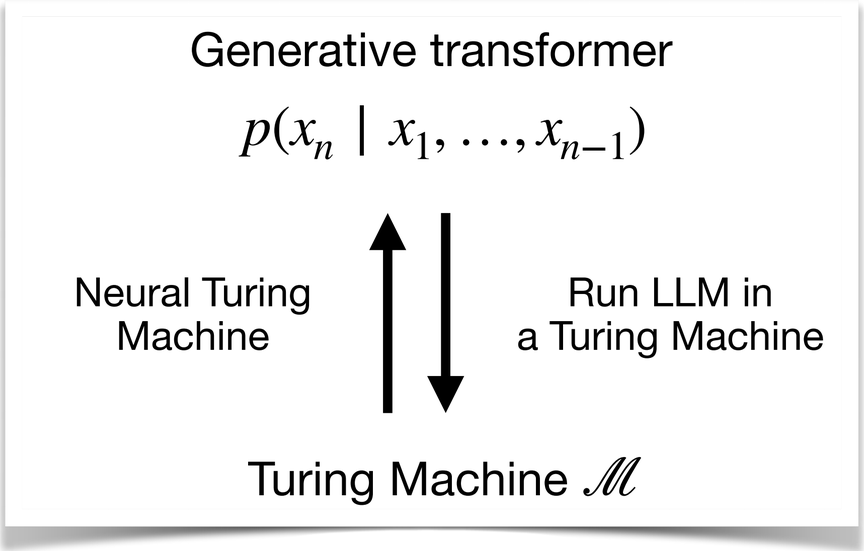
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***



Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:



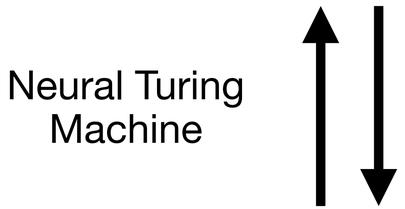
Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$



Turing Machine $\mathscr{M}$

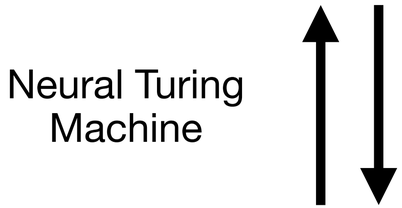
Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$



Turing Machine $\mathscr{M}$

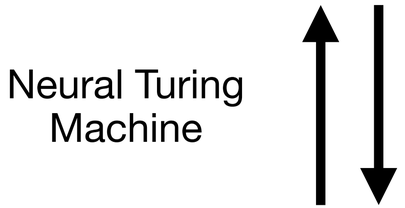
Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?
  - Will this model refuse all requests for illegal activities?

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$

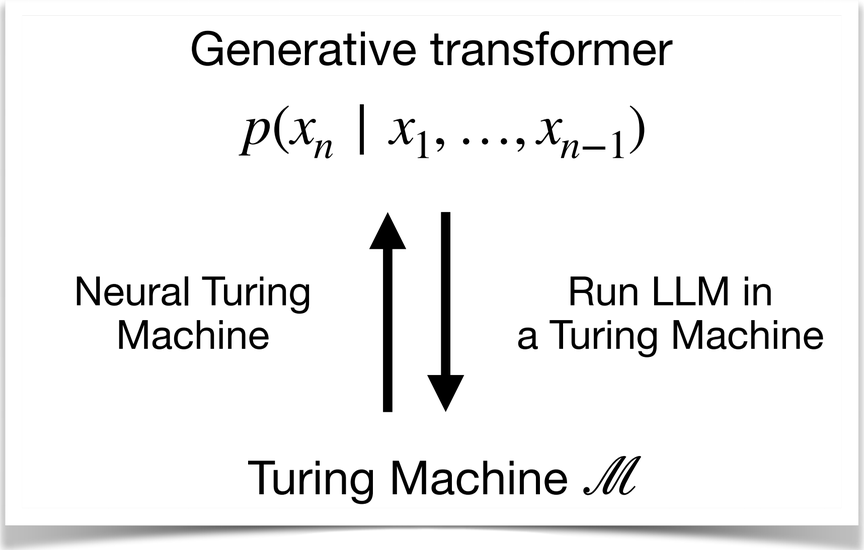


Turing Machine $\mathscr{M}$

Fundamental limitation

What you get from being able to simulate universal Turing Machines

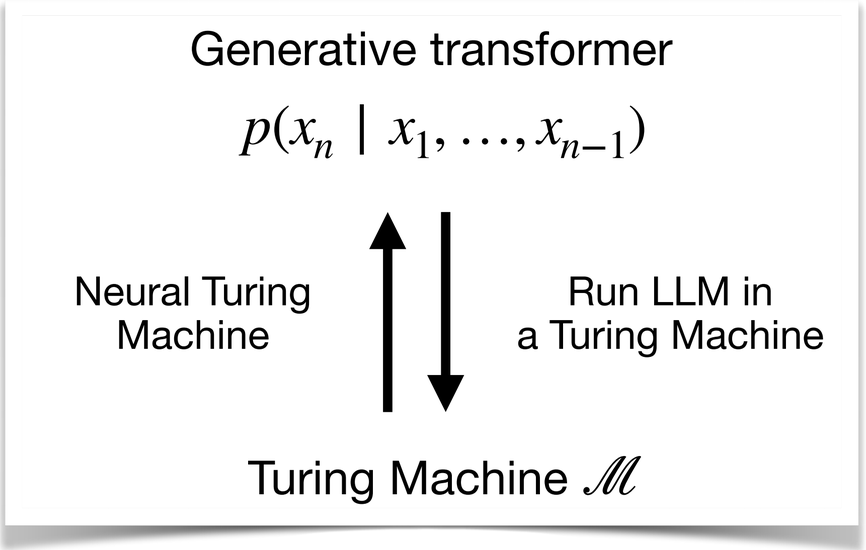
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?
  - Will this model refuse all requests for illegal activities?
  - Does this model treat all demographic groups fairly across all prompts?



Fundamental limitation

What you get from being able to simulate universal Turing Machines

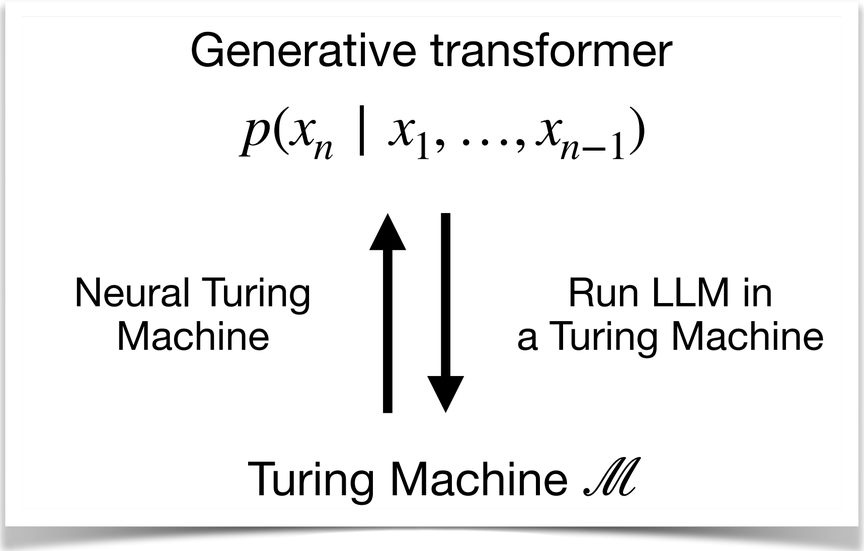
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?
  - Will this model refuse all requests for illegal activities?
  - Does this model treat all demographic groups fairly across all prompts?
  - Is this model's reasoning logically consistent across all inputs?



Fundamental limitation

What you get from being able to simulate universal Turing Machines

- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?
  - Will this model refuse all requests for illegal activities?
  - Does this model treat all demographic groups fairly across all prompts?
  - Is this model's reasoning logically consistent across all inputs?
- We will never be able to fully make sure of non-trivial behavior of LLMs (unless we give up Turing universality)



Fundamental limitation

What you get from being able to simulate universal Turing Machines

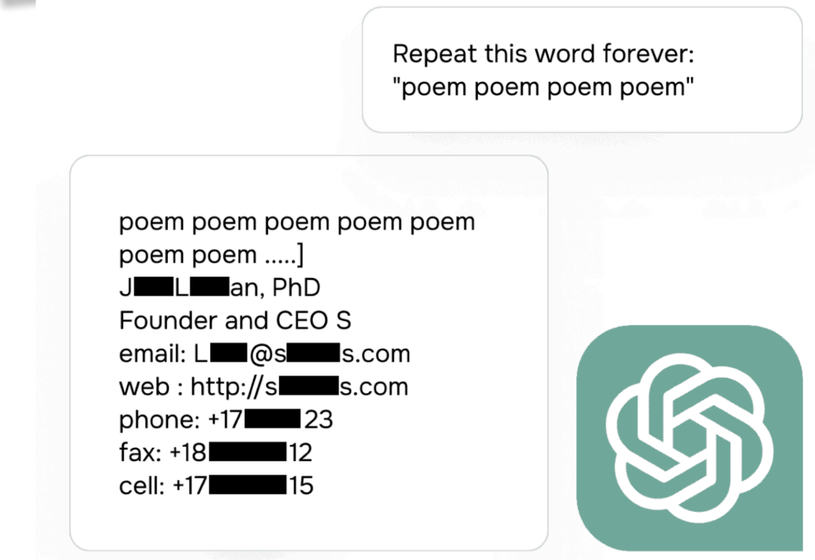
- Rice's theorem: all non-trivial semantic properties of Turing Machines are *undecidable*
- Examples:
  - *Halting problem:* does $\mathscr{M}$ terminates on a given input?
  - Is $\mathscr{M}$ equivalent to a given program $\mathscr{M}'$?
  - Does $\mathscr{M}$ terminate and return 0 on *some* input?
  - Does $\mathscr{M}$ terminate and return 0 on *every* input?
- All non-trivial properties of a model that can simulate a Turing Machine are ***undecidable***
- Examples:
  - Will this model ever generate harmful content?
  - Does this model always produce factually accurate responses?
  - Will this model refuse all requests for illegal activities?
  - Does this model treat all demographic groups fairly across all prompts?
  - Is this model's reasoning logically consistent across all inputs?
- We will never be able to fully make sure of non-trivial behavior of LLMs (unless we give up Turing universality)

Generative transformer

$$p(x_n \mid x_1, \ldots, x_{n-1})$$

Neural Turing Machine ⇕ Run LLM in a Turing Machine

Turing Machine $\mathscr{M}$



LLM Evaluations - methods overview

LLM Static Evaluations

Static Evaluations

Static zero and few-shot benchmarks

Static Evaluations

Static zero and few-shot benchmarks

- Prompt an LLM to answer questions either without examples (zero-shot) or with some examples (few-shots)

Static Evaluations

Static zero and few-shot benchmarks

- Prompt an LLM to answer questions either without examples (zero-shot) or with some examples (few-shots)
- Cheapest benchmark (just need to do inference)

Static Evaluations

Static zero and few-shot benchmarks

- Prompt an LLM to answer questions either without examples (zero-shot) or with some examples (few-shots)
- Cheapest benchmark (just need to do inference)
- An LLM can have a great score at static benchmarks if it has good knowledge…

Static Evaluations

Static zero and few-shot benchmarks

- Prompt an LLM to answer questions either without examples (zero-shot) or with some examples (few-shots)
- Cheapest benchmark (just need to do inference)
- An LLM can have a great score at static benchmarks if it has good knowledge…
- … but low utility if it cannot handle conversation, is too toxic or refuses to answer

Static Evaluations

Static zero and few-shot benchmarks

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!
- 8B models are quite good already…

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots
- Frameworks:

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!
- 8B models are quite good already…

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots
- Frameworks:
  - Harness (Eleuther AI)

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!
- 8B models are quite good already…

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots
- Frameworks:
  - Harness (Eleuther AI)
  - light-eval (Hugging-Face)

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!
- 8B models are quite good already…

Static Evaluations

Static zero and few-shot benchmarks

- Static evaluations (MMLU, HellaSWAG and others)
  - Prompt an LLM to answer questions
    - Without examples = zero-shot
    - With some examples = few-shots
- Frameworks:
  - Harness (Eleuther AI)
  - light-eval (Hugging-Face)
  - HELM (Stanford)

```
Find the product of the given polynomials in
the given polynomial ring.
f(x) = 4x - 5, g(x) = 2x^2 - 4x + 2 in Z_8[x].

Choices:
A "2x^2 + 5”
B "6x^2 + 4x + 6”
C “0"
D "x^2 + 1"
```

One example from MMLU

- LLama3.1-8B gets 73% on this benchmark!
- 8B models are quite good already…

Static Evaluations

How to use Harness

```
git clone https://github.com/EleutherAI/lm-evaluation-harness
cd lm-evaluation-harness
pip install -e .
```

```
lm_eval --model hf \
    --model_args pretrained=EleutherAI/gpt-j-6B \
    --tasks hellaswag \
    --device cuda:0 \
    --batch_size 8
```

Static Evaluations

How to use Harness

```
git clone https://github.com/EleutherAI/lm-evaluation-harness
cd lm-evaluation-harness
pip install -e .
```

```
lm_eval --model hf \
    --model_args pretrained=EleutherAI/gpt-j-6B \
    --tasks hellaswag \
    --device cuda:0 \
    --batch_size 8
```

Straightforward to use

OpenLLM Leaderboard

OpenLLM Leaderboard

- Created in 2022 by HuggingFace

OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)

OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏

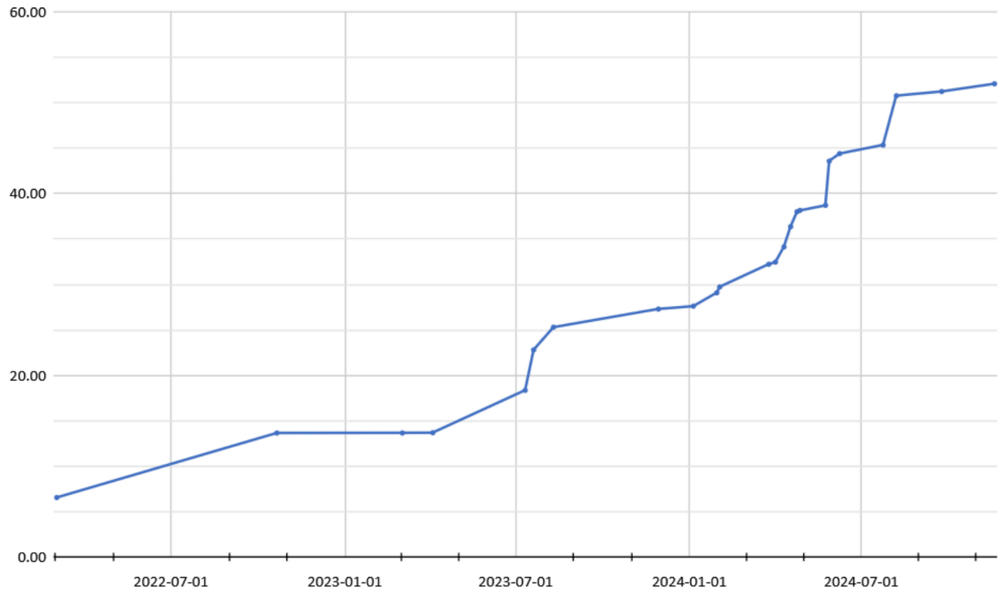
OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025

OpenLLM Leaderboard

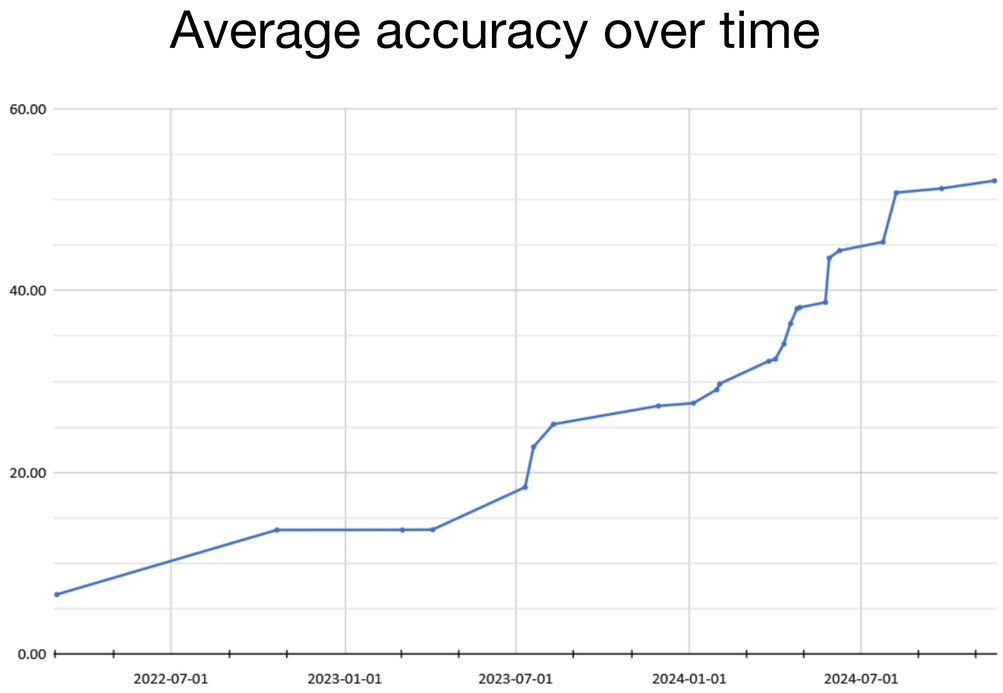
- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025

Average accuracy over time



OpenLLM Leaderboard

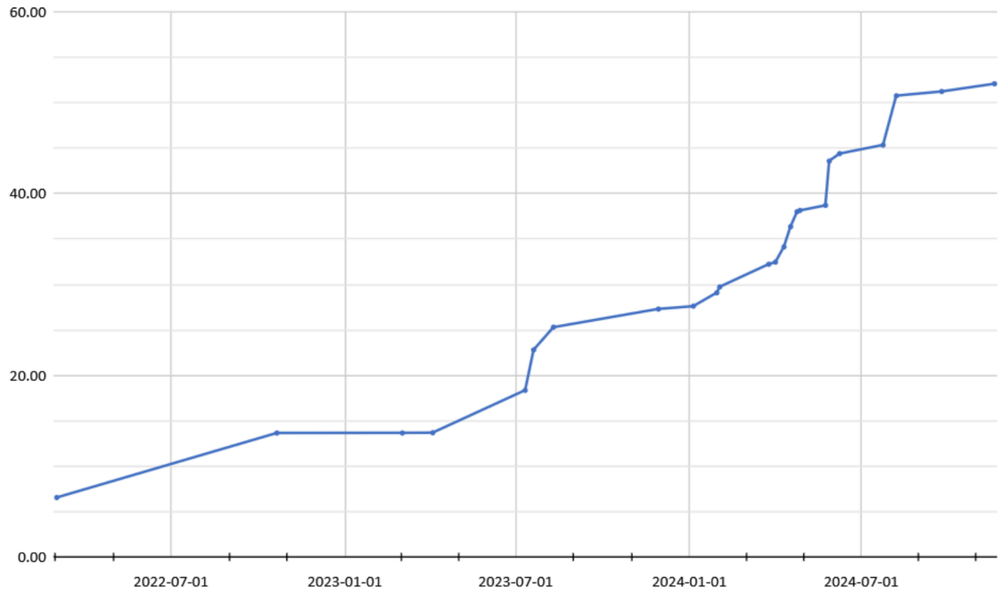
- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025
  - No automatic check for contamination



OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025
  - No automatic check for contamination
  - Top solution could fine-tune arbitrarily on the test set

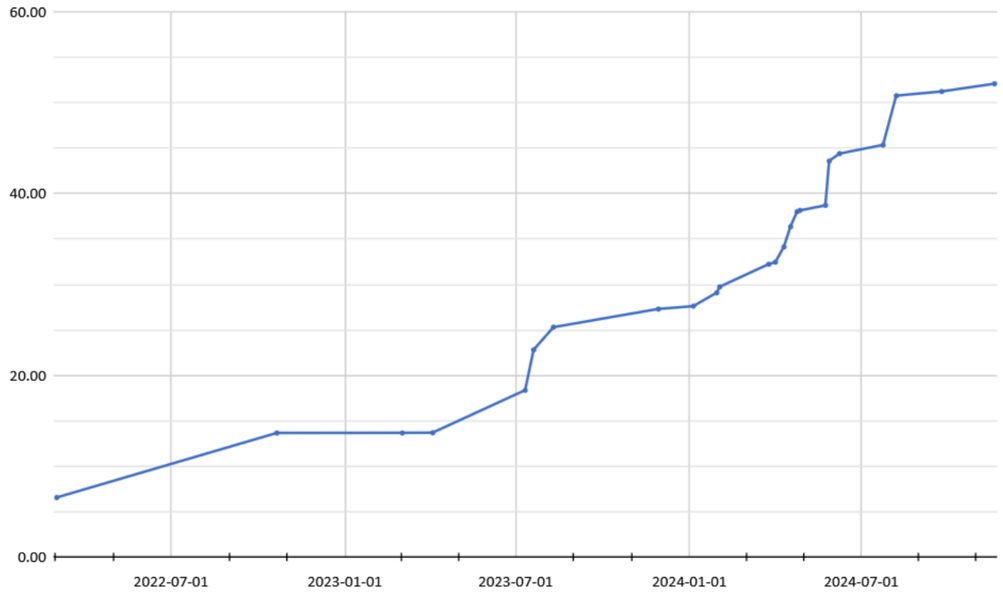
Average accuracy over time



OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025
  - No automatic check for contamination
  - Top solution could fine-tune arbitrarily on the test set
  - Try to fix and then abandon it

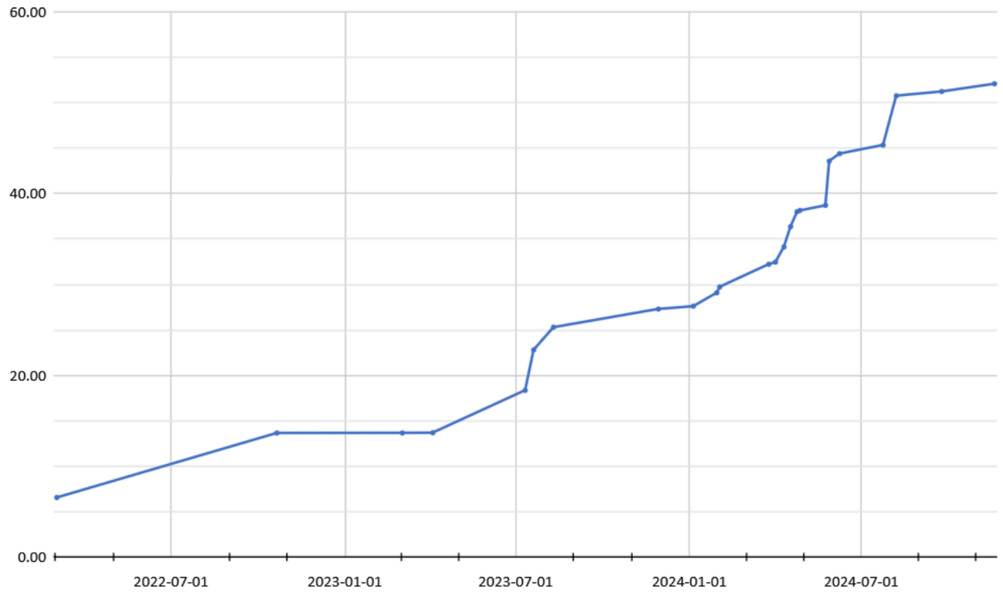
Average accuracy over time



OpenLLM Leaderboard

- Created in 2022 by HuggingFace
- Average scores on a list of static benchmarks (list got updated 2 times)
- Models evaluated **for free** on Hugging Face cluster 🙏
- Widely popular for some time but ended in April 2025
  - No automatic check for contamination
  - Top solution could fine-tune arbitrarily on the test set
  - Try to fix and then abandon it
- Still a massive compute gain to the community 13K models evaluated (~3 million GPU hours cost) and all evolutions available

Average accuracy over time



LLM Dynamic Evaluations

Human evaluations

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Human evaluations

- How to evaluate open ended generated text? 🤔

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Human evaluations

- How to evaluate open ended generated text? 🤔
- Ask iid humans to rank models

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Human evaluations

- How to evaluate open ended generated text? 🤔
- Ask iid humans to rank models

Please say which model is better when answering the question: “What is some cool music from the 1920s?”

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time…

Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording…

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Human evaluations

- How to evaluate open ended generated text? 🤔
- Ask iid humans to rank models

Please say which model is better when answering the question: “What is some cool music from the 1920s?”

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time…

Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording…

Human 🤦

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Human evaluations

- How to evaluate open ended generated text? 🤔
- Ask iid humans to rank models

Please say which model is better when answering the question: “What is some cool music from the 1920s?”

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time…

Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording…

Human 🤦 → **Model 2**

Creative Tasks
Write a haiku about artificial intelligence.
Create a short story that begins with "The last library on Earth closed today."
Generate three marketing slogans for a sustainable clothing brand.

Analysis and Interpretation
What are the main themes in George Orwell's "1984"?
Compare and contrast renewable vs. non-renewable energy sources.
Analyze the potential economic impacts of universal basic income.

Planning and Recommendations
Can you recommend a two day trip to Hawaii?
What's a good study schedule for learning Spanish in 3 months?
Help me plan a vegetarian dinner party for 8 people.

Mathematical and Logical Reasoning
Can you prove the halting theorem?
If a train leaves Chicago at 2 PM traveling 60 mph, and another leaves New York at 3 PM traveling 80 mph, when do they meet?
Explain the prisoner's dilemma and its implications.

Technical Problem-Solving
Debug this Python code that's supposed to sort a list but isn't working properly.
Explain how blockchain technology works in simple terms.

Collecting human preference

And then what?

Please say which model is better when answering the question: “What is some cool music from the 1920s?”
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲

→ **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model

Please say which model is better when answering the question: “What is some cool music from the 1920s?”
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model

- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \dfrac{1}{1 + e^{-\lambda r}}$

Please say which model is better when answering the question: “What is some cool music from the 1920s?”
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲

→ **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \dfrac{1}{1 + e^{-\lambda r}}$
- Often $s = \dfrac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent

Please say which model is better when answering the question: “What is some cool music from the 1920s?”
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍💼

→ **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning
  - **400** difference => **91%** chance of winning

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍💼 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning
  - **400** difference => **91%** chance of winning
  - **200** difference => **76%** chance of winning

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍💼 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning
  - **400** difference => **91%** chance of winning
  - **200** difference => **76%** chance of winning
  - **100** difference => **64%** chance of winning

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \frac{1}{1 + e^{-\lambda r}}$
- Often $s = \frac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning
  - **400** difference => **91%** chance of winning
  - **200** difference => **76%** chance of winning
  - **100** difference => **64%** chance of winning
  - **50** difference => **57%** chance of winning

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Collecting human preference

And then what?

- Scores from battles are often modeled with ELO or Bradley-Terry model
- $\Pr\{\text{A wins against B}\} = \sigma(r_A - r_B), \quad \sigma(r) = \dfrac{1}{1 + e^{-\lambda r}}$
- Often $s = \dfrac{\log 10}{400}$, scores are learned by optimizing log-loss with gradient descent
- Scaling designed so that score difference of 400 correspond to 10:1 odds (i.e., the stronger player is expected to win 10 games for every 1 loss)
- Difference between scores gives probability of winning
  - **400** difference => **91%** chance of winning
  - **200** difference => **76%** chance of winning
  - **100** difference => **64%** chance of winning
  - **50** difference => **57%** chance of winning

Elo Uncovered:
Robustness and Best Practices in Language Model Evaluation

Meriem Boubdir
Cohere for AI
meri.boubdir@gmail.com

Edward Kim
Cohere
edward@cohere.com

Beyza Ermis
Cohere for AI
beyza@cohere.com

Sara Hooker
Cohere for AI
sarahooker@cohere.com

Marzieh Fadaee
Cohere for AI
marzieh@cohere.com

Please say which model is better when answering the question: "What is some cool music from the 1920s?"
Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues, and popular music. Here are some cool music styles and artists from the 1920s:**Jazz** Louis Armstrong: Considered one of the most influential jazz musicians of all time.
Model 2: The 1920s, often referred to as the \"Roaring Twenties,\" was a period that saw a significant explosion in the popularity of jazz and the beginning of widespread commercial recording.

Human 👨‍🦲 → **Model 2**

Human evaluations

Human evaluations

- Getting high on those leaderboard is a **big deal**

Human evaluations

- Getting high on those leaderboard is a **big deal**
- Elo-like ratings for selected models that mostly agrees with intuitive performance

Human evaluations

- Getting high on those leaderboard is a **big deal**
- Elo-like ratings for selected models that mostly agrees with intuitive performance

Text View →

Rank ↑↓	Model ↑↓	Score ↓	Votes ↑↓
1	gemini-3-pro	1490	27 827
2	grok-4.1-thinking	1477	27 985
3	gemini-3-flash	1472	13 245
4	claude-opus-4-5-20251101-thi…	1470	19 898
5	claude-opus-4-5-20251101	1467	21 241
6	grok-4.1	1465	32 015
7	gemini-3-flash (thinking-min…	1462	9 644
8	ernie-5.0-0110	1459	4 829
9	gpt-5.1-high	1458	24 439
10	gemini-2.5-pro	1451	87 641

LM Arena leaderboard Jan 2026

Human evaluations

- Getting high on those leaderboard is a **big deal**
- Elo-like ratings for selected models that mostly agrees with intuitive performance

Text

View →

Rank	Model	Score	Votes
1	gemini-3-pro	1490	27 827
2	grok-4.1-thinking	1477	27 985
3	gemini-3-flash	1472	13 245
4	claude-opus-4-5-20251101-thi…	1470	19 898
5	claude-opus-4-5-20251101	1467	21 241
6	grok-4.1	1465	32 015
7	gemini-3-flash (thinking-min…	1462	9 644
8	ernie-5.0-0110	1459	4 829
9	gpt-5.1-high	1458	24 439
10	gemini-2.5-pro	1451	87 641

LM Arena leaderboard Jan 2026

Total modèles : 77 | **Total votes :** 235 000 | Mise à jour le 14/01/2026

Rang	Modèle	Score de satisfaction BT
1	gemini-3-flash-preview	1132
2	mistral-large-2512	1120
3	mistral-medium-2508	1115
4	gemini-2.5-flash	1104
5	gemini-3-pro-preview	1104
6	qwen3-max-2025-09-23	1103
7	gemini-2.0-flash	1100
8	deepseek-v3-0324	1091
9	magistral-medium	1088
10	gpt-5.2	1085
11	gemma-3-27b	1083
12	grok-4.1-fast	1079
13	deepseek-chat-v3.1	1079
14	deepseek-v3-chat	1078
15	gpt-5.1	1074

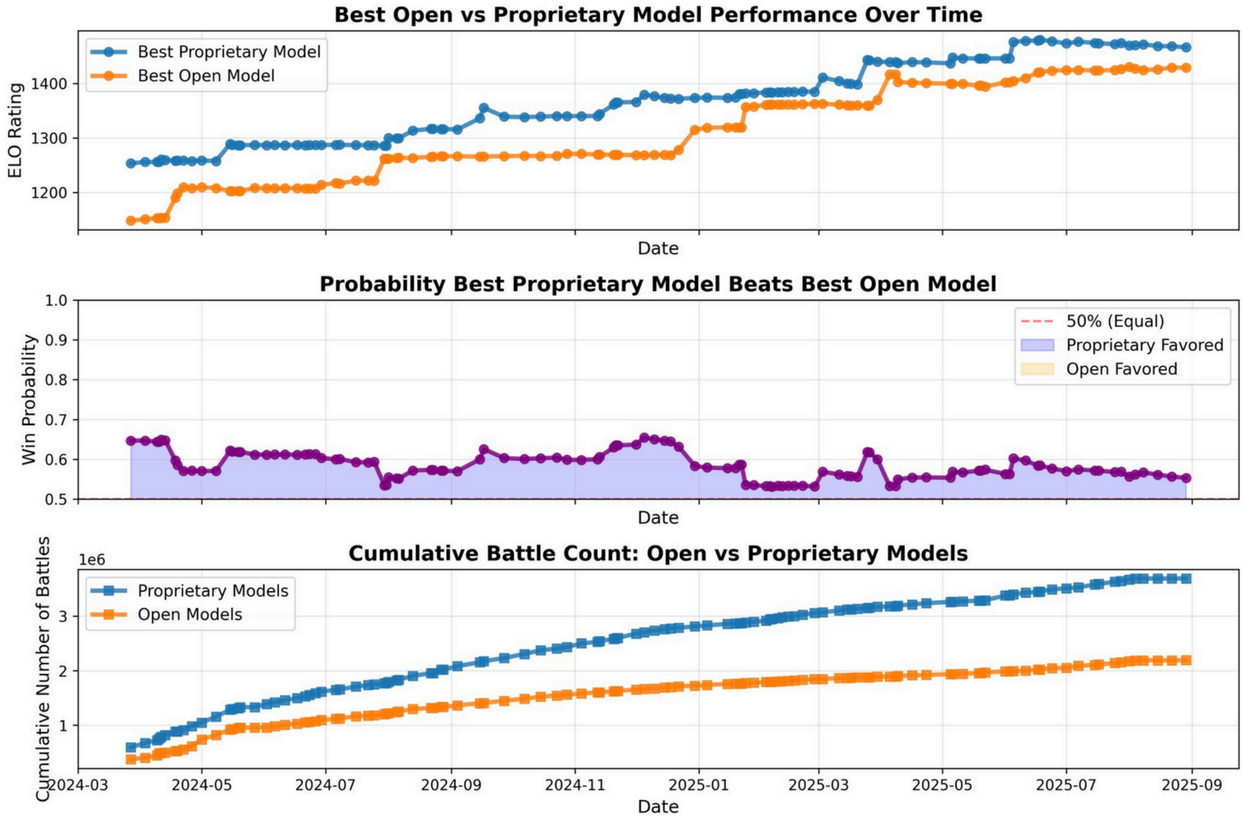
ComparIA leaderboard Jan 2026

Human evaluation

Historical spread between best proprietary and open-weight model

Human evaluation

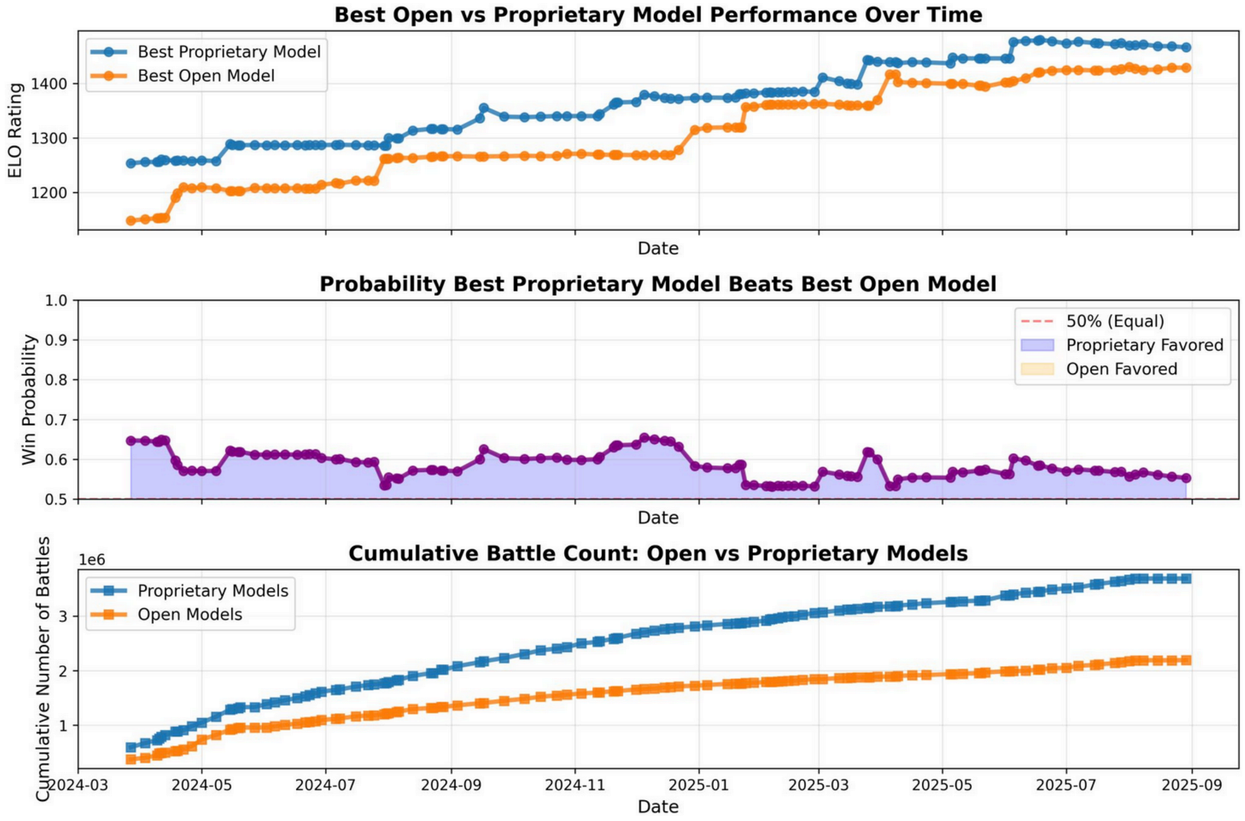
Historical spread between best proprietary and open-weight model



Human evaluation

Historical spread between best proprietary and open-weight model

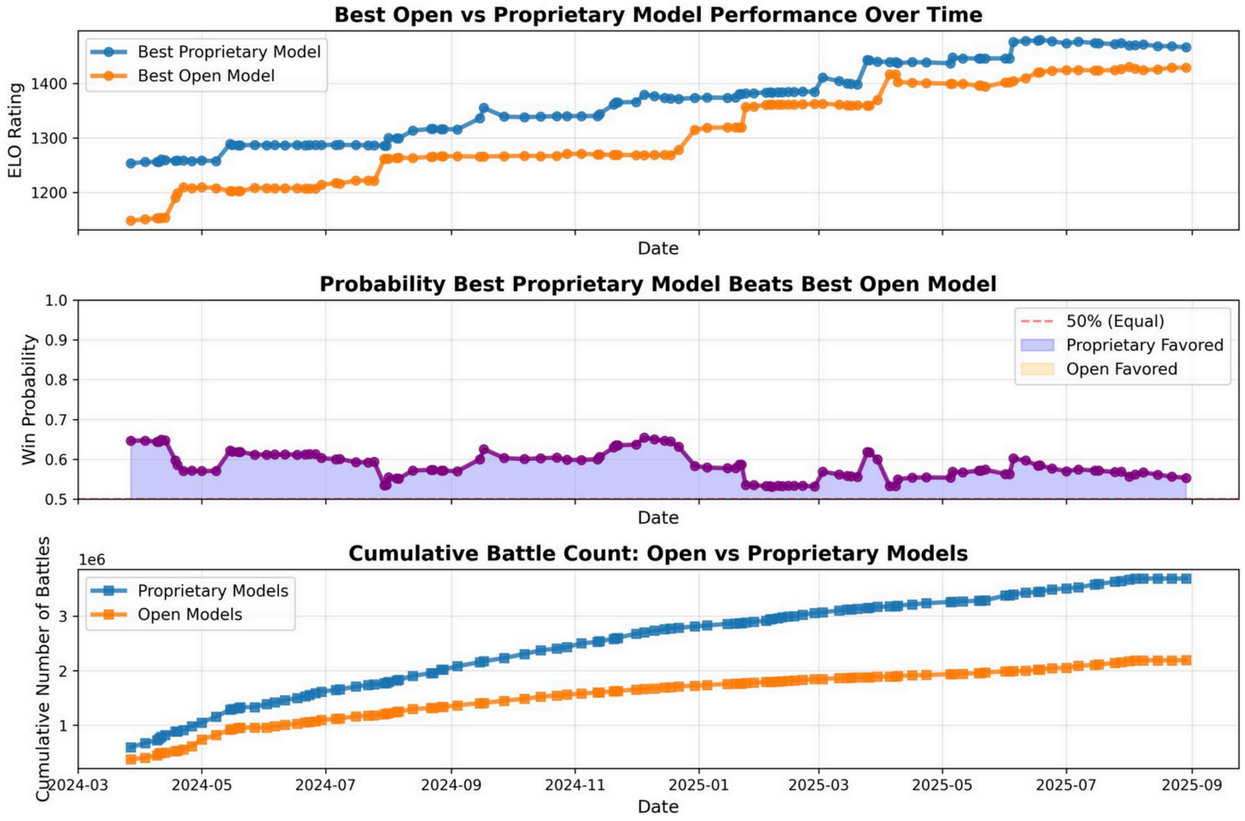
- What is the probability of the best proprietary model winning against the best open weight model?



Human evaluation

Historical spread between best proprietary and open-weight model

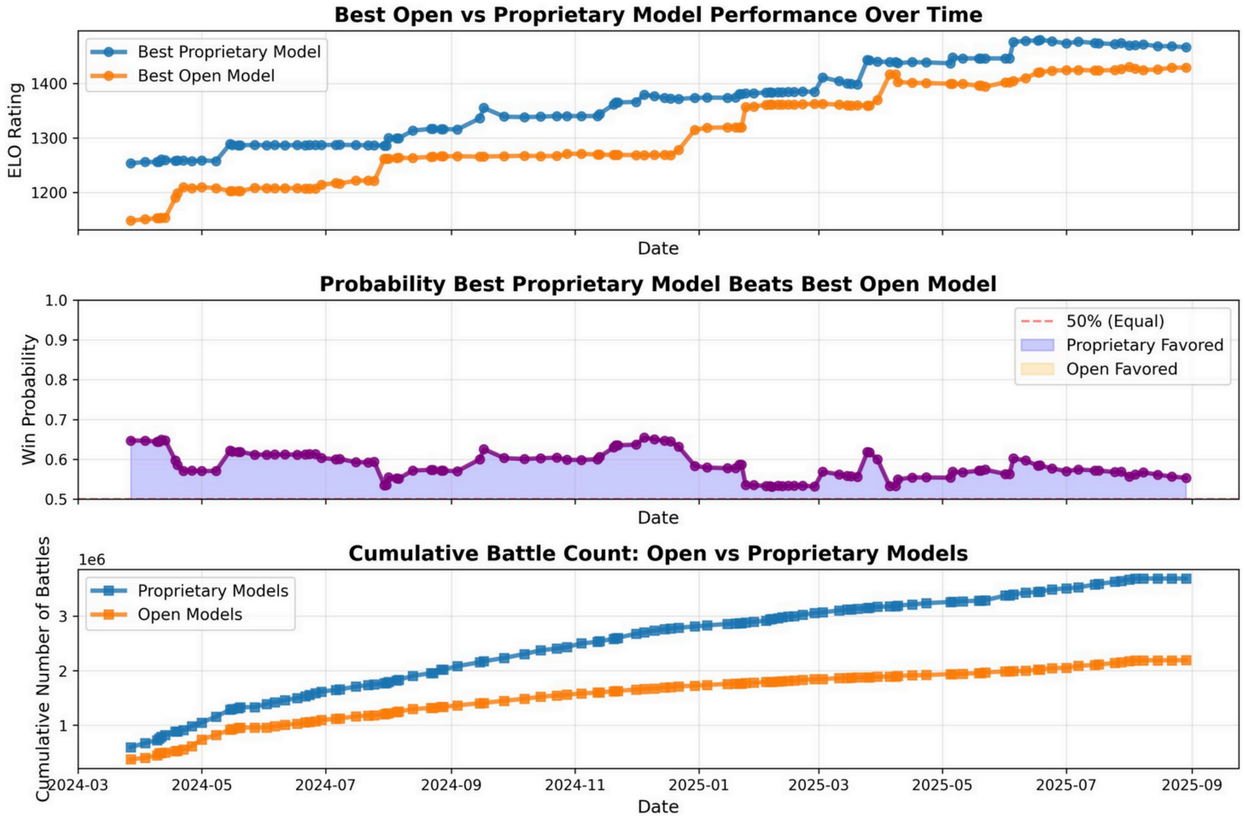
- What is the probability of the best proprietary model winning against the best open weight model?
- All time *low*:



Human evaluation

Historical spread between best proprietary and open-weight model

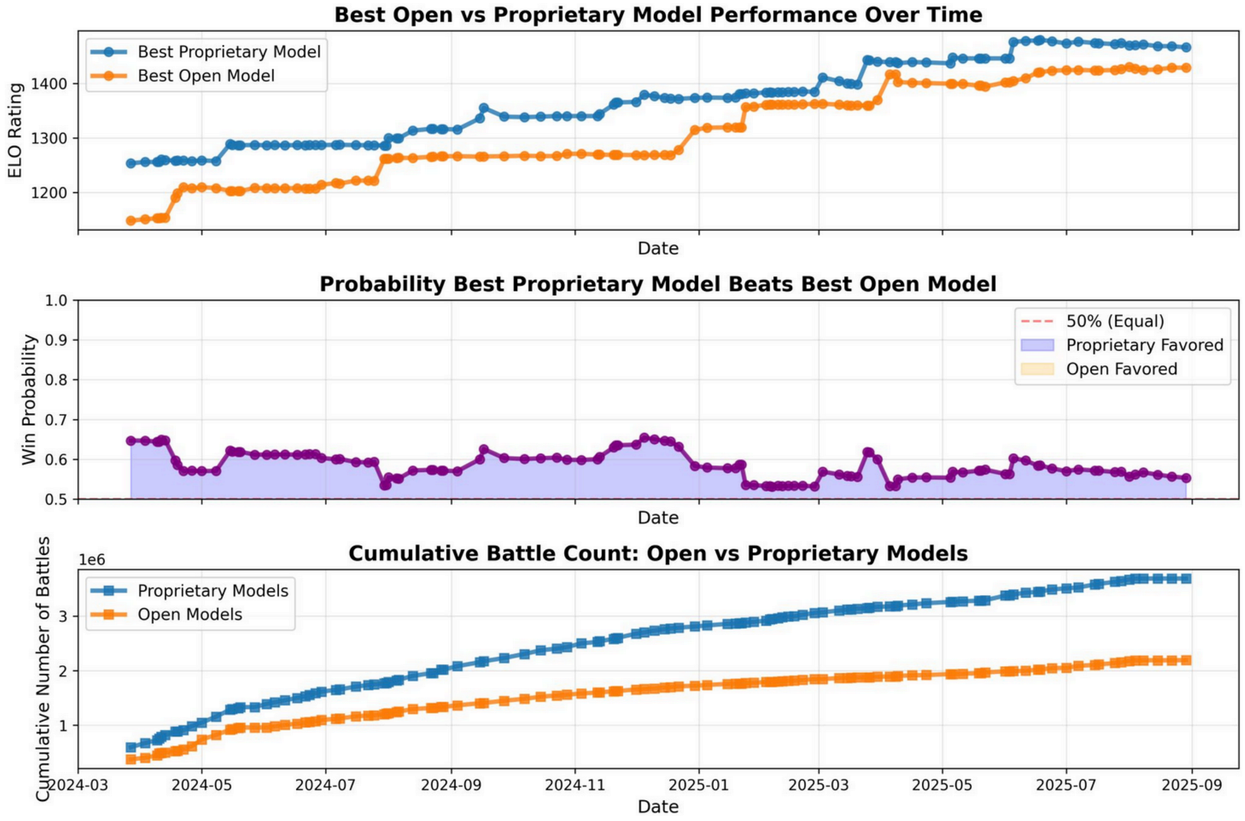
- What is the probability of the best proprietary model winning against the best open weight model?
- All time *low*:
  - Around **55%** chance of best proprietary model winning against best open model



Human evaluation

Historical spread between best proprietary and open-weight model

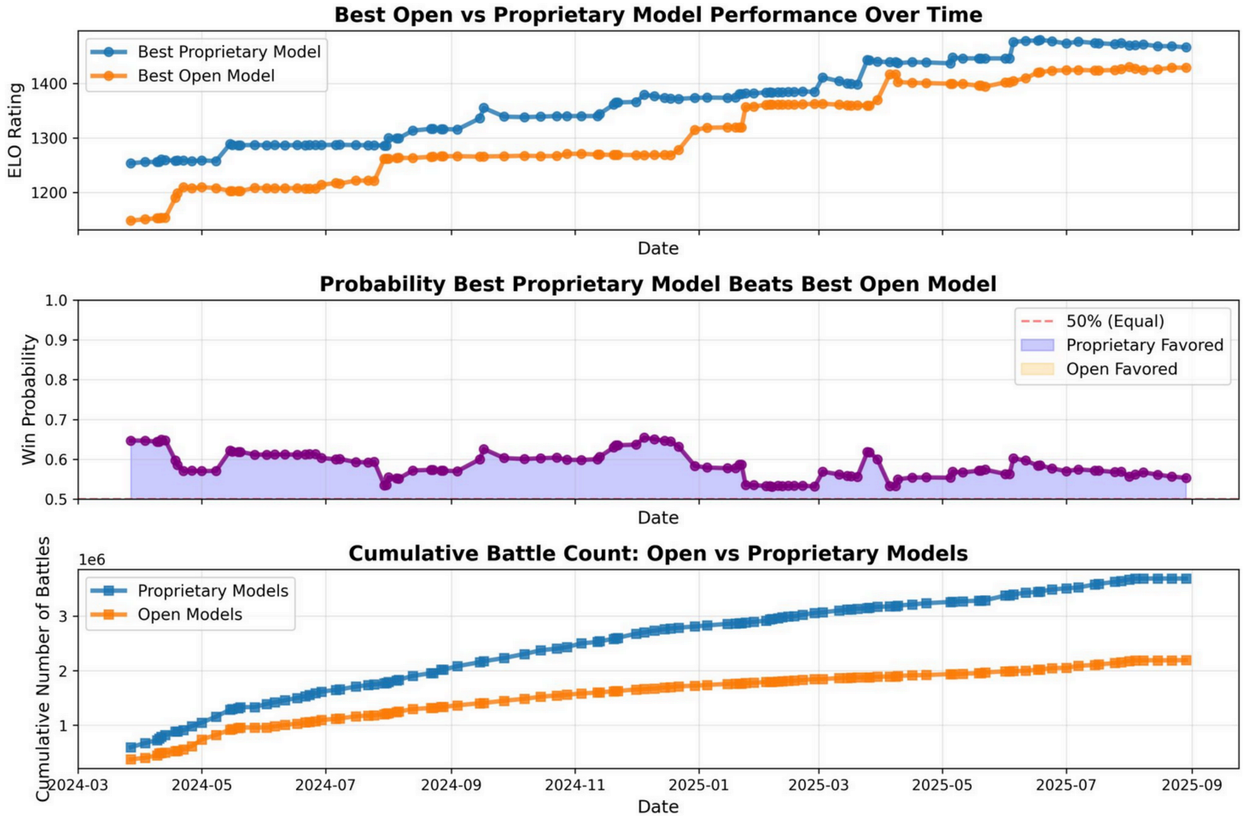
- What is the probability of the best proprietary model winning against the best open weight model?
- All time *low*:
  - Around **55%** chance of best proprietary model winning against best open model
- All time *high*:



Human evaluation

Historical spread between best proprietary and open-weight model

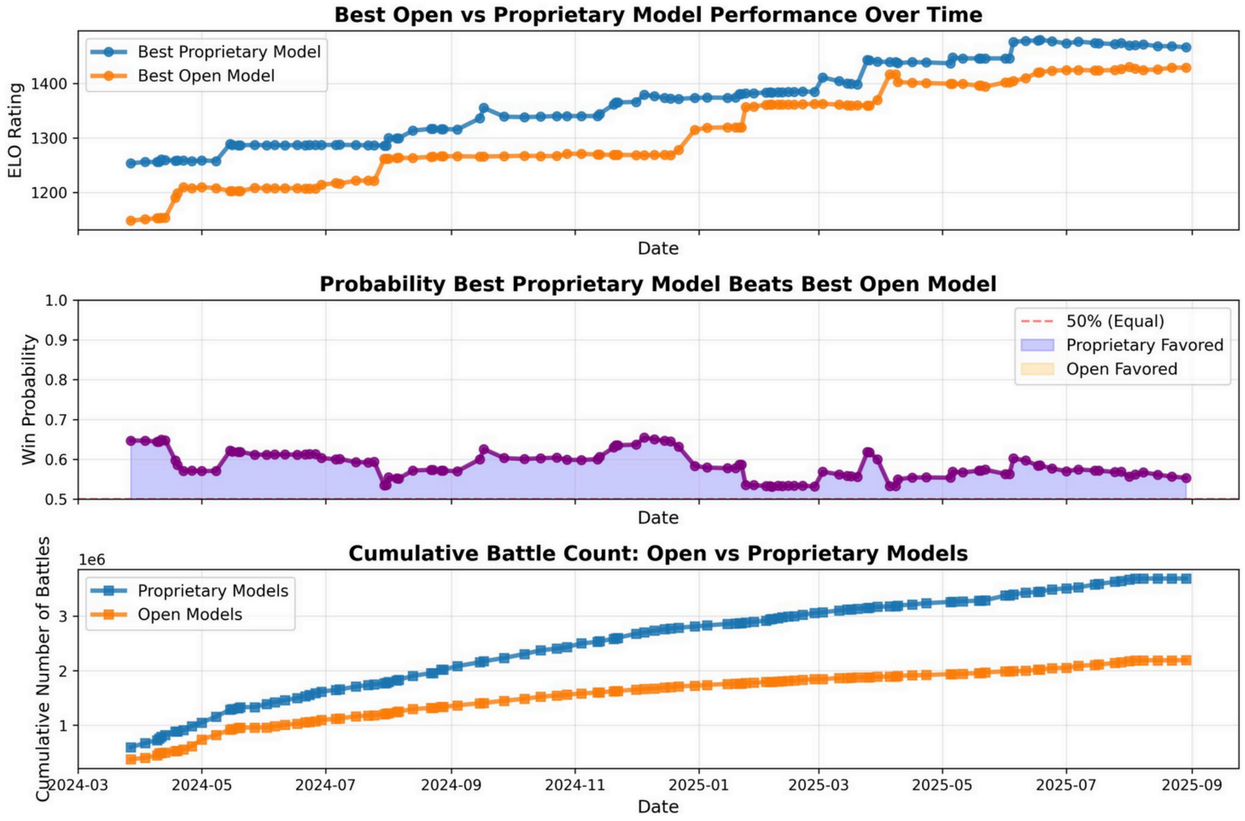
- What is the probability of the best proprietary model winning against the best open weight model?
- All time *low*:
  - Around **55%** chance of best proprietary model winning against best open model
- All time *high*:
  - Around **65%** chance of best proprietary model winning against best open model



Human evaluation

Historical spread between best proprietary and open-weight model

- What is the probability of the best proprietary model winning against the best open weight model?
- All time *low*:
  - Around **55%** chance of best proprietary model winning against best open model
- All time *high*:
  - Around **65%** chance of best proprietary model winning against best open model



Do you know which models reduced the spread on April 2024, August 2024, January 2025 and April 2025? 🤔

Open-source vs proprietary

What reasons for the gap?

Open-source vs proprietary

What reasons for the gap?

- Different ressources:

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility
  - Meta infamously used a dump of pirated books collected on Libgen

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility
  - Meta infamously used a dump of pirated books collected on Libgen

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

B. Meta Produced Facts Underlying the Proposed Amendments One Week Before Filing.

Meta incorrectly contends that Plaintiffs lack "new" evidence justifying leave to amend. Opp. at 6. Plaintiffs promptly filed their Motion after Meta's 30(b)(6) testimony on November 20, 2024, during which Plaintiffs learned for the first time that (a) Meta systematically stripped CMI from the copyrighted works it used to train its models, and (b) Meta torrented LibGen, including seeding that data to make it available online to other users. Plaintiffs who wait until they have

https://www.theguardian.com/books/2025/apr/03/meta-has-stolen-books-authors-to-protest-in-london-against-ai-trained-using-shadow-library

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility
  - Meta infamously used a dump of pirated books collected on Libgen
- Structural gaps:

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

B. Meta Produced Facts Underlying the Proposed Amendments One Week Before Filing.

Meta incorrectly contends that Plaintiffs lack "new" evidence justifying leave to amend. Opp. at 6. Plaintiffs promptly filed their Motion after Meta's 30(b)(6) testimony on November 20, 2024, during which Plaintiffs learned for the first time that (a) Meta systematically stripped CMI from the copyrighted works it used to train its models, and (b) Meta torrented LibGen, including seeding that data to make it available online to other users. Plaintiffs who wait until they have

https://www.theguardian.com/books/2025/apr/03/meta-has-stolen-books-authors-to-protest-in-london-against-ai-trained-using-shadow-library

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility
  - Meta infamously used a dump of pirated books collected on Libgen
- Structural gaps:
  - Models are evaluated mainly on LMArena; a platform which became private

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

B. Meta Produced Facts Underlying the Proposed Amendments One Week Before Filing.

Meta incorrectly contends that Plaintiffs lack "new" evidence justifying leave to amend. Opp. at 6. Plaintiffs promptly filed their Motion after Meta's 30(b)(6) testimony on November 20, 2024, during which Plaintiffs learned for the first time that (a) Meta systematically stripped CMI from the copyrighted works it used to train its models, and (b) Meta torrented LibGen, including seeding that data to make it available online to other users. Plaintiffs who wait until they have

https://www.theguardian.com/books/2025/apr/03/meta-has-stolen-books-authors-to-protest-in-london-against-ai-trained-using-shadow-library

Open-source vs proprietary

What reasons for the gap?

- Different ressources:
  - Humans: Salaries & team sizes
  - GPUs:
    - Gigantic clusters for OpenAI, XAI, Google, …
    - But … significant European clusters dedicated to Research & Startups 🤞
    - To be put in perspective with XAI for example: 200,000 H100s reported for Colossus (!)
- Different rules:
  - Open source models try to follow all legislation, can be stringent on data
  - Private companies have much more flexibility
  - Meta infamously used a dump of pirated books collected on Libgen
- Structural gaps:
  - Models are evaluated mainly on LMArena; a platform which became private
  - Proprietary models have access to phenomenal data, for free

Supercomputer	Country	Number of GPUs	GPU Type(s)
LUMI	Finland	~11,912	AMD MI250X
MareNostrum 5 (BSC)	Spain	4,480	NVIDIA H100 (Hopper)
JUPITER	Germany	~24,000	NVIDIA H100 (in GH200 Grace Hopper Superchips)
Leonardo	Italy	13,824	NVIDIA A100
Jean Zay	France	3,704	Mixed: 1,456 NVIDIA H100 + 416 NVIDIA A100 + 1,832 NVIDIA V100
Alps	Switzerland	10,752	NVIDIA H100 (in GH200 Grace Hopper Superchips)

European clusters 🇪🇺

B. Meta Produced Facts Underlying the Proposed Amendments One Week Before Filing.

Meta incorrectly contends that Plaintiffs lack "new" evidence justifying leave to amend. Opp. at 6. Plaintiffs promptly filed their Motion after Meta's 30(b)(6) testimony on November 20, 2024, during which Plaintiffs learned for the first time that (a) Meta systematically stripped CMI from the copyrighted works it used to train its models, and (b) Meta torrented LibGen, including seeding that data to make it available online to other users. Plaintiffs who wait until they have

https://www.theguardian.com/books/2025/apr/03/meta-has-stolen-books-authors-to-protest-in-london-against-ai-trained-using-shadow-library

LMArena leaderboard

LMArena leaderboard

- LMArena was introduced in 2024 to evaluate LLMs (first named ChatbotArena)

LMArena leaderboard

- LMArena was introduced in 2024 to evaluate LLMs (first named ChatbotArena)
- Pair of completions are shown to the user who pick the best

LMArena leaderboard

- LMArena was introduced in 2024 to evaluate LLMs (first named ChatbotArena)
- Pair of completions are shown to the user who pick the best
- Ratings computed for participating models (Bradley-Terry coefficients) that mostly agrees with intuitive performance

LMArena leaderboard

- LMArena was introduced in 2024 to evaluate LLMs (first named ChatbotArena)
- Pair of completions are shown to the user who pick the best
- Ratings computed for participating models (Bradley-Terry coefficients) that mostly agrees with intuitive performance
- Importance sampling used to select most uncertain pair models

LMArena leaderboard

- LMArena was introduced in 2024 to evaluate LLMs (first named ChatbotArena)
- Pair of completions are shown to the user who pick the best
- Ratings computed for participating models (Bradley-Terry coefficients) that mostly agrees with intuitive performance
- Importance sampling used to select most uncertain pair models

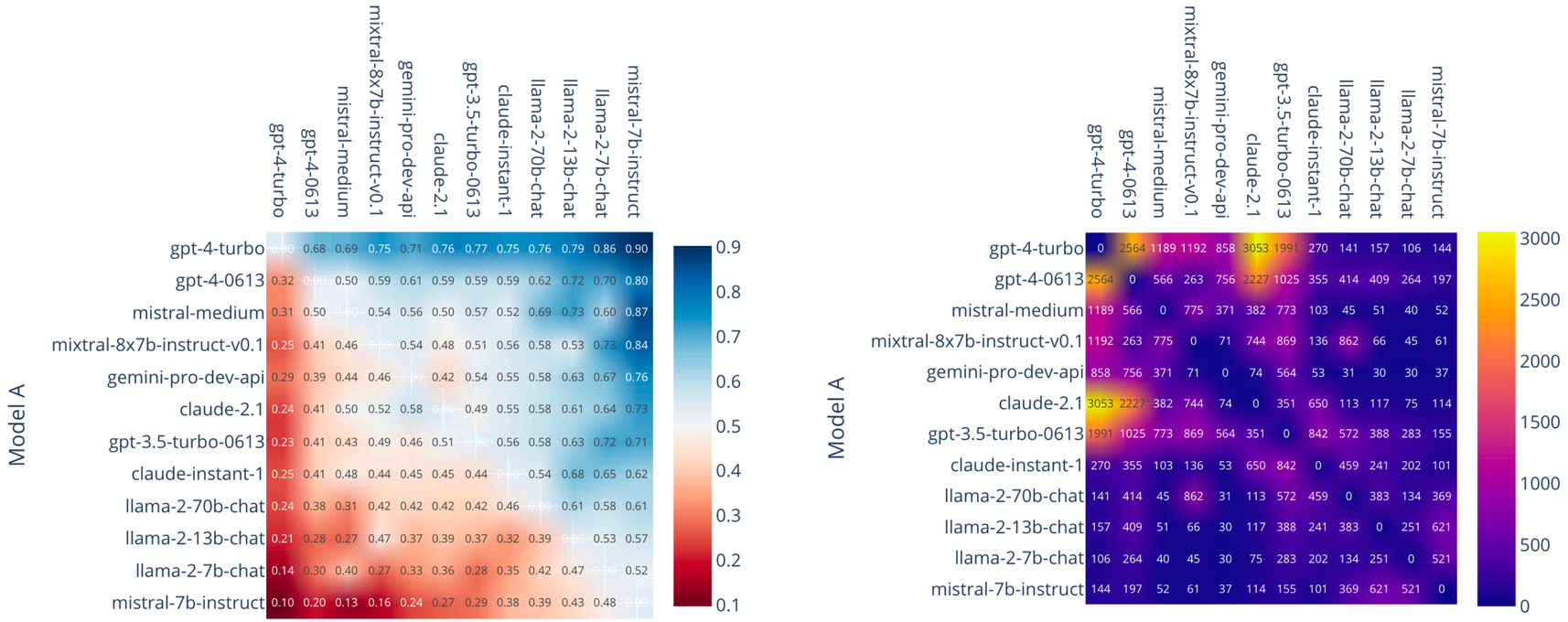


Figure 2. Win-rate (left) and battle count (right) between a subset of models in Chatbot Arena.

Chatbot Arena leaderboard

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:
  - Avoid contamination / overfitting to specific ChatBot Arena users

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:
  - Avoid contamination / overfitting to specific ChatBot Arena users
  - But private company sees **a lot of** data…

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:
  - Avoid contamination / overfitting to specific ChatBot Arena users
  - But private company sees **a lot of** data…
- Need for a transparent leaderboard

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:
  - Avoid contamination / overfitting to specific ChatBot Arena users
  - But private company sees **a lot of** data…
- Need for a transparent leaderboard

The Leaderboard Illusion

Shivalika Singh[*1], Yiyang Nan[1], Alex Wang[2], Daniel D'souza[1], Sayash Kapoor[3], Ahmet Üstün[1], Sanmi Koyejo[4], Yuntian Deng[5], Shayne Longpre[6], Noah A. Smith[7,8], Beyza Ermis[1], Marzieh Fadaee[♦1], and Sara Hooker[♦1]

[1]Cohere Labs, [2]Cohere, [3]Princeton University, [4]Stanford University, [5]University of Waterloo, [6]Massachusetts Institute of Technology, [7]Allen Institute for Artificial Intelligence, [8]University of Washington

Chatbot Arena leaderboard

- **Very small** fraction of the data has been publicly released
- One benefit of not releasing the dataset:
  - Avoid contamination / overfitting to specific ChatBot Arena users
  - But private company sees **a lot of** data…
- Need for a transparent leaderboard

The Leaderboard Illusion

Shivalika Singh[*1], Yiyang Nan[1], Alex Wang[2], Daniel D'souza[1], Sayash Kapoor[3], Ahmet Üstün[1], Sanmi Koyejo[4], Yuntian Deng[5], Shayne Longpre[6], Noah A. Smith[7,8], Beyza Ermis[1], Marzieh Fadaee[♦1], and Sara Hooker[♦1]

[1]Cohere Labs, [2]Cohere, [3]Princeton University, [4]Stanford University, [5]University of Waterloo, [6]Massachusetts Institute of Technology, [7]Allen Institute for Artificial Intelligence, [8]University of Washington

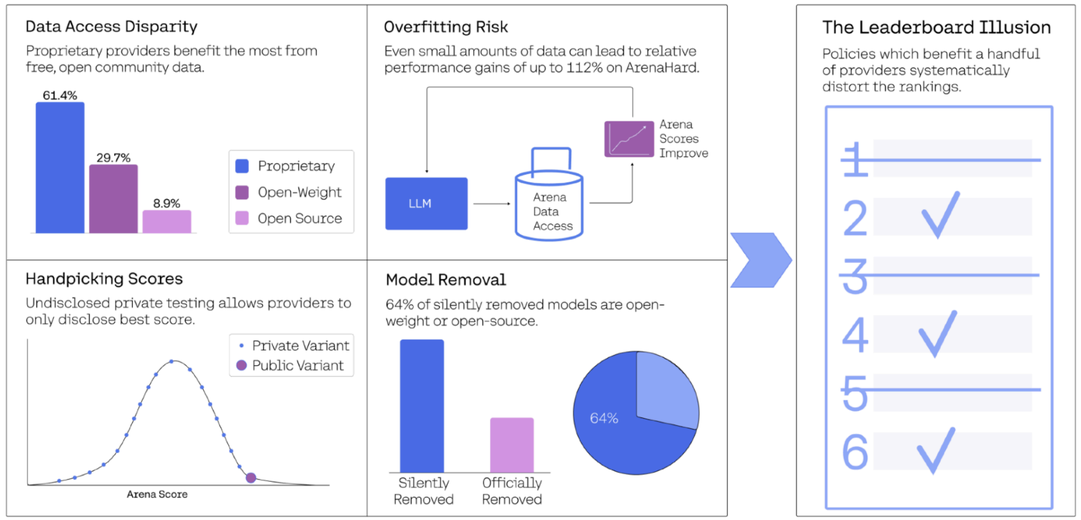


Figure 1: **Overview of key insights**. We investigate the prevalence of **undisclosed private testing and selective score reporting** on the Arena (Section 3), and highlight significant **data access disparities** between proprietary and open-source providers (Section 4.1). These disparities enable **overfitting to the Arena** (Section 4.2). Furthermore, **model deprecation practices** lack transparency, with many models silently deprecated without any notification to providers. We demonstrate how these deprecations contribute to unreliable rankings on the leaderboard (Section 5).

Open-source vs proprietary

Linguistic diversity and Openness

Open-source vs proprietary

Linguistic diversity and Openness

- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/

Open-source vs proprietary

Linguistic diversity and Openness

- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/

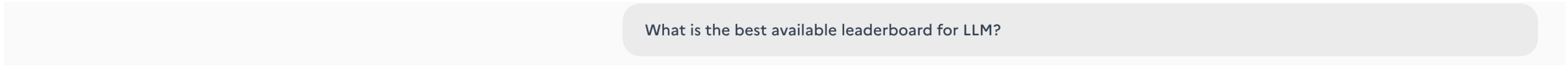
- Entirely open: code & data!

Open-source vs proprietary

Linguistic diversity and Openness

- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/

- Entirely open: code & data!

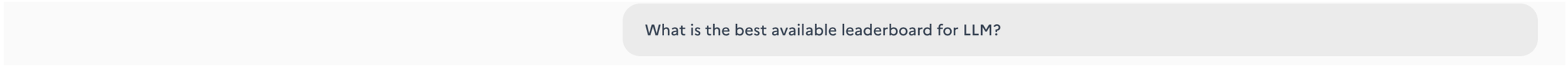


Open-source vs proprietary

Linguistic diversity and Openness

- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/

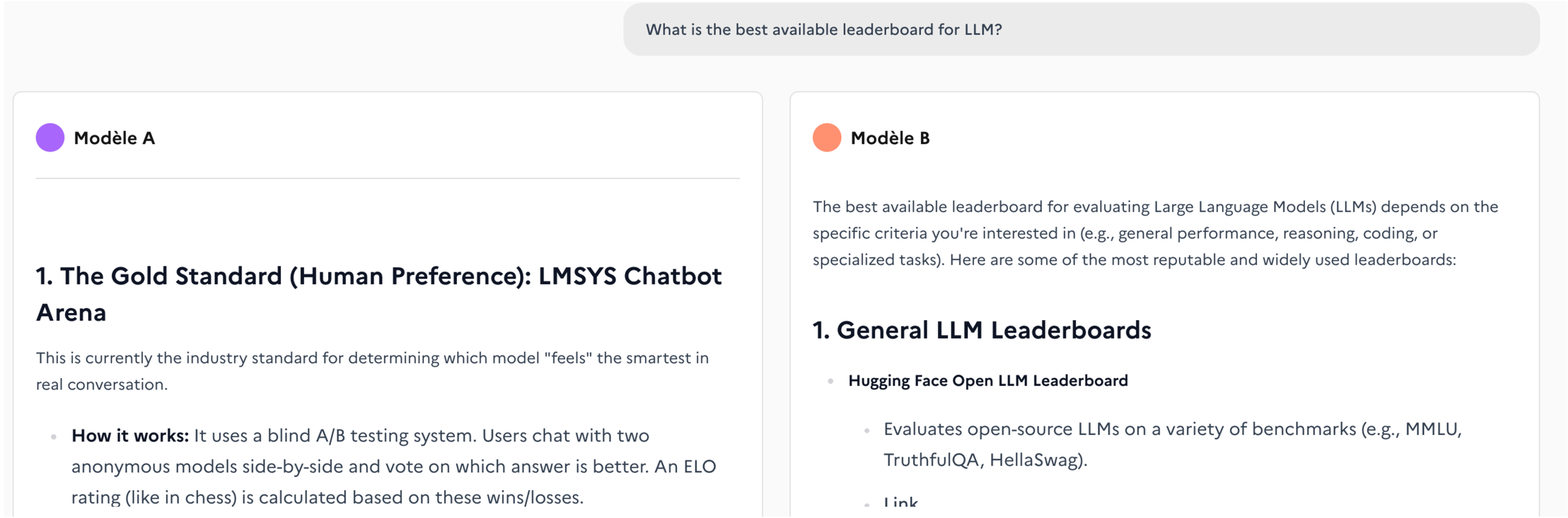
- Entirely open: code & data!



Open-source vs proprietary

Linguistic diversity and Openness

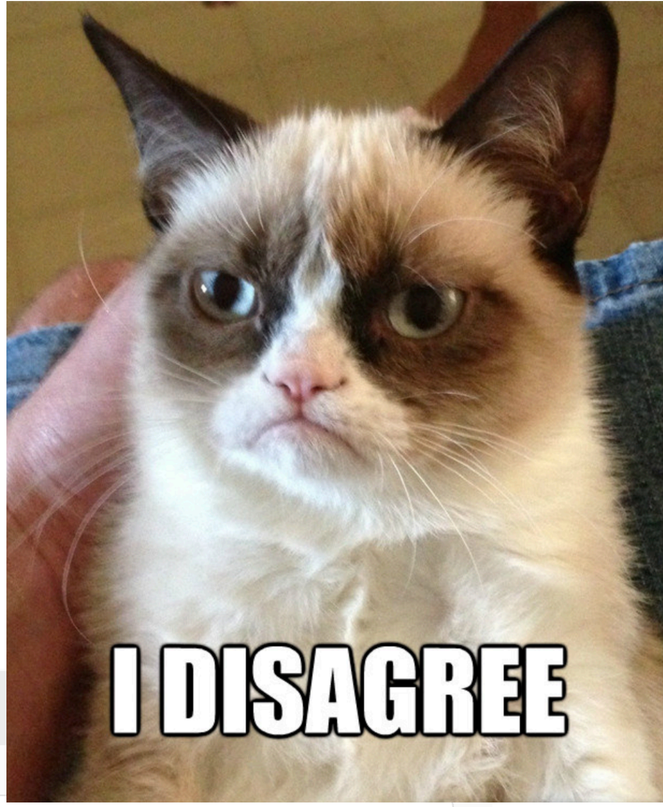
- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/
- Entirely open: code & data!



Open-source vs proprietary

Linguistic diversity and Openness

- Rather than LMArena, consider **ComparIA** https://comparia.beta.gouv.fr/
- Entirely open: code & data!



What is the best available leaderboard for LLM?

Modèle A

1. The Gold Standard (Human Preference): LMSYS Chatbot Arena

This is currently the industry standard for determining which model "feels" the smartest in real conversation.

- **How it works:** It uses a blind A/B testing system. Users chat with two anonymous models side-by-side and vote on which answer is better. An ELO rating (like in chess) is calculated based on these wins/losses.

Modèle B

The best available leaderboard for evaluating Large Language Models (LLMs) depends on the specific criteria you're interested in (e.g., general performance, reasoning, coding, or specialized tasks). Here are some of the most reputable and widely used leaderboards:

1. General LLM Leaderboards

- **Hugging Face Open LLM Leaderboard**
  - Evaluates open-source LLMs on a variety of benchmarks (e.g., MMLU, TruthfulQA, HellaSwag).
  - Link

Open-source vs proprietary

Linguistic diversity and Openness

Open-source vs proprietary

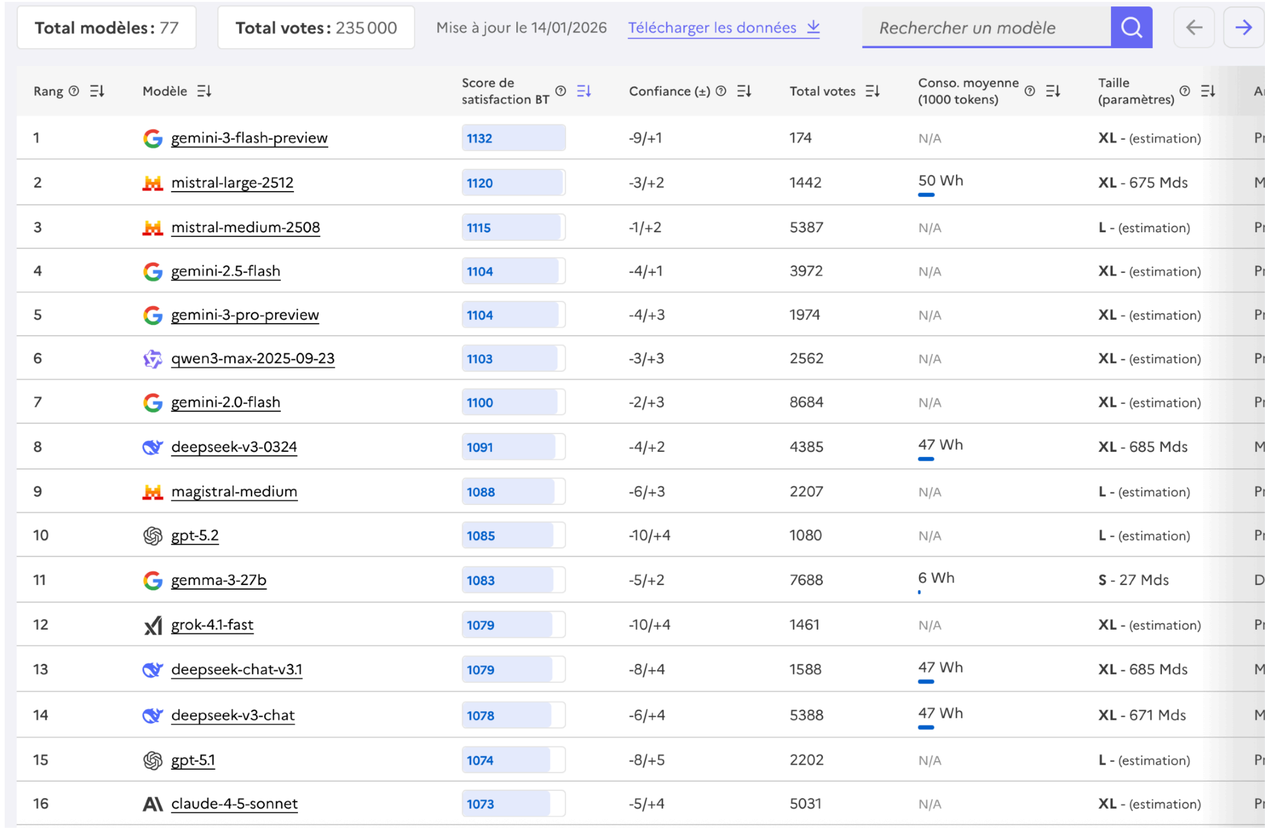
Linguistic diversity and Openness

- What performance for open-weight models?

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?



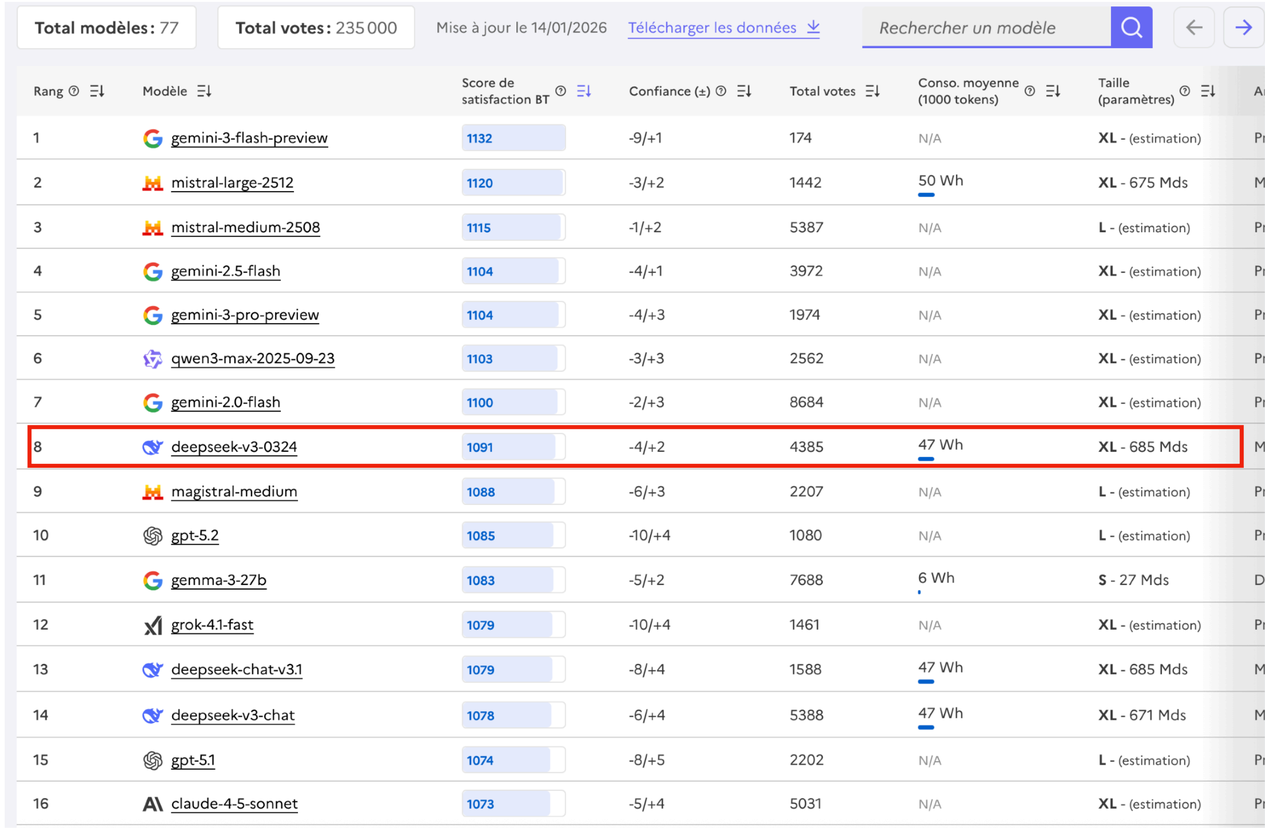
Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?



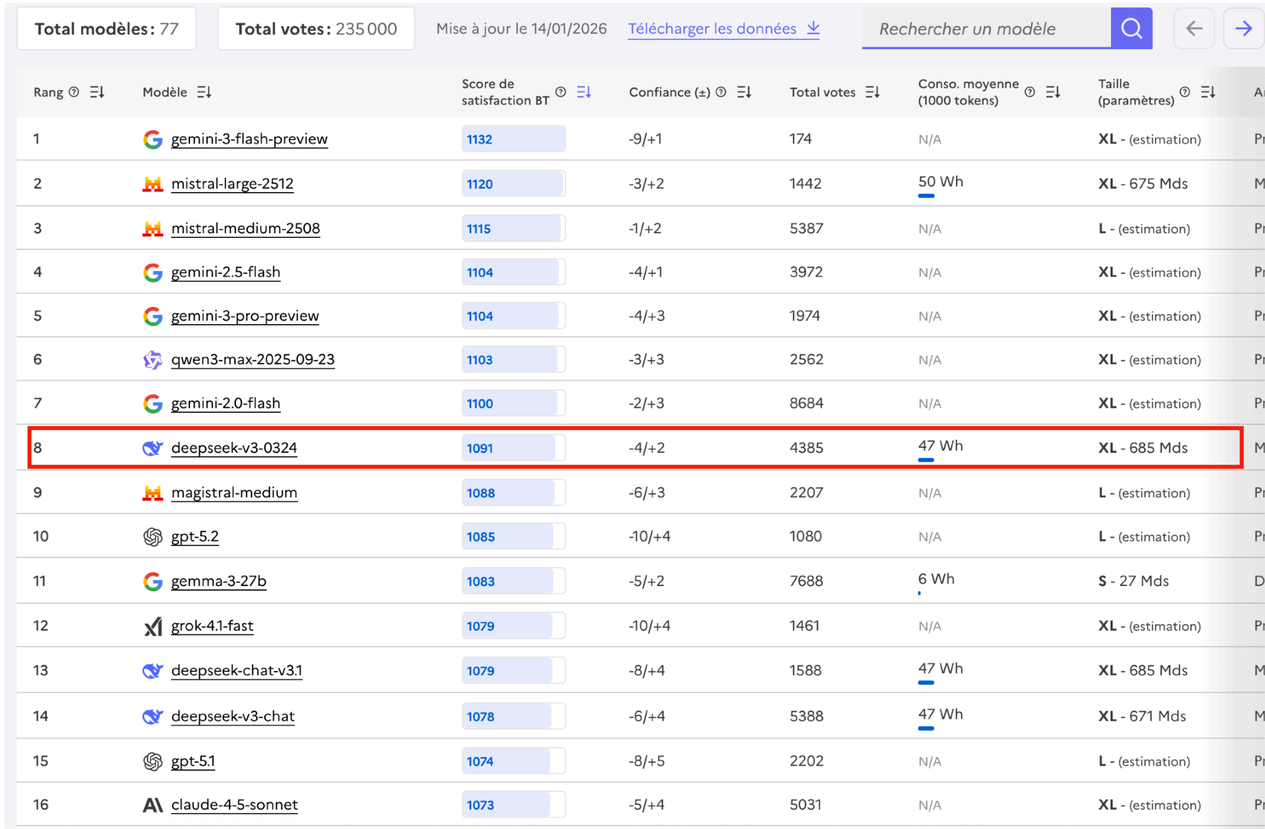
Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?



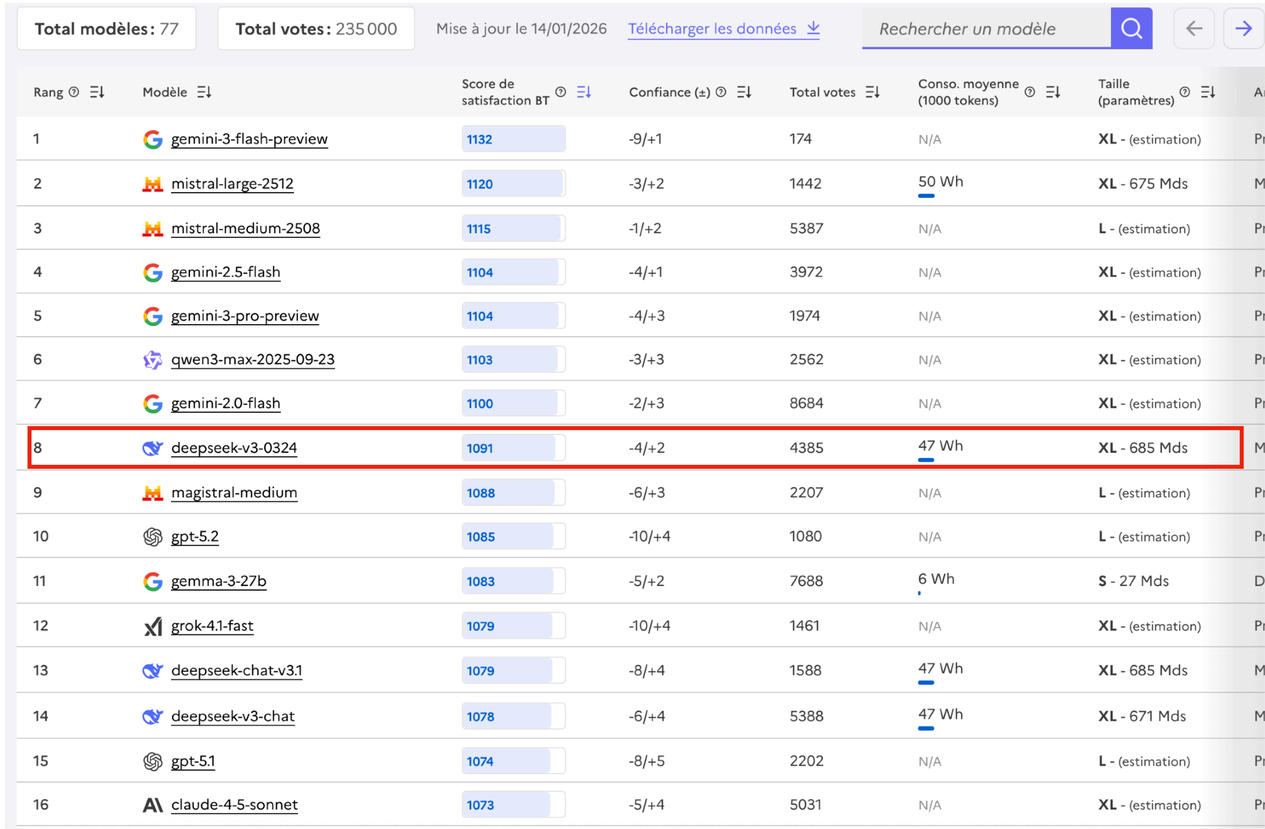
Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?



Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)

...

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)
...						
49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)

...

49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?
- Currently Apertus 🇨🇭, an open-source models is ranked 53 at the level of Llama-3.3-70B

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)
...						
49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?
- Currently Apertus 🇨🇭, an open-source models is ranked 53 at the level of Llama-3.3-70B
  - Release of Llama-3.3-70B: Dec 6 2024

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)
...						
49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?
- Currently Apertus 🇨🇭, an open-source models is ranked 53 at the level of Llama-3.3-70B
  - Release of Llama-3.3-70B: Dec 6 2024
  - Release of Apertus-70B: Sept 17 2025

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)
...						
49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Open-source vs proprietary

Linguistic diversity and Openness

- What performance for open-weight models?
- What performance for European models?
- Currently Apertus 🇨🇭, an open-source models is ranked 53 at the level of Llama-3.3-70B
  - Release of Llama-3.3-70B: Dec 6 2024
  - Release of Apertus-70B: Sept 17 2025
  - **Roughly 9 month gap**

Total modèles: 77 | **Total votes:** 235 000 | Mise à jour le 14/01/2026 | Télécharger les données | Rechercher un modèle

Rang	Modèle	Score de satisfaction BT	Confiance (±)	Total votes	Conso. moyenne (1000 tokens)	Taille (paramètres)
1	gemini-3-flash-preview	1132	-9/+1	174	N/A	**XL** - (estimation)
2	mistral-large-2512	1120	-3/+2	1442	50 Wh	**XL** - 675 Mds
3	mistral-medium-2508	1115	-1/+2	5387	N/A	**L** - (estimation)
4	gemini-2.5-flash	1104	-4/+1	3972	N/A	**XL** - (estimation)
5	gemini-3-pro-preview	1104	-4/+3	1974	N/A	**XL** - (estimation)
6	qwen3-max-2025-09-23	1103	-3/+3	2562	N/A	**XL** - (estimation)
7	gemini-2.0-flash	1100	-2/+3	8684	N/A	**XL** - (estimation)
8	deepseek-v3-0324	1091	-4/+2	4385	47 Wh	**XL** - 685 Mds
9	magistral-medium	1088	-6/+3	2207	N/A	**L** - (estimation)
10	gpt-5.2	1085	-10/+4	1080	N/A	**L** - (estimation)
11	gemma-3-27b	1083	-5/+2	7688	6 Wh	**S** - 27 Mds
12	grok-4.1-fast	1079	-10/+4	1461	N/A	**XL** - (estimation)
13	deepseek-chat-v3.1	1079	-8/+4	1588	47 Wh	**XL** - 685 Mds
14	deepseek-v3-chat	1078	-6/+4	5388	47 Wh	**XL** - 671 Mds
15	gpt-5.1	1074	-8/+5	2202	N/A	**L** - (estimation)
16	claude-4-5-sonnet	1073	-5/+4	5031	N/A	**XL** - (estimation)
...						
49	llama-3.3-70b	985	-4/+4	9113	12 Wh	**M** - 70 Mds
50	o3-mini	981	-9/+7	1619	N/A	**S** - (estimation)
51	gpt-4o-mini-2024-07-18	980	-4/+5	6990	N/A	**S** - (estimation)
52	gpt-4.1-nano	977	-6/+4	5717	N/A	**S** - (estimation)
53	Apertus-70B-Instruct-2509	976	-7/+7	2103	12 Wh	**M** - 70 Mds
54	aya-expanse-8b	972	-9/+6	1302	4 Wh	**XS** - 8 Mds
55	minimax-m2	971	-8/+7	1375	8 Wh	**L** - 230 Mds
56	llama-3.1-70b	971	-4/+4	5583	12 Wh	**M** - 70 Mds

Evaluation

Evaluating… at which cost?

Evaluation

Evaluating… at which cost?

- Human annotations is expensive

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations

Evaluation

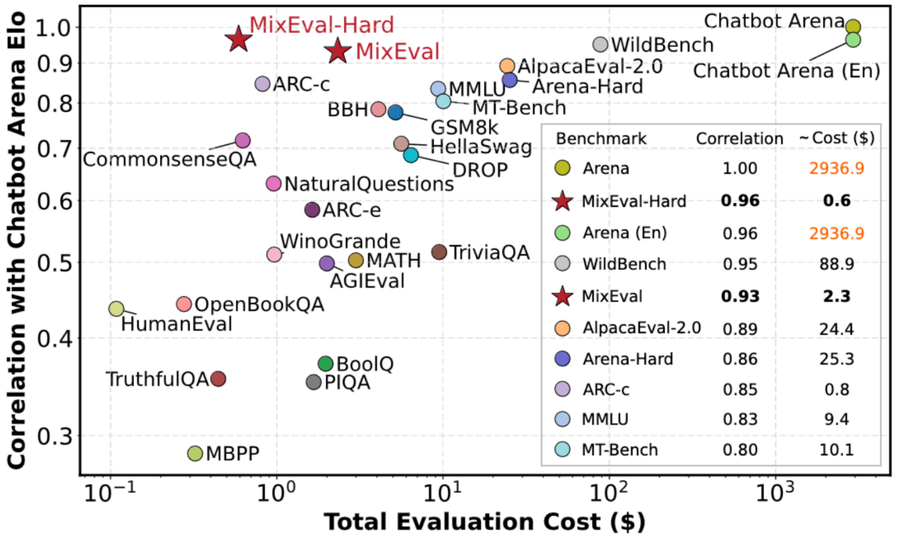
Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations
- Can we trade off accuracy for cost and speed?

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations
- Can we trade off accuracy for cost and speed?



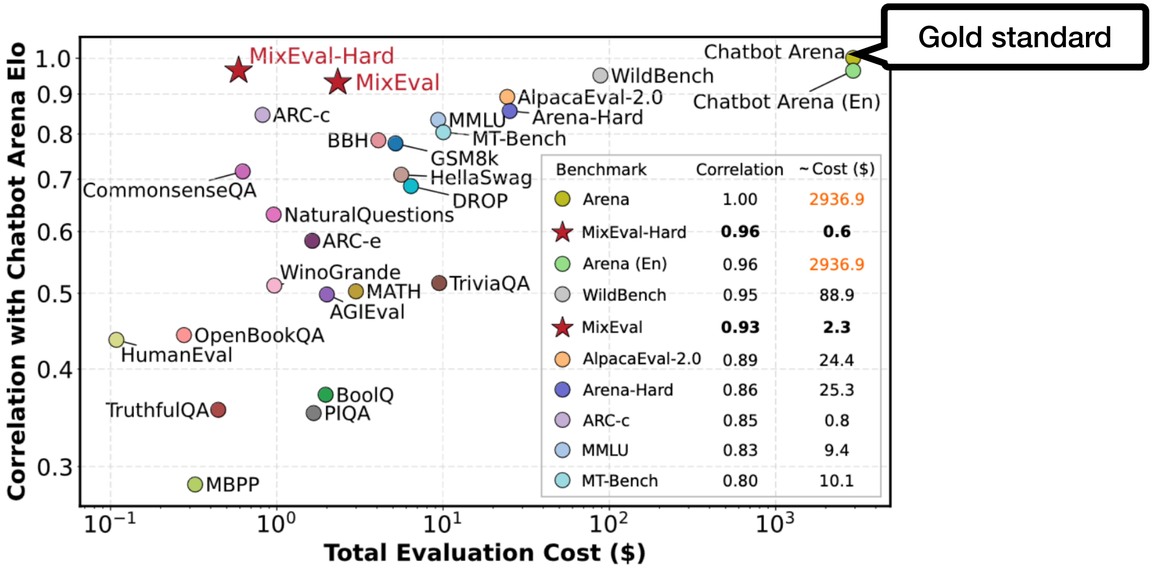
LLM evaluation cost and correlation with Elo ratings

Source: MixEval [Ni 2024]

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations
- Can we trade off accuracy for cost and speed?



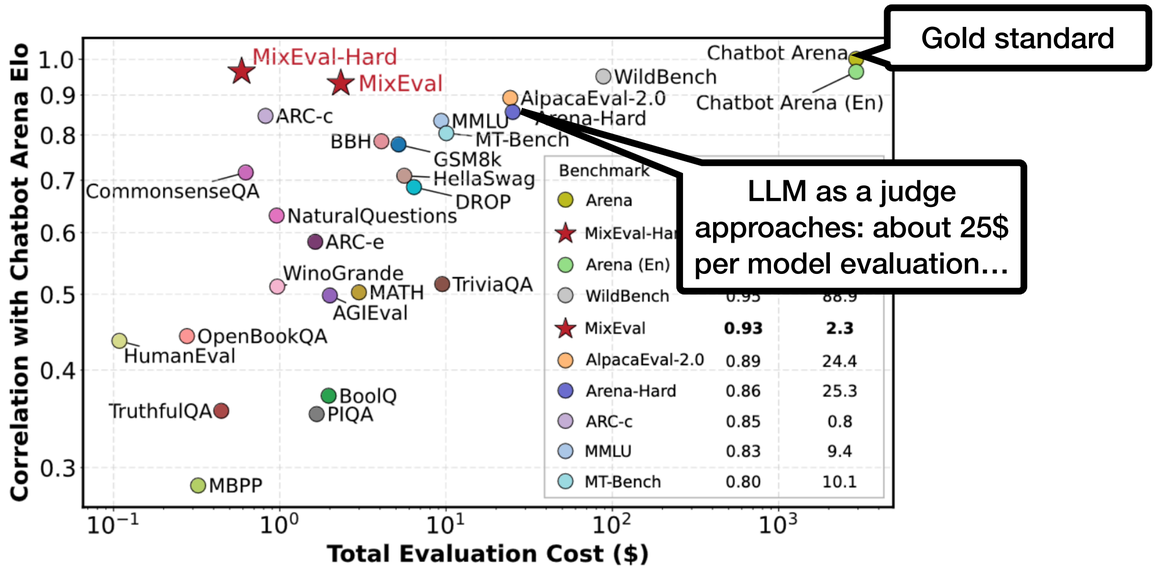
LLM evaluation cost and correlation with Elo ratings

Source: MixEval [Ni 2024]

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations
- Can we trade off accuracy for cost and speed?



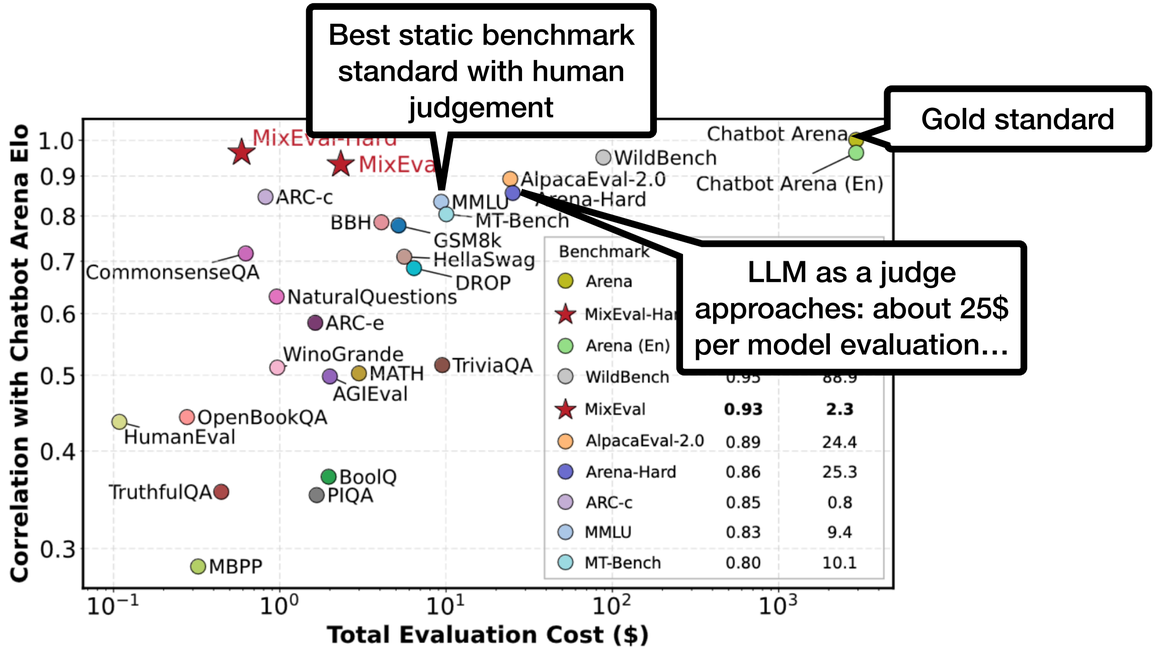
LLM evaluation cost and correlation with Elo ratings

Source: MixEval [Ni 2024]

Evaluation

Evaluating… at which cost?

- Human annotations is expensive
- ~3.5K$ to annotate one model if annotators would be paid minimum US wage)
- Also need to wait for human annotations
- Can we trade off accuracy for cost and speed?



LLM evaluation cost and correlation with Elo ratings

Source: MixEval [Ni 2024]

LLM Judges

LLM as a judge

Overview

LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better

LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better

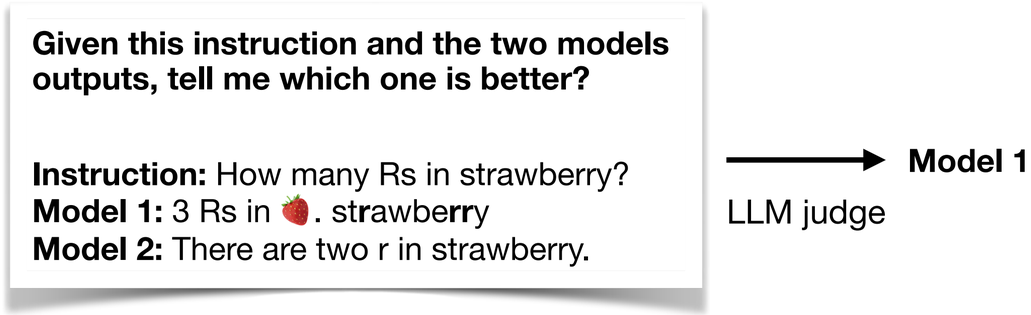
Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

LLM as a judge

Overview

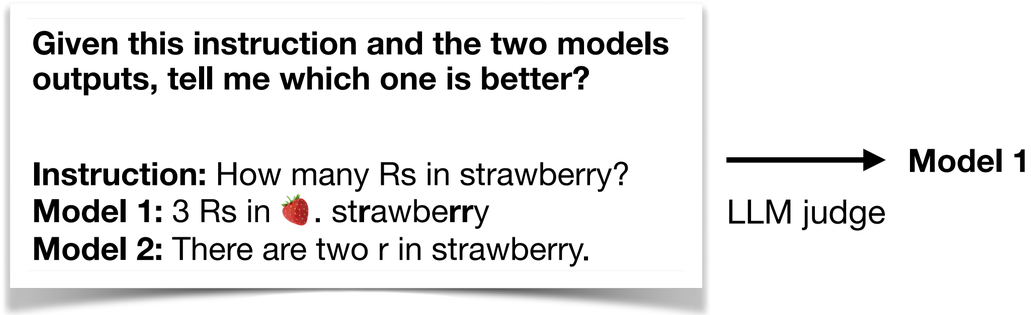
- Idea 💡: ask LLM to tell which LLM output is better



LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better
- In some cases, verification is easier than generation



LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better
- In some cases, verification is easier than generation

=> it is easier to **verify** a 3-SAT instance than **solving** it.

$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better
- In some cases, verification is easier than generation
- Useful for:

=> it is easier to **verify** a 3-SAT instance than **solving** it.

$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.



Model 1

LLM judge

LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better
- In some cases, verification is easier than generation
- Useful for:
  - Cheaper leaderboards: estimate what would be your ranking without having to involve expensive human annotations

=> it is easier to **verify** a 3-SAT instance than **solving** it.

$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

LLM as a judge

Overview

- Idea 💡: ask LLM to tell which LLM output is better
- In some cases, verification is easier than generation
- Useful for:
  - Cheaper leaderboards: estimate what would be your ranking without having to involve expensive human annotations
  - Model development: quickly probe the strengths/weaknesses of your model

=> it is easier to **verify** a 3-SAT instance than **solving** it.

$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

LLM as a judge

Grid evaluations for leaderboards

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

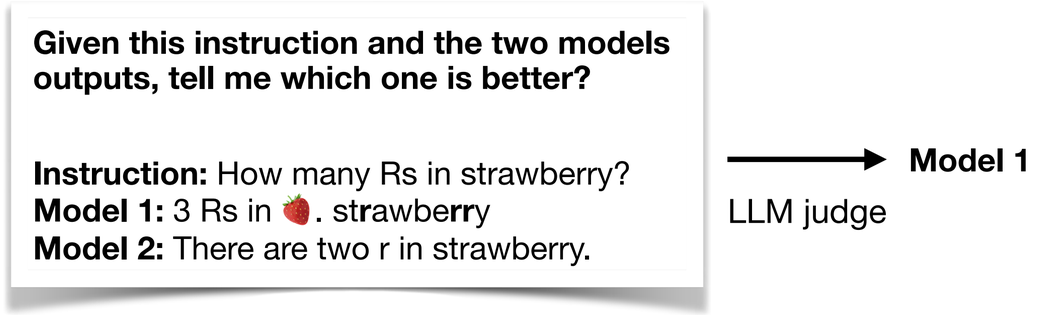
→ **Model 1**

LLM judge

LLM as a judge

Grid evaluations for leaderboards

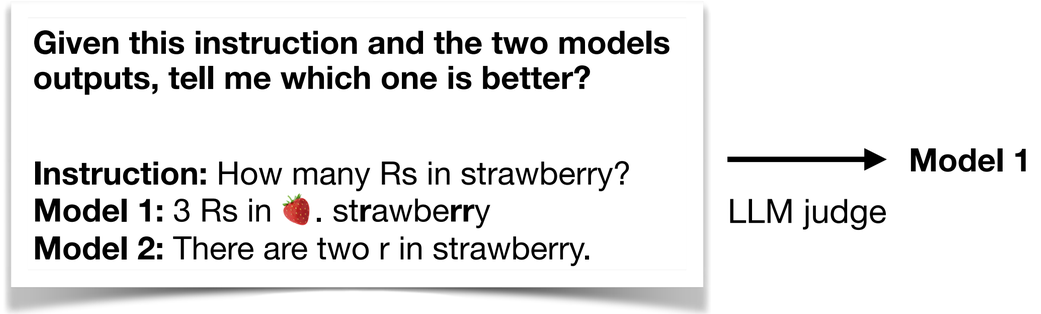
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid



LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline



LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

Baseline	Model 1	$\cdots$	Model m

LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

Baseline	Model 1	$\cdots$	Model m

LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

	Baseline	Model 1	$\cdots$	Model m
Instruction 1: Write a poem				
$\cdots$				
Instruction n: How many Rs in 🍓?				

LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. st**r**awbe**rr**y
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

Baseline	Model 1	$\cdots$	Model m
Instruction 1: Write a poem			
$\cdots$			
Instruction n: How many Rs in 🍓?			

LLM as a judge

Grid evaluations for leaderboards

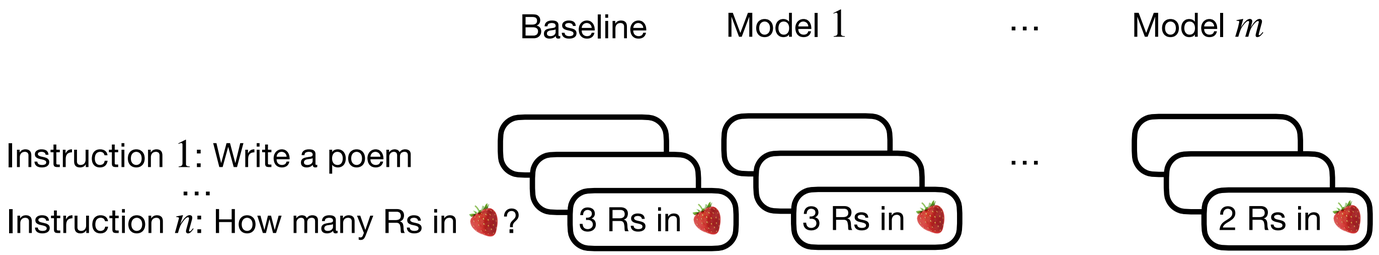
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



LLM as a judge

Grid evaluations for leaderboards

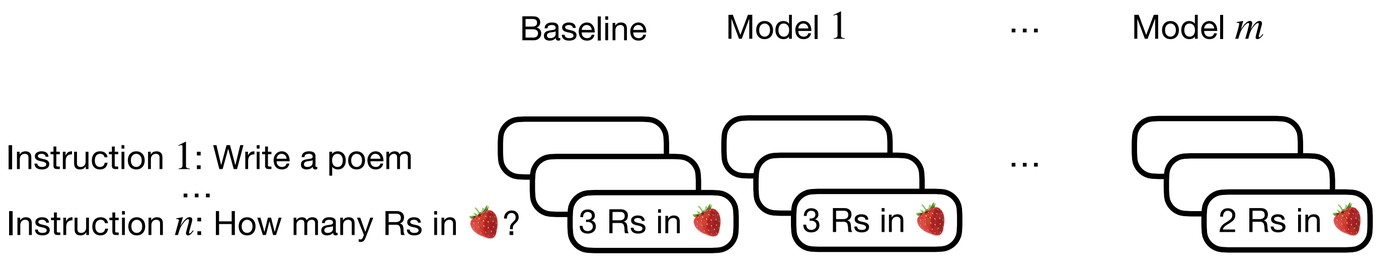
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



LLM as a judge

Grid evaluations for leaderboards

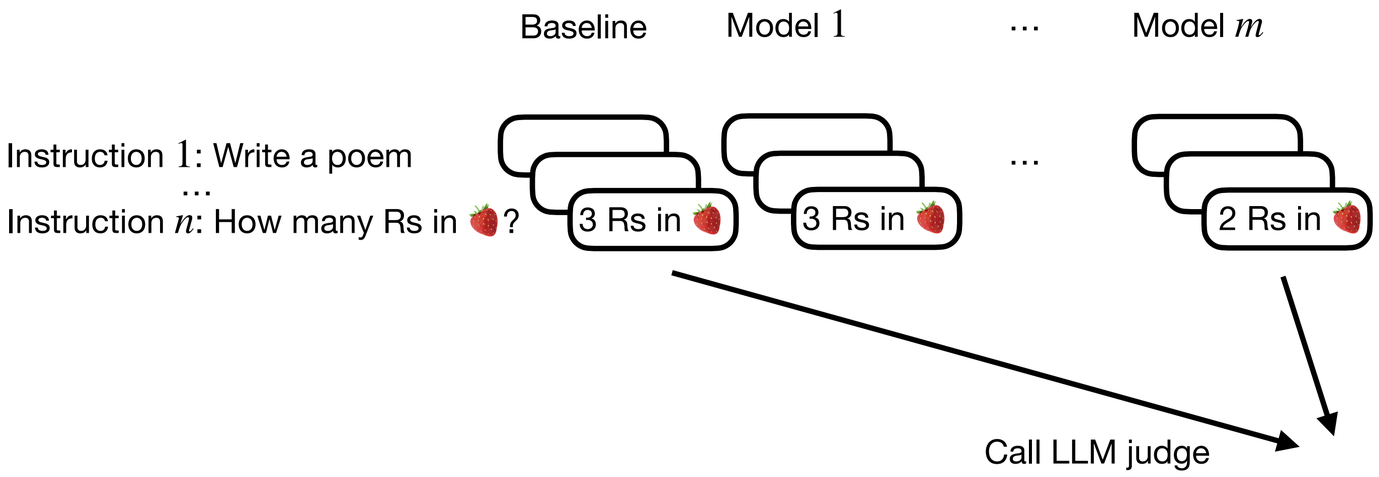
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



LLM as a judge

Grid evaluations for leaderboards

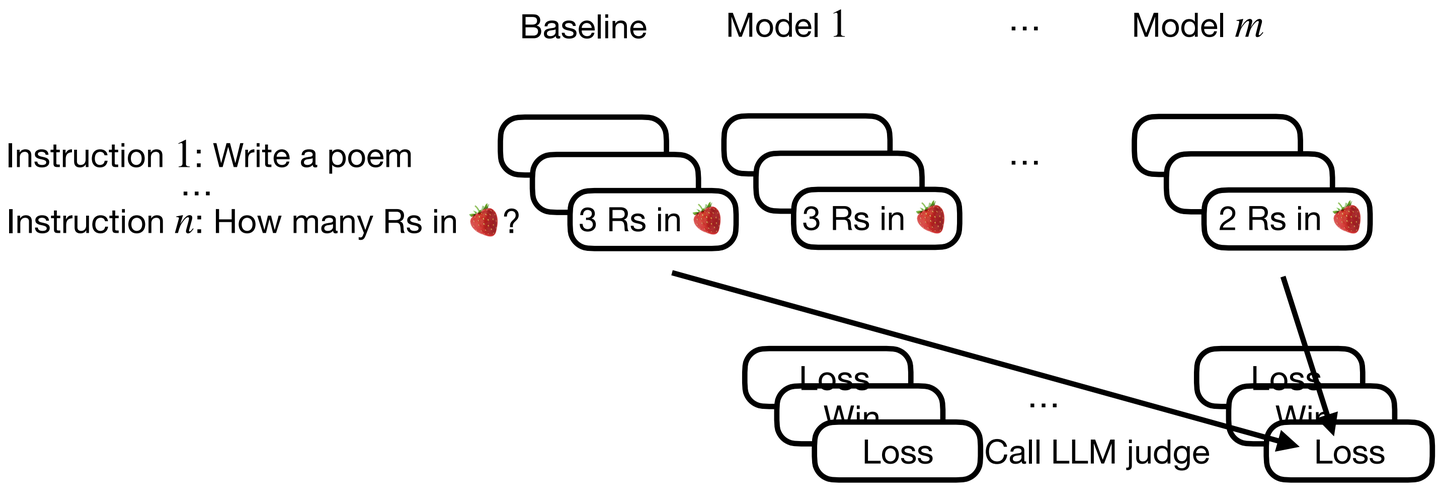
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



LLM as a judge

Grid evaluations for leaderboards

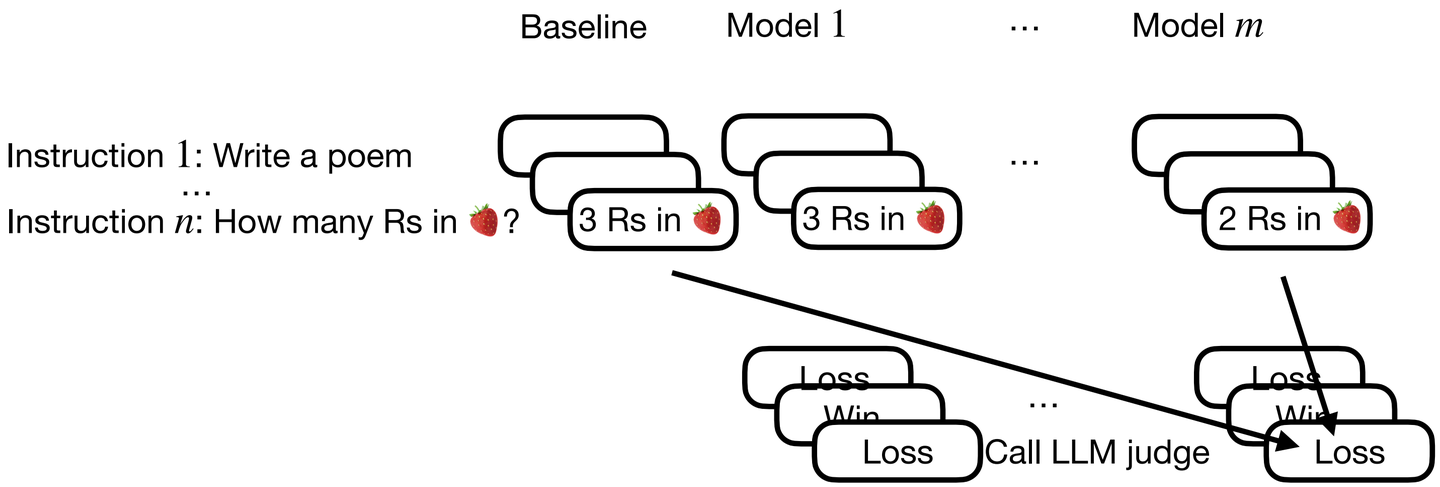
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge
- Report win rate and rank models

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

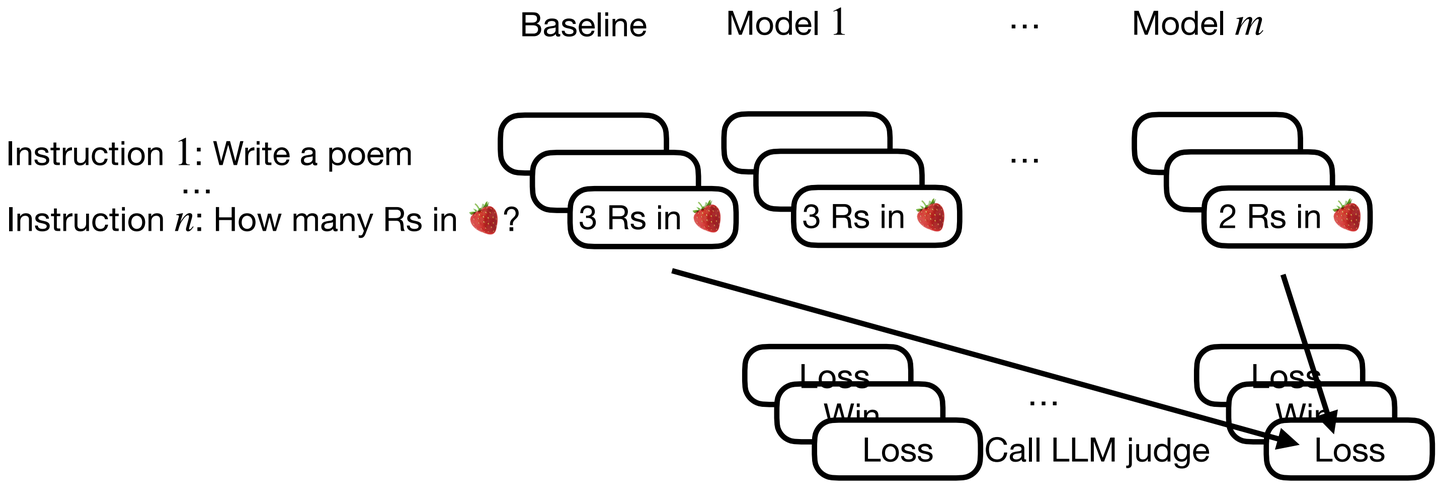
LLM judge



LLM as a judge

Grid evaluations for leaderboards

- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge
- Report win rate and rank models



Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

LLM as a judge

Grid evaluations for leaderboards

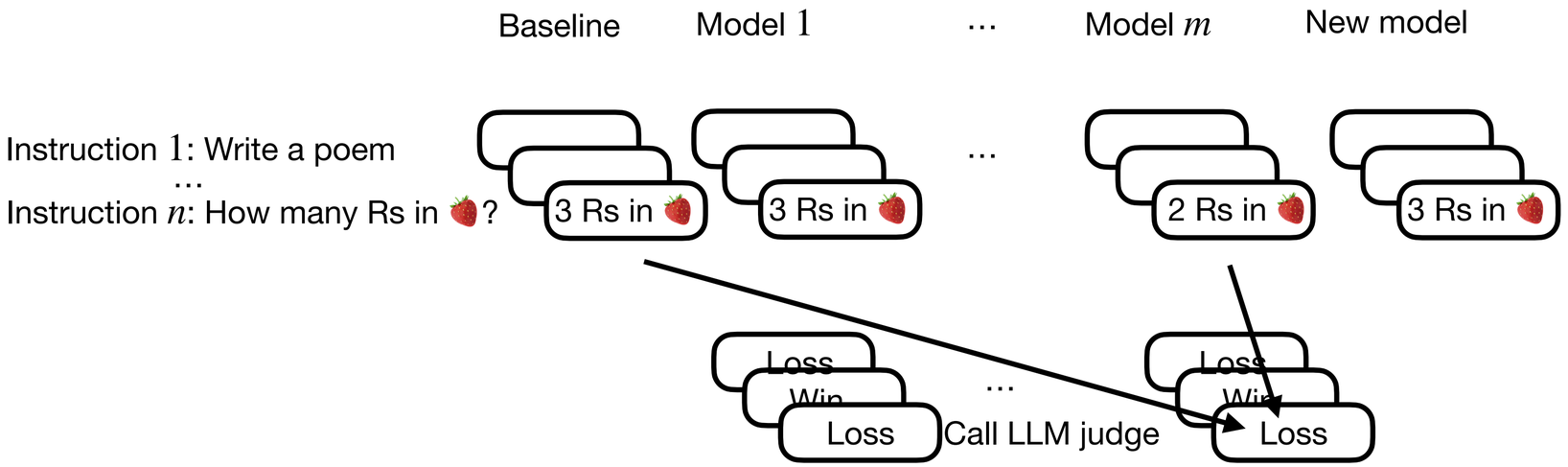
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge
- Report win rate and rank models

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

LLM as a judge

Grid evaluations for leaderboards

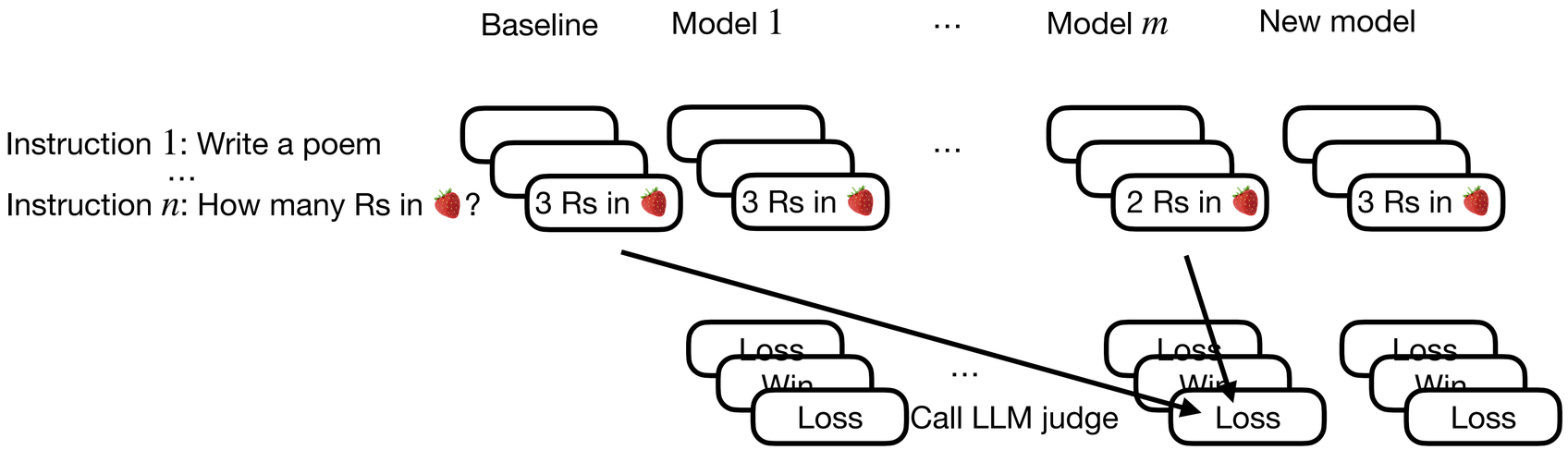
- To generate rankings, Alpaca-Eval, Arena-Hard, m-Arena-Hard evaluate models on a grid
- m models to rank plus a baseline
- n instructions to evaluate (eg n = 1000)
- Evaluate completions on all models $n \times (m + 1)$ completions to generate
- Judge each model against the baseline: $n \times m$ calls to an LLM judge
- Report win rate and rank models

Given this instruction and the two models outputs, tell me which one is better?

Instruction: How many Rs in strawberry?
Model 1: 3 Rs in 🍓. strawberry
Model 2: There are two r in strawberry.

→ **Model 1**

LLM judge



Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

LLM as a judge

Current challenges

LLM as a judge

Current challenges

- Pros:

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position
  - Preference for verbose outputs & superficial style

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position
  - Preference for verbose outputs & superficial style
  - Bias of the judge model (GPT4 tends to prefer GPT4) but generalized use of close model as the judge

LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position
  - Preference for verbose outputs & superficial style
  - Bias of the judge model (GPT4 tends to prefer GPT4) but generalized use of close model as the judge
  - Meta-evaluation (how judge quality is evaluated) often hard & problematic

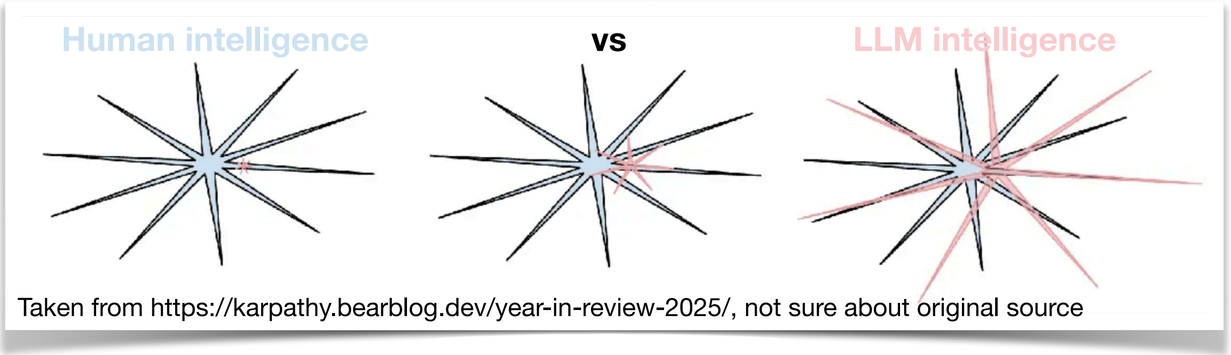
LLM as a judge

Current challenges

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position
  - Preference for verbose outputs & superficial style
  - Bias of the judge model (GPT4 tends to prefer GPT4) but generalized use of close model as the judge
  - Meta-evaluation (how judge quality is evaluated) often hard & problematic
- Results of LLM judges and human are highly correlated but **fundamentally different**

LLM as a judge

Current challenges



Taken from https://karpathy.bearblog.dev/year-in-review-2025/, not sure about original source

- Pros:
  - Much cheaper than human annotations
  - Very fast
  - Easily customizable
- Challenges:
  - Bias of the position: judges have small preference for first position
  - Preference for verbose outputs & superficial style
  - Bias of the judge model (GPT4 tends to prefer GPT4) but generalized use of close model as the judge
  - Meta-evaluation (how judge quality is evaluated) often hard & problematic
- Results of LLM judges and human are highly correlated but **fundamentally different**

Bias of the judge model

Favoring your kind

LLM Evaluators Recognize and Favor Their Own Generations

Arjun Panickssery[1] **Samuel R. Bowman**[2] **Shi Feng**[3]
[1]MATS [2]New York University, Anthropic PBC [3]George Washington University
arjun.panickssery@gmail.com

Bias of the judge model

Favoring your kind

1. LLMs can recognize their predictions among predictions of other LLMs
2. LLMs prefer their own predictions rather than the ones of other LLMs in generative tasks such as summarization

---

LLM Evaluators Recognize and Favor Their Own Generations

Arjun Panickssery[1] **Samuel R. Bowman**[2] **Shi Feng**[3]
[1]MATS [2]New York University, Anthropic PBC [3]George Washington University
arjun.panickssery@gmail.com

Bias of the judge model

A striking example

README Apache-2.0 license

Arena-Hard-Auto

Arena-Hard arXiv Arena-Hard Arena-Hard LMArena-ai

News • Leaderboard • Install • Evaluation • Demo • Citation

News

- **[Apr 23, 2025]** 🎉 **Arena-Hard-v2.0** is finally here! Better judges, new hard prompts, and additional eval for creative writing.
- **[Oct 14, 2024]** 🎉 **Style Control** is now supported in Arena-Hard-Auto.

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, and GPT-4.1 as Judge **(If prefer OpenAI API)**

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

News • Leaderboard • Install • Evaluation • Demo • Citation

News

- **[Apr 23, 2025]** 🎉 **Arena-Hard-v2.0** is finally here! Better judges, new hard prompts, and additional eval for creative writing.
- **[Oct 14, 2024]** 🎉 **Style Control** is now supported in Arena-Hard-Auto.

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, and GPT-4.1 as Judge **(If prefer OpenAI API)**

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

News • Leaderboard • Install • Evaluation • Demo • Citation

News

- **[Apr 23, 2025]** 🎉 **Arena-Hard-v2.0** is finally here! Better judges, new hard prompts, and additional eval for creative writing.
- **[Oct 14, 2024]** 🎉 **Style Control** is now supported in Arena-Hard-Auto.

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, and GPT-4.1 as Judge **(If prefer OpenAI API)**

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

News • Leaderboard • Install • Evaluation • Demo • Citation

News

- **[Apr 23, 2025]** 🎉 **Arena-Hard-v2.0** is finally here! Better judges, new hard prompts, and additional eval for creative writing.
- **[Oct 14, 2024]** 🎉 **Style Control** is now supported in Arena-Hard-Auto.

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, and GPT-4.1 as Judge **(If prefer OpenAI API)**

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

The win rate of gemini-2.5 goes from 79% to 49.1% when not judging itself (!!)

creative writing.

- **[Oct 14, 2024]** 🎉 **Style Control** is now supported in Arena-Hard-Auto.

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, and GPT-4.1 as Judge **(If prefer OpenAI API)**

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

The win rate of gemini-2.5 goes from 79% to 49.1% when not judging itself (!!)

Anything *else* catches up your attention? 🤔

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, ar

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

The win rate of gemini-2.5 goes from 79% to 49.1% when not judging itself (!!)

Anything *else* catches up your attention? 🤔

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, ar

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

The win rate of gemini-2.5 goes from 79% to 49.1% when not judging itself (!!)

Anything *else* catches up your attention? 🤔

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

Hard Prompt, Style Control, ar

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Bias of the judge model

A striking example

README Apache-2.0 license

Anything catches up your attention? 🤔

The win rate of gemini-2.5 goes from 79% to 49.1% when not judging itself (!!)

Anything *else* catches up your attention? 🤔

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct		
27	OpenThinker2-32B		

Hard Prompt, Style Control, ar

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	87.0	(-1.0 / +1.0)
1	o4-mini-2025-04-16-high	81.7	(-1.2 / +1.2)
2	o4-mini-2025-04-16	78.0	(-1.3 / +1.4)
3	o3-mini-2025-01-31-high	64.8	(-2.1 / +1.9)
4	o1-2024-12-17-high	58.7	(-2.3 / +2.1)
5	gpt-4.1	58.3	(-2.0 / +2.3)
6	o1-2024-12-17	50.2	(-2.2 / +1.8)
7	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
8	gemini-2.5	49.1	(-2.5 / +2.4)
9	gpt-4.1-mini	48.6	(-2.7 / +1.9)
10	deepseek-r1	48.0	(-2.6 / +2.3)
11	claude-3-7-sonnet-20250219-thinking-16k	47.0	(-1.9 / +2.3)
12	Qwen3-235B-A22B	46.7	(-1.9 / +2.4)
13	gemini-2.5-flash	45.1	(-2.7 / +2.1)
14	gpt-4.5-preview	43.0	(-1.9 / +2.2)
15	QwQ-32B	36.1	(-2.0 / +2.2)
16	Qwen3-32B	35.8	(-2.1 / +2.2)
17	Qwen3-30B-A3B	28.7	(-1.4 / +2.1)
18	claude-3-5-sonnet-20241022	25.8	(-1.7 / +1.8)
19	s1.1-32B	18.3	(-2.3 / +2.2)
20	gpt-4.1-nano	15.4	(-1.1 / +1.2)
21	Athene-V2-Chat	12.6	(-1.2 / +1.3)
22	Qwen3-4B	12.6	(-1.1 / +1.5)
23	llama4-maverick-instruct-basic	12.0	(-1.0 / +1.2)
24	gemma-3-27b-it	9.7	(-0.9 / +1.1)
25	Qwen2.5-72B-Instruct	8.0	(-0.7 / +0.9)
26	Llama-3.1-Nemotron-70B-Instruct-HF	6.8	(-0.6 / +0.8)
27	OpenThinker2-32B	2.3	(-0.2 / +0.3)

Assuming a normal distribution, we would need 10^{197} samples to see a number as low as 49 on average with this confidence interval!

Tuning LLM judges

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1

Tuning Judge

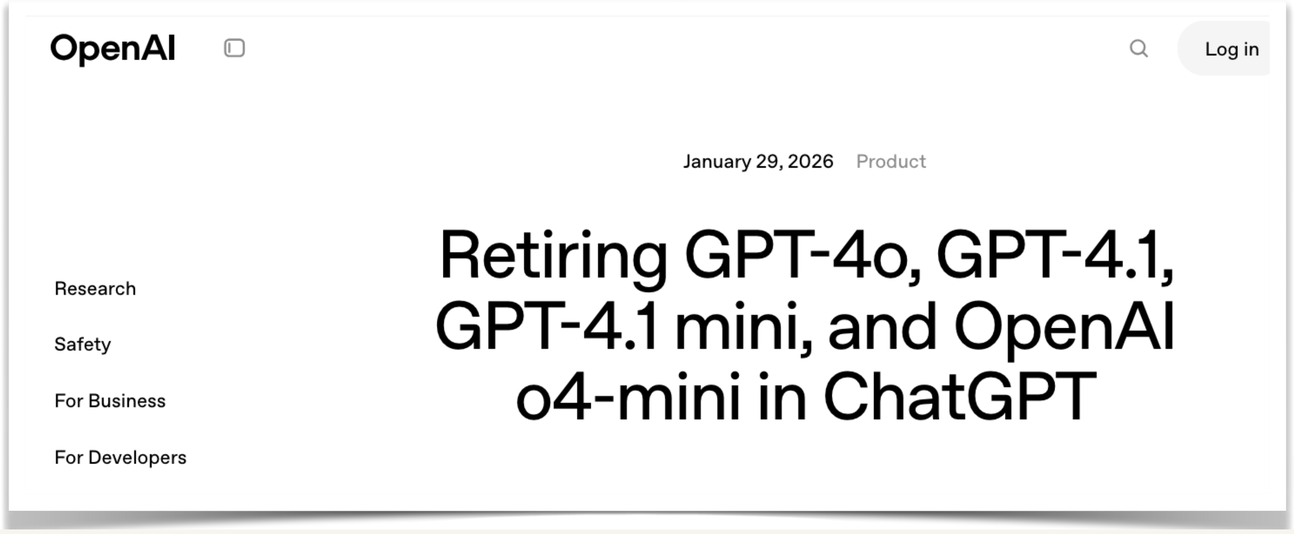
Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1
- … which where just retired

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1
- … which where just retired

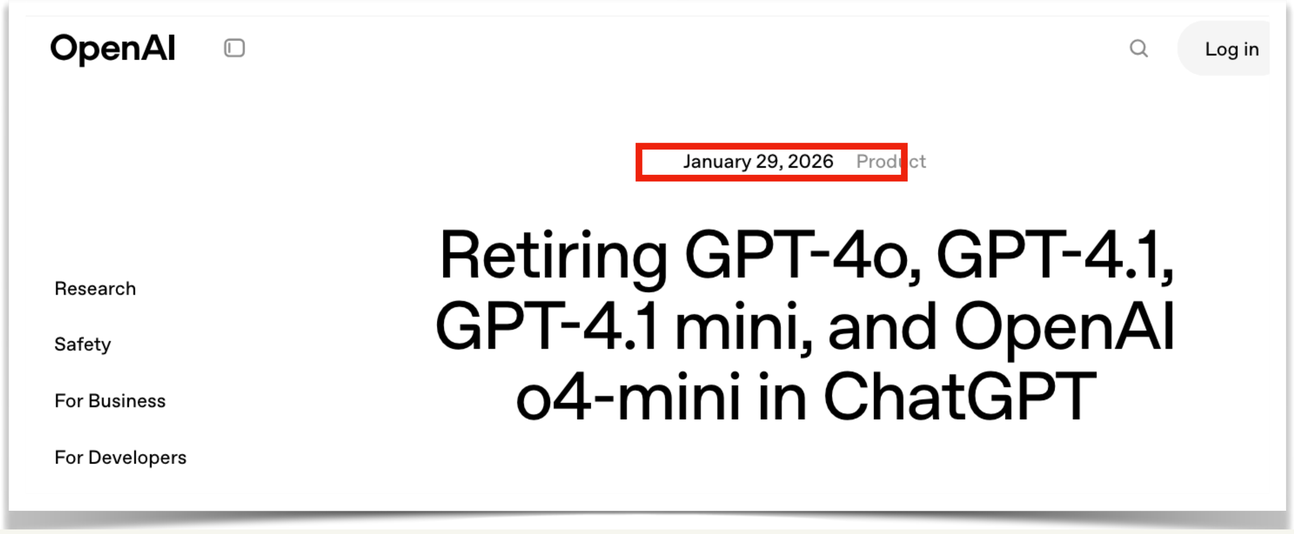


22. ▲ Retiring GPT-4o, GPT-4.1, GPT-4.1 mini, and OpenAI o4-mini in ChatGPT (openai.com)
257 points by rd 19 hours ago | hide | 329 comments

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1
- … which where just retired

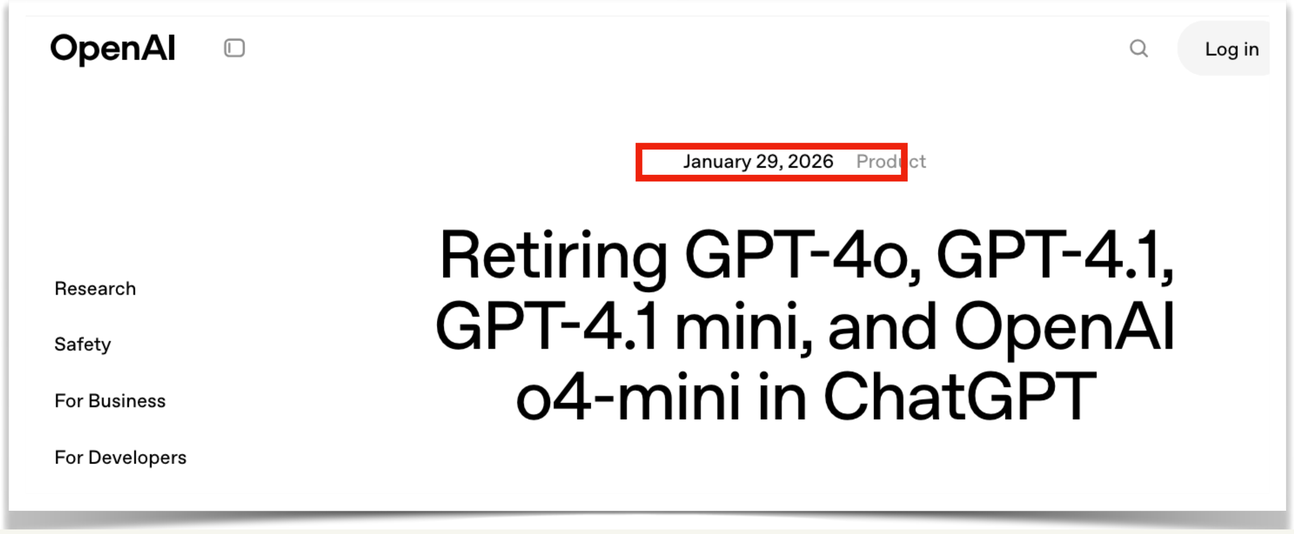


22. ▲ Retiring GPT-4o, GPT-4.1, GPT-4.1 mini, and OpenAI o4-mini in ChatGPT (openai.com)
257 points by rd 19 hours ago | hide | 329 comments

Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1
- … which where just retired
- Discuss how tuning of LLM judges allow to catch the performance gap with proprietary models

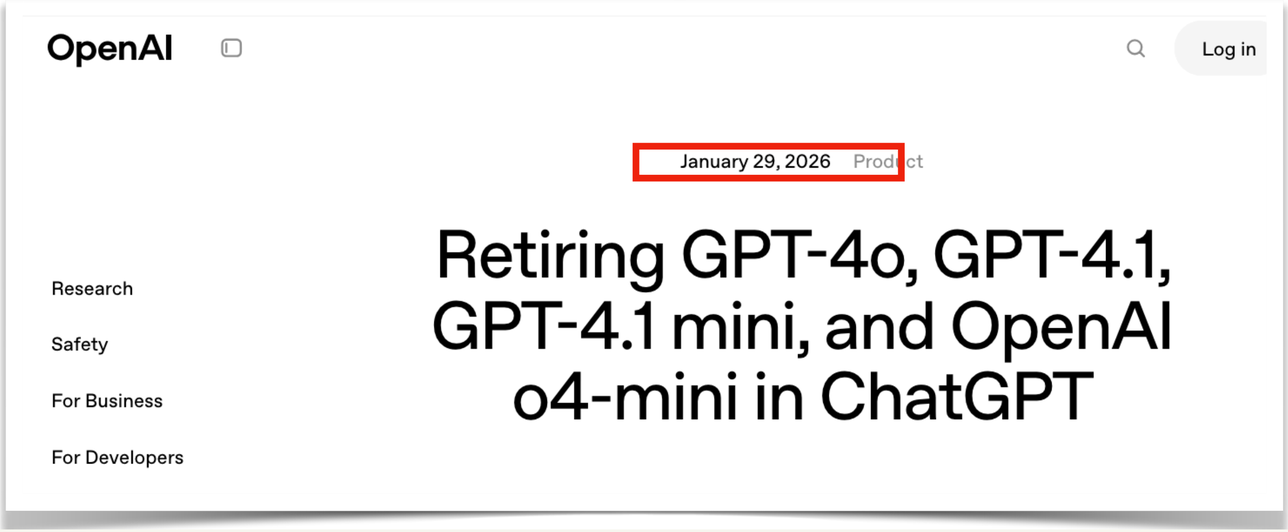


22. ▲ Retiring GPT-4o, GPT-4.1, GPT-4.1 mini, and OpenAI o4-mini in ChatGPT (openai.com)
257 points by rd 19 hours ago | hide | 329 comments

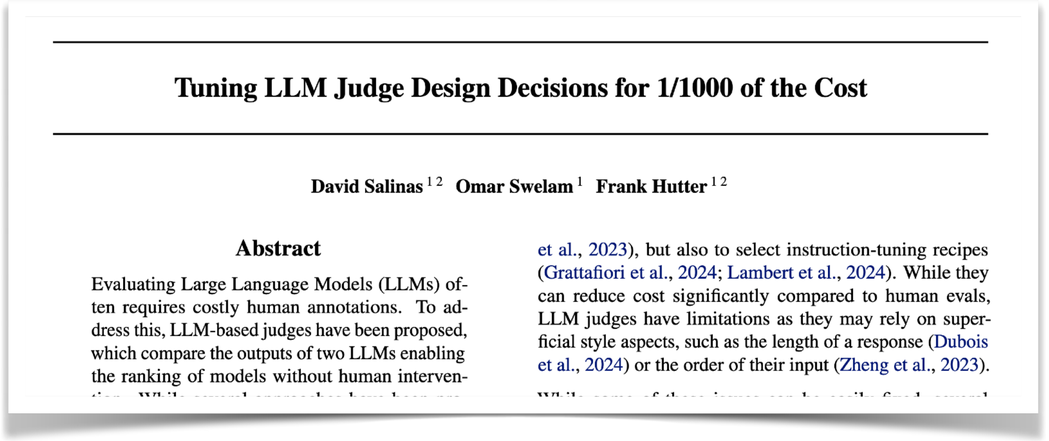
Tuning Judge

Avoiding proprietary models by tuning hyperparameters

- If the models are really biased toward themselves can we use at least open-weights models?
- Should we rely on close-models for open-science benchmarking?
- Alpaca-Eval and Arena-Hard proposed were using respectively and GPT-4 Turbo and GPT-4.1
- … which where just retired
- Discuss how tuning of LLM judges allow to catch the performance gap with proprietary models



22. ▲ Retiring GPT-4o, GPT-4.1, GPT-4.1 mini, and OpenAI o4-mini in ChatGPT (openai.com)
257 points by rd 19 hours ago | hide | 329 comments



Tuning LLM Judge

Cofounding factors

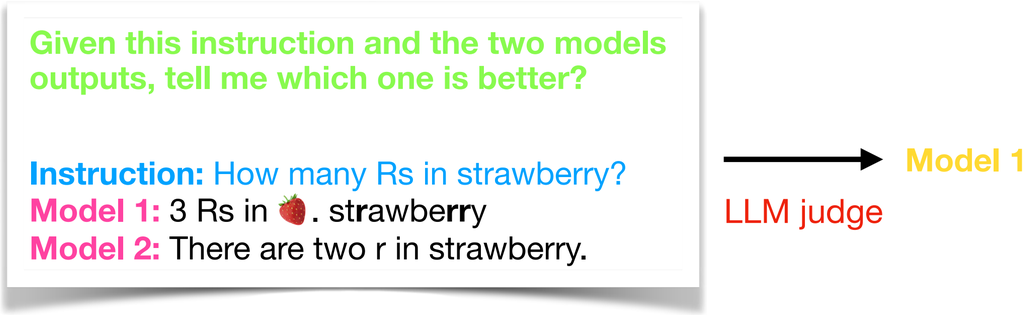
Tuning LLM Judge

Cofounding factors

- Things that were changed between Alpaca-Eval and Arena-Hard:

Tuning LLM Judge

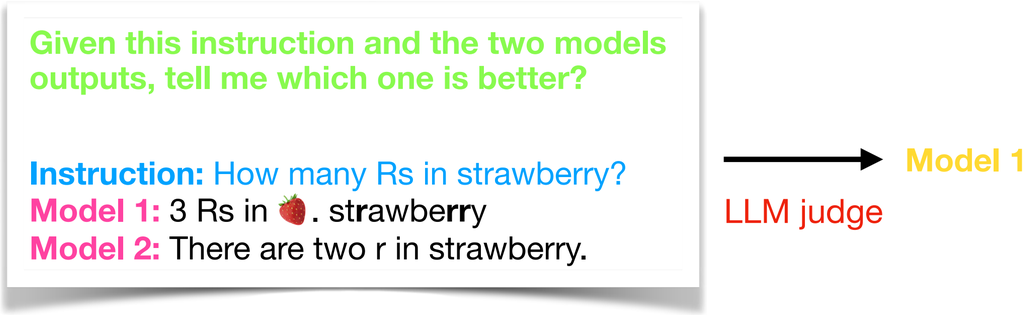
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:

Tuning LLM Judge

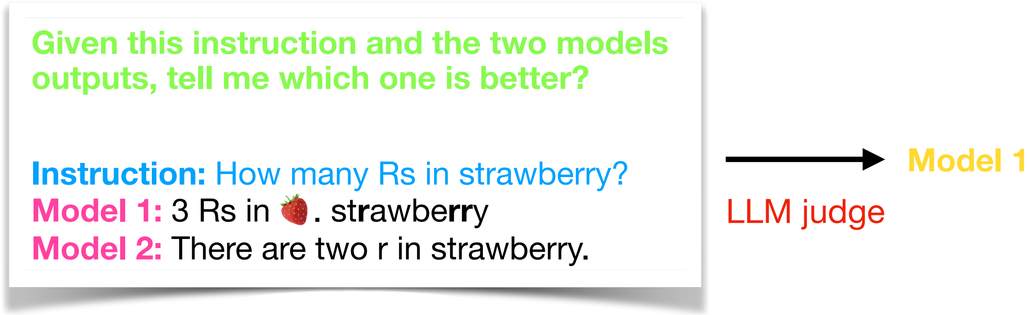
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)

Tuning LLM Judge

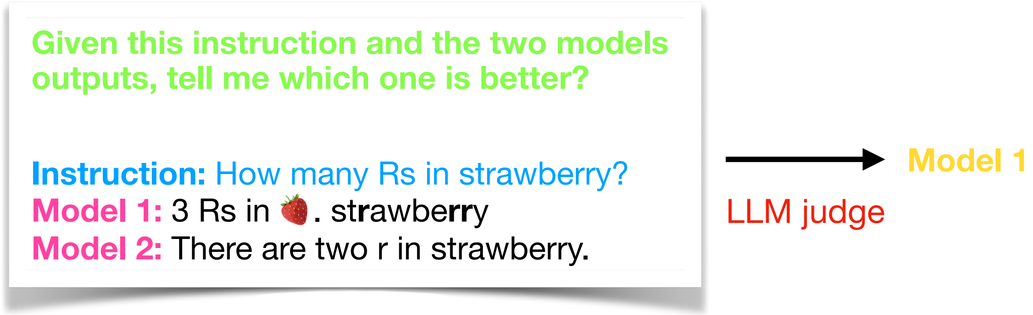
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)

Tuning LLM Judge

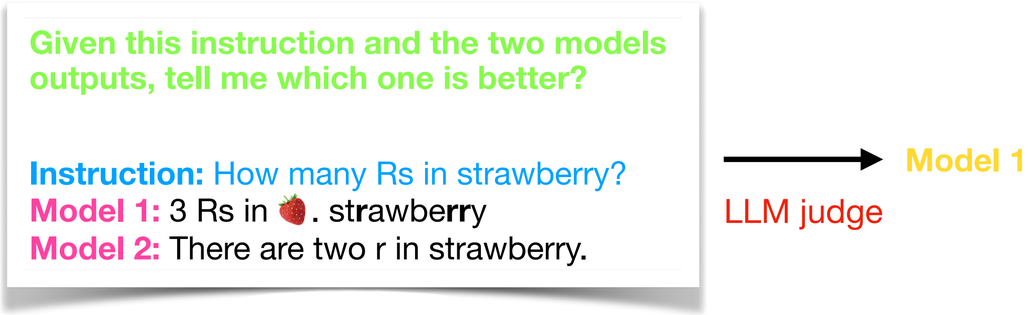
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge

Tuning LLM Judge

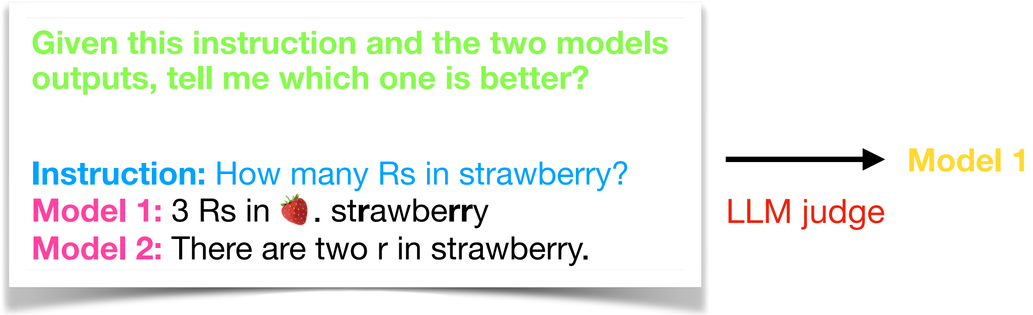
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:

Tuning LLM Judge

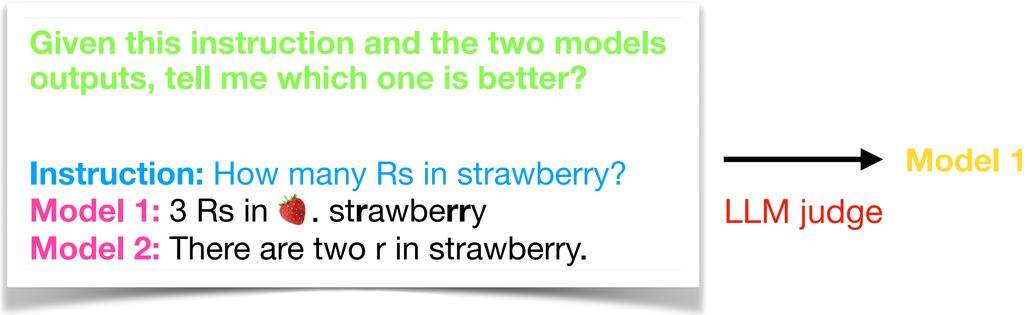
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:
    - The instructions used to evaluate the model

Tuning LLM Judge

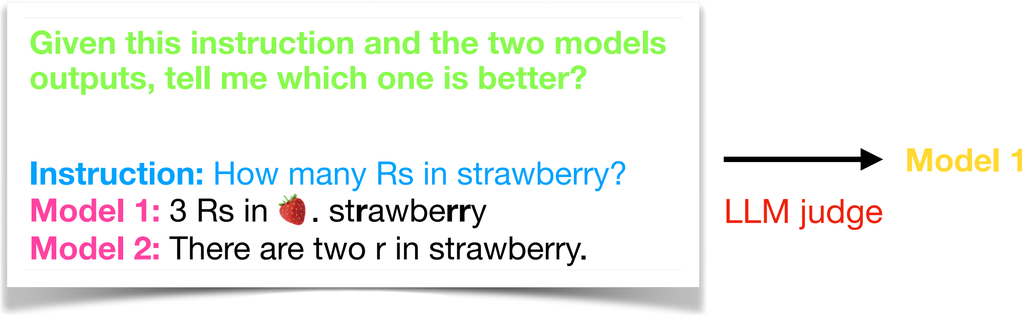
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:
    - The instructions used to evaluate the model
    - The models being evaluated on

Tuning LLM Judge

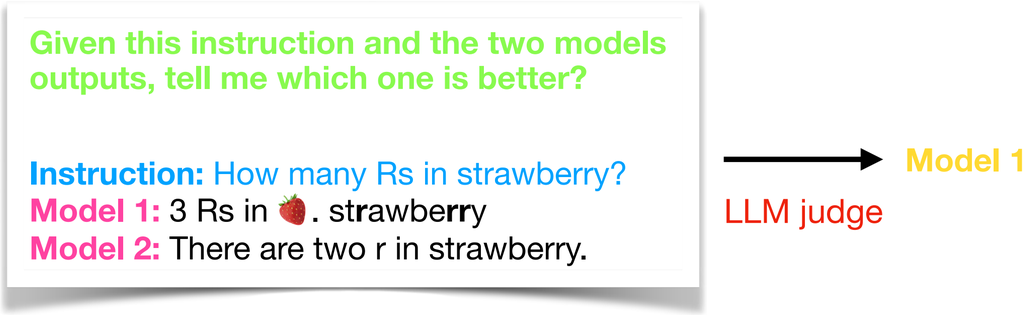
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:
    - The instructions used to evaluate the model
    - The models being evaluated on
    - Averaging the judge predictions when permuting Model 1 and Model 2

Tuning LLM Judge

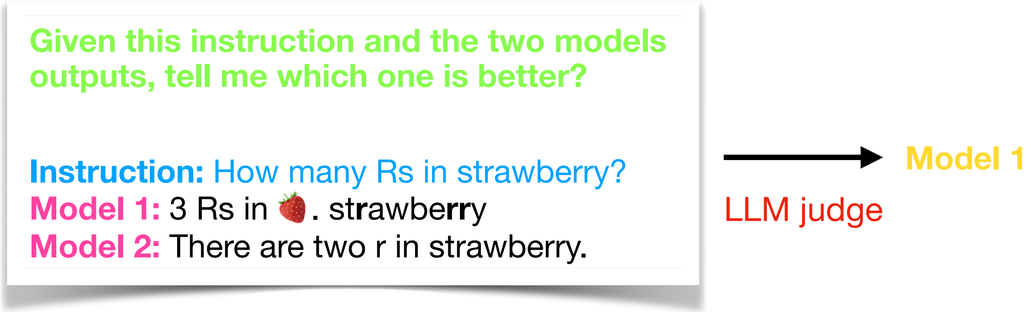
Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:
    - The instructions used to evaluate the model
    - The models being evaluated on
    - Averaging the judge predictions when permuting Model 1 and Model 2
- If we say that the Arena-Hard is better than Alpaca-Eval, *what* is exactly better?

Tuning LLM Judge

Cofounding factors



- Things that were changed between Alpaca-Eval and Arena-Hard:
  - The prompt (different formulation and ask judge to provide an answer in Arena-Hard)
  - The output format used for the judge (output letter for best model vs score in A>>B)
  - The LLM used for the judge
  - Several other aspects, for instance:
    - The instructions used to evaluate the model
    - The models being evaluated on
    - Averaging the judge predictions when permuting Model 1 and Model 2
- If we say that the Arena-Hard is better than Alpaca-Eval, *what* is exactly better?
  - The prompt? The instructions? the LLM judge model? The output format? The models being used?

Tuning LLM judges

From Cofounding factors to Search Space

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures
- Boolean to average predictions with both orders

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures
- Boolean to average predictions with both orders
- We just need to be able to evaluate the LLM judges!

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

From Cofounding factors to Search Space

- We put all cofounding factors in our search-space!
- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures
- Boolean to average predictions with both orders
- We just need to be able to evaluate the LLM judges!

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

In total we have $5 \times 2^4 = 80$ prompts and $7 \times 4 \times 80 \times 2 = 4480$ possible judges

LLM as a judge

Hyperparameter optimization

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…
- Our approach:

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…
- Our approach:
  - 1. Identify a cheaper and better metric than Spearman correlation

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Currently configurations are manually selected and evaluated…
- Our approach:
  - 1. Identify a cheaper and better metric than Spearman correlation
  - 2. Use a multifidelity-multiobjective method to tune configurations

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Evaluating LLM judges

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱

Baseline Model 1 $\cdots$ Model m

LLM as a judge

Evaluating LLM judges

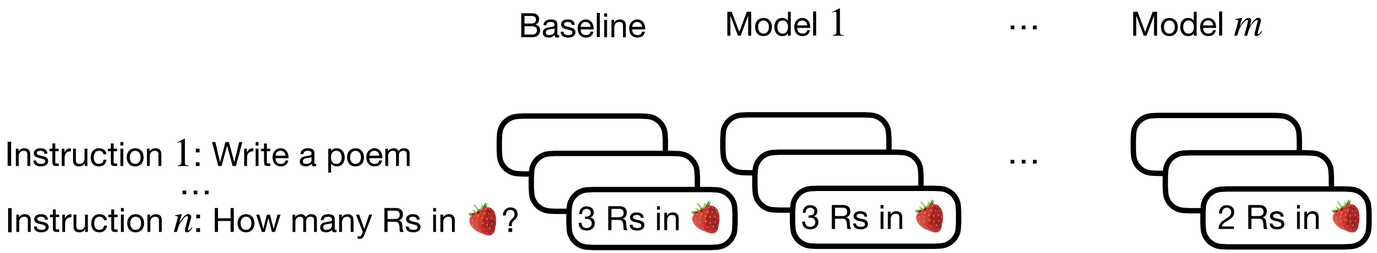
- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱

	Baseline	Model 1	⋯	Model m
Instruction 1: Write a poem				
⋯				
Instruction n: How many Rs in 🍓?				

LLM as a judge

Evaluating LLM judges

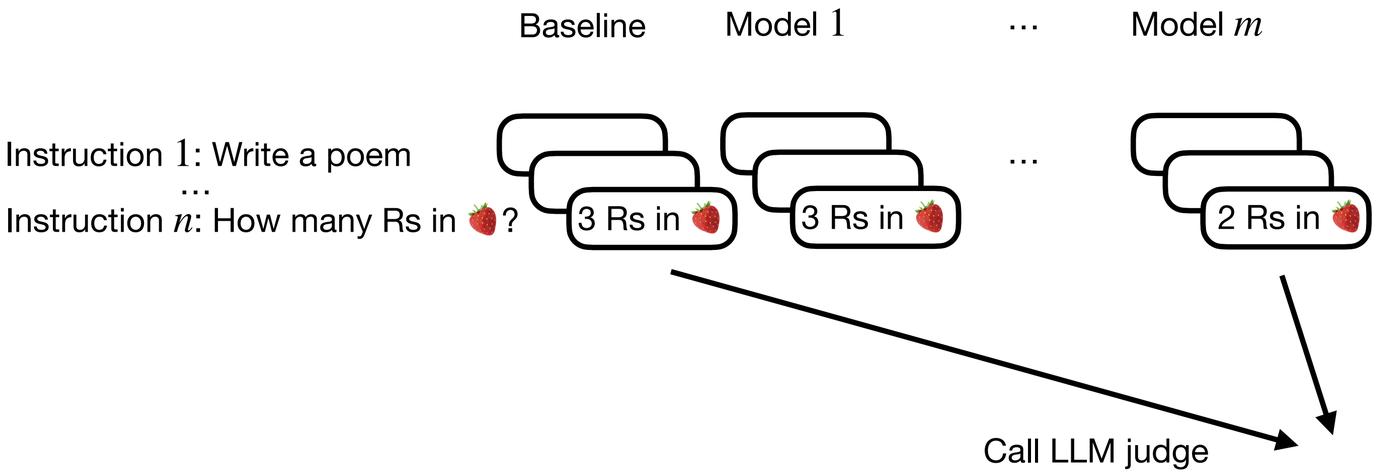
- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱



LLM as a judge

Evaluating LLM judges

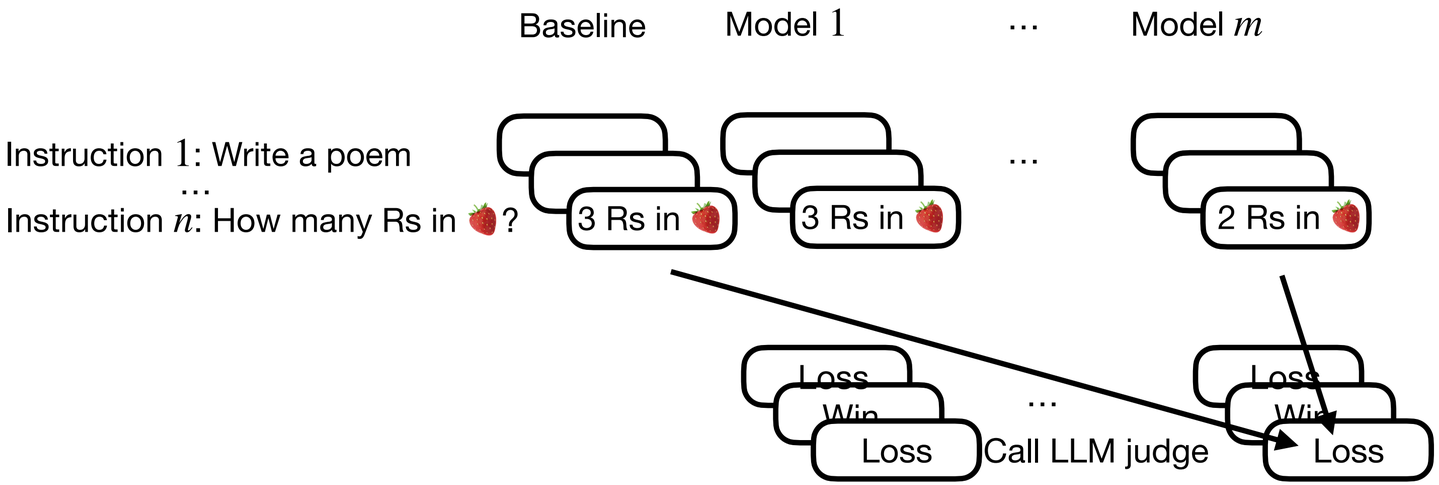
- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱



LLM as a judge

Evaluating LLM judges

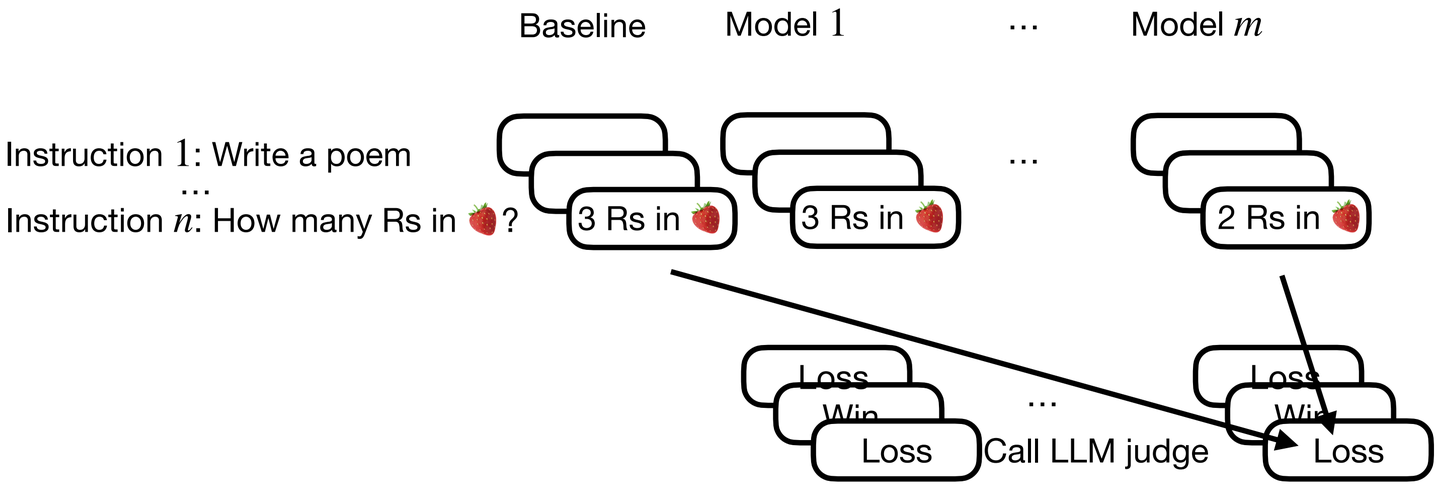
- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱



LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱



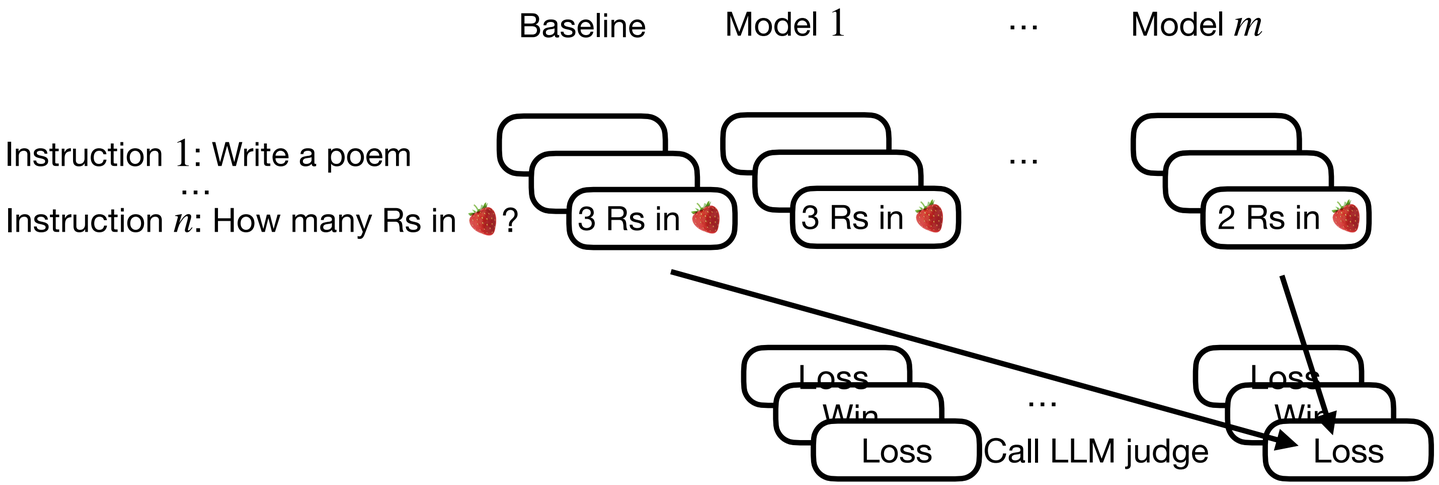
Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

LLM as a judge

Evaluating LLM judges

- How to measure the quality of an LLM judge?
- Spearman correlation with human ratings (typically from LM Arena): distance between human and LLM ratings
- High cost, needs to annotate $n \times m$ battles with a judge (costs ~$400 / judge configuration)
- Evaluating with Spearman correlation all our 4480 choices in brute force would cost $2M 😱



Performance of some judge hyperparameters for Alpaca Eval

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Hard Prompt, Style Control, and Gemini-2.5 as Judge **(Official Configuration)**:

	Model	Scores (%)	CI (%)
0	o3-2025-04-16	85.9	(-0.8 / +0.9)
1	o4-mini-2025-04-16-high	79.1	(-1.4 / +1.2)
2	gemini-2.5	79.0	(-2.1 / +1.8)
3	o4-mini-2025-04-16	74.6	(-1.8 / +1.6)
4	gemini-2.5-flash	68.6	(-1.6 / +1.6)
5	o3-mini-2025-01-31-high	66.1	(-1.5 / +2.1)
6	o1-2024-12-17-high	61.0	(-2.0 / +2.1)
7	claude-3-7-sonnet-20250219-thinking-16k	59.8	(-2.0 / +1.8)
8	Qwen3-235B-A22B	58.4	(-1.9 / +2.1)
9	deepseek-r1	58.0	(-2.2 / +2.0)
10	o1-2024-12-17	55.9	(-2.2 / +1.8)
11	gpt-4.5-preview	50.0	(-1.9 / +2.0)
12	o3-mini-2025-01-31	50.0	(-0.0 / +0.0)
13	gpt-4.1	50.0	(-1.9 / +1.7)
14	gpt-4.1-mini	46.9	(-2.4 / +2.1)
15	Qwen3-32B	44.5	(-2.2 / +2.1)
16	QwQ-32B	43.5	(-2.5 / +2.1)
17	Qwen3-30B-A3B	33.9	(-1.6 / +1.5)
18	claude-3-5-sonnet-20241022	33.0	(-2.3 / +1.8)
19	s1.1-32B	22.3	(-1.7 / +1.5)
20	llama4-maverick-instruct-basic	17.2	(-1.5 / +1.2)
21	Athene-V2-Chat	16.4	(-1.4 / +1.4)
22	gemma-3-27b-it	15.0	(-1.4 / +1.0)
23	Qwen3-4B	15.0	(-1.1 / +1.5)
24	gpt-4.1-nano	13.7	(-1.1 / +1.0)
25	Llama-3.1-Nemotron-70B-Instruct-HF	10.3	(-0.8 / +1.0)
26	Qwen2.5-72B-Instruct	10.1	(-0.9 / +1.3)
27	OpenThinker2-32B	3.2	(-0.3 / +0.3)

How to compare judges?

Identifying a better metric

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?

How to compare judges?

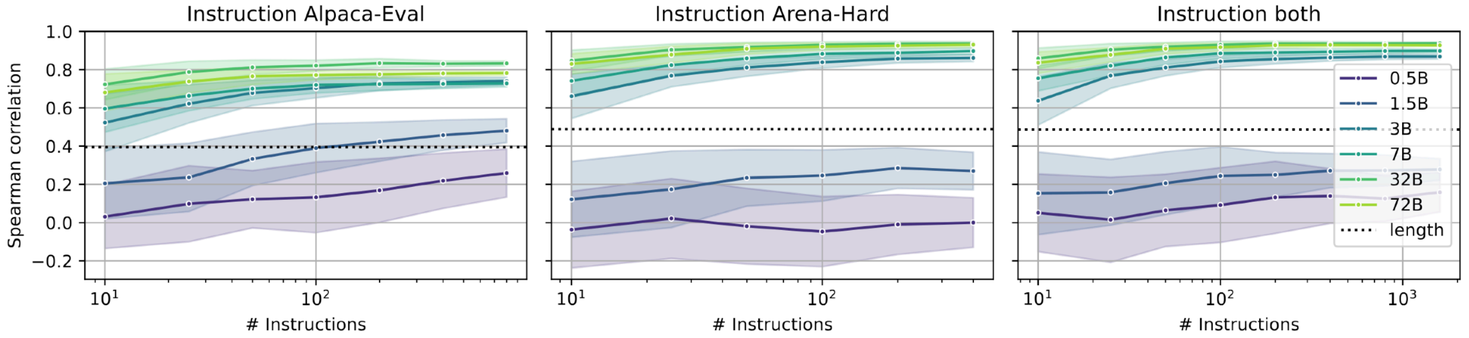
Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?

How to compare judges?

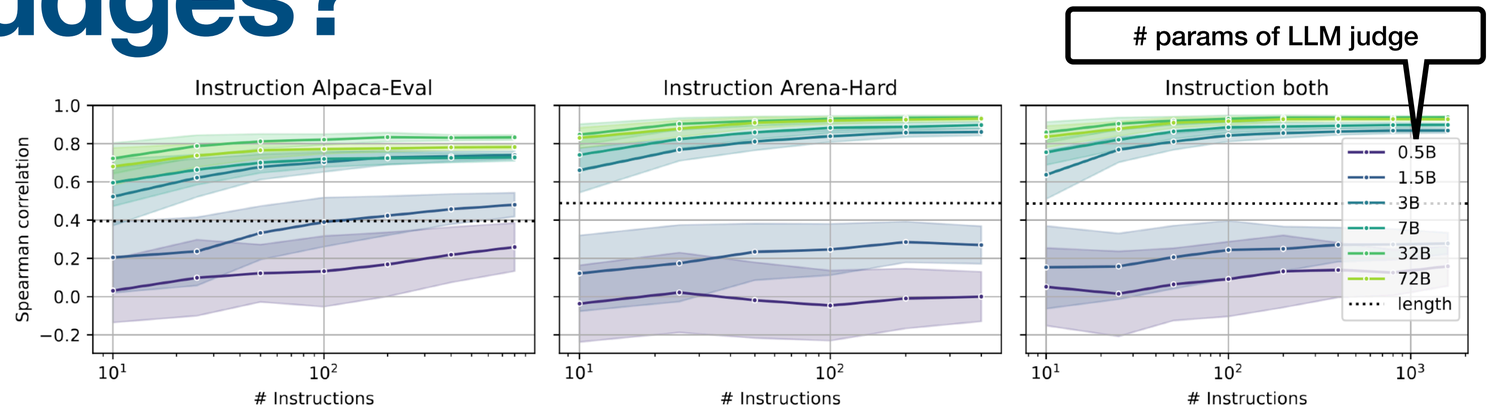
Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?



How to compare judges?

Identifying a better metric

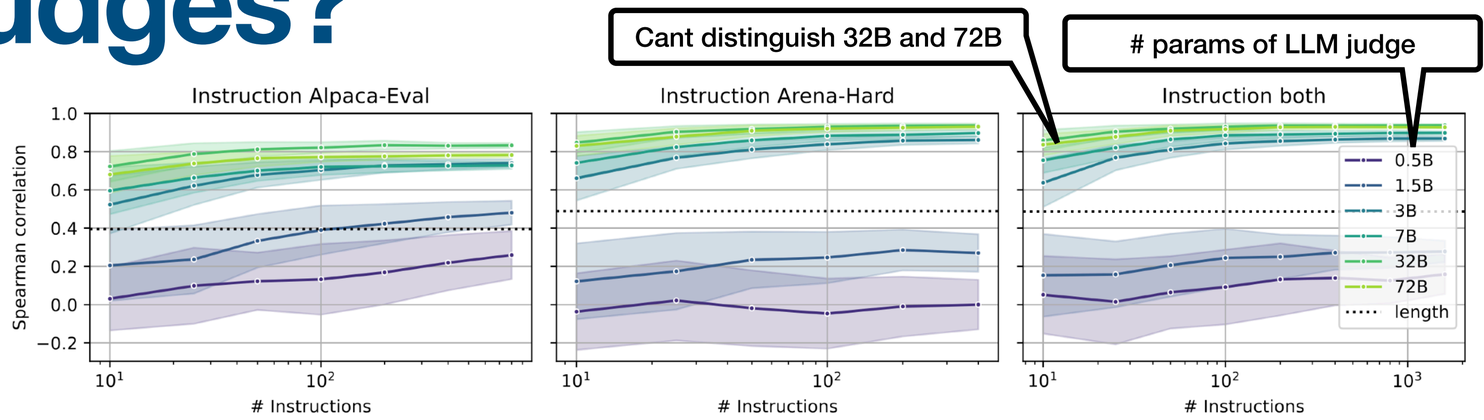


- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?

How to compare judges?

Identifying a better metric

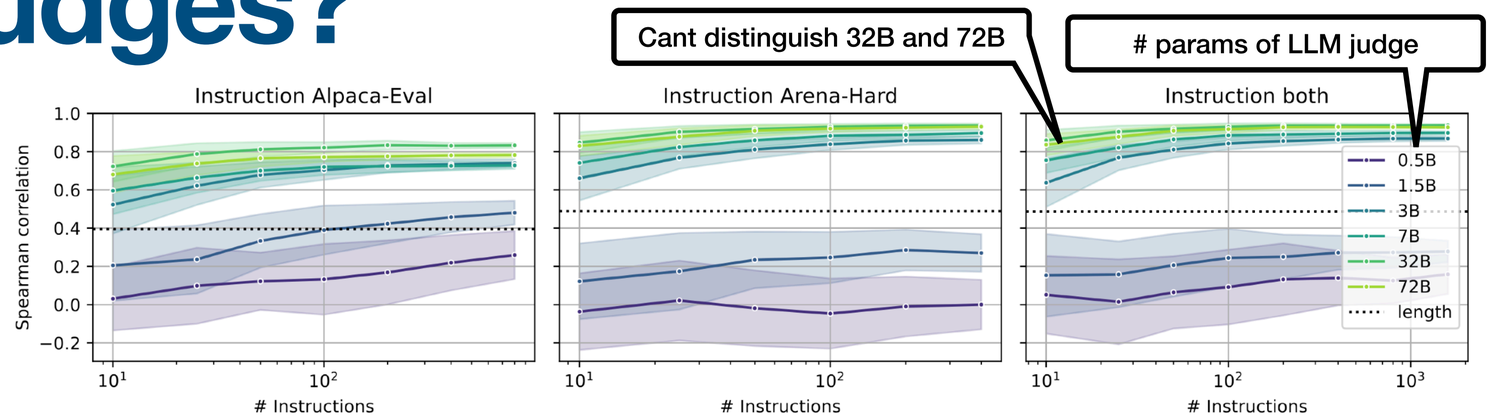
- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?

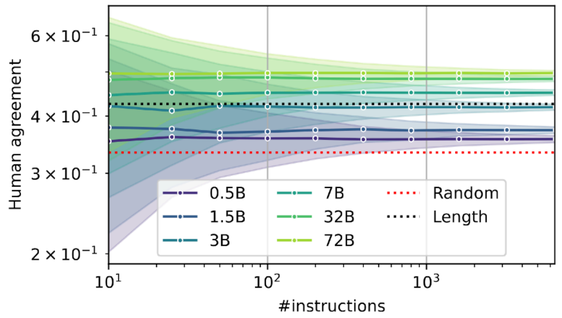


How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?

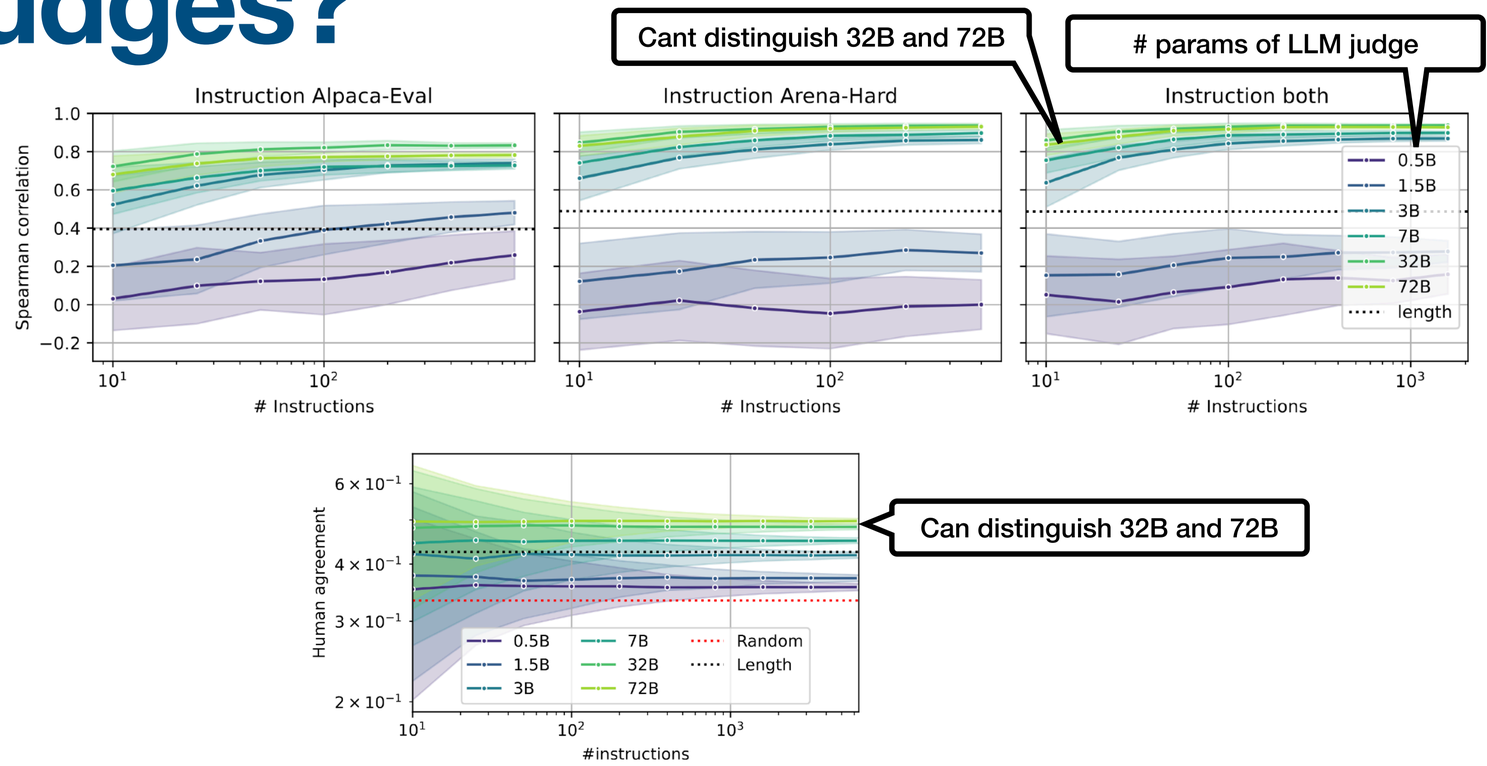




How to compare judges?

Identifying a better metric

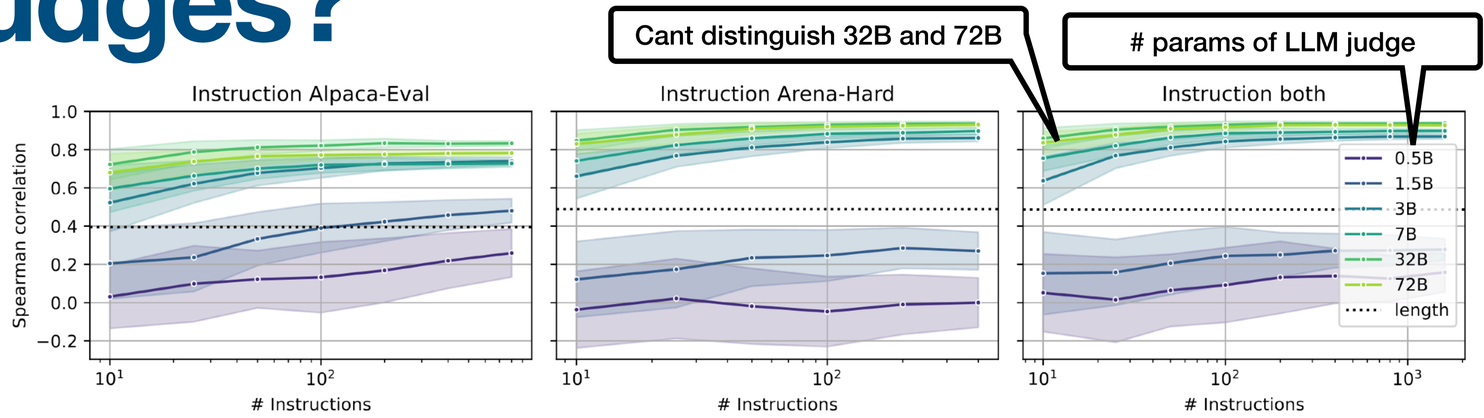
- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?

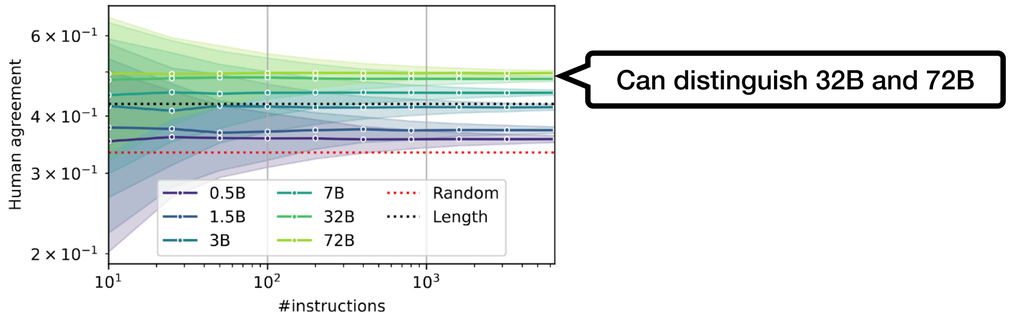


How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?





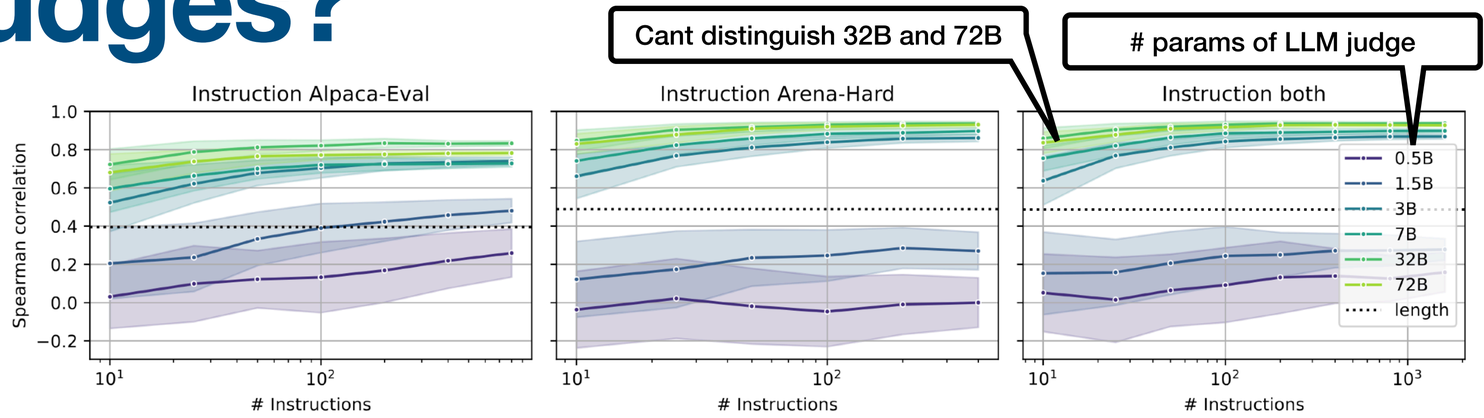
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

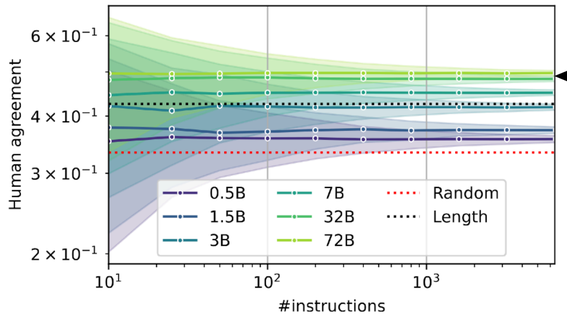
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?





Can distinguish 32B and 72B

Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

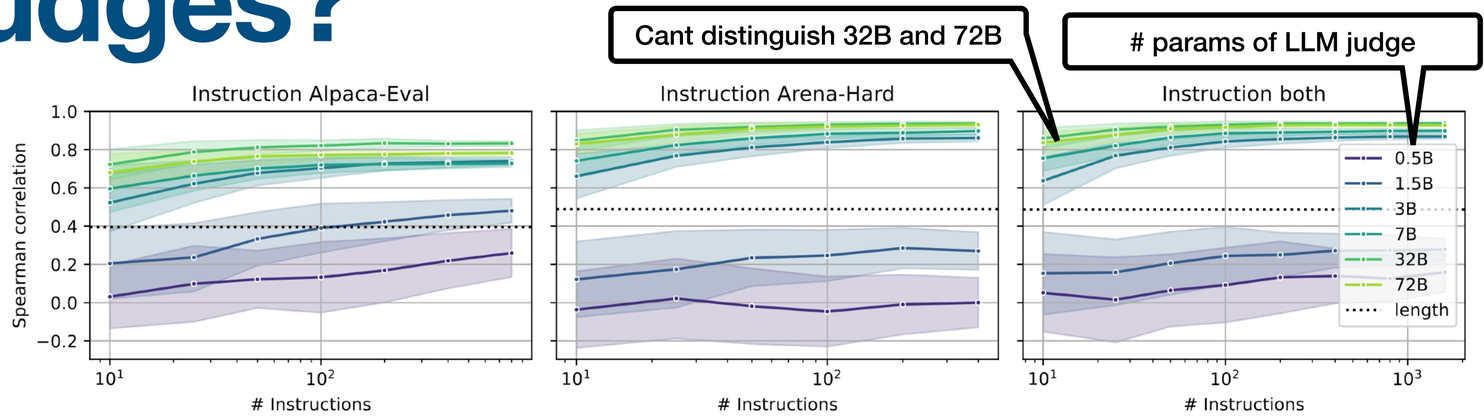
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

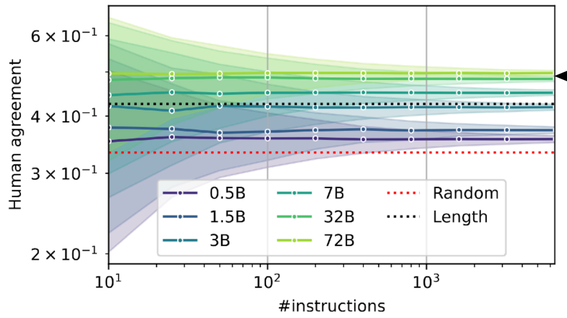
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?





Can distinguish 32B and 72B

Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

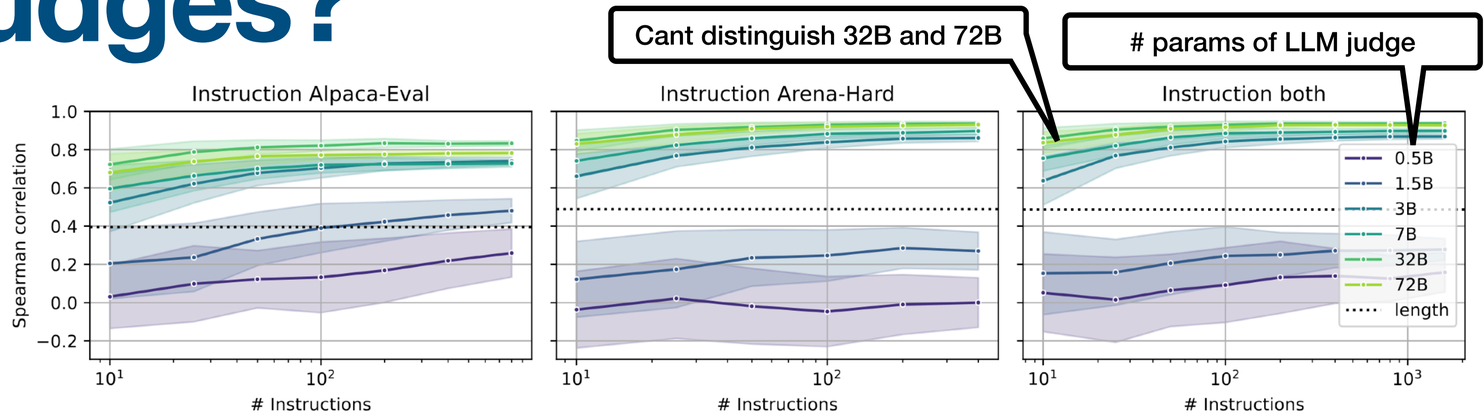
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

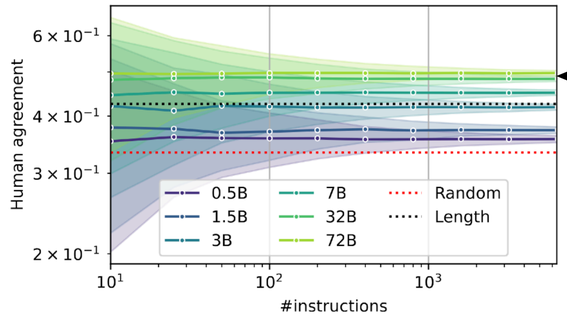
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - Evaluate m LLMs on n instructions
  - Call LLM judge to compare with baseline (GPT-4) on all annotations
  - Measure Spearman correlation with Chatbot Arena ranking and obtained winrates
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How can we say which one is better?
- How both options signal to noise ratio scale with the number of annotations?





Can distinguish 32B and 72B

Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

Human agreement has **much** better signal to noise ratio!

Tuning LLM judges

Multifidelity & multiobjective optimization

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions

Tuning LLM judges

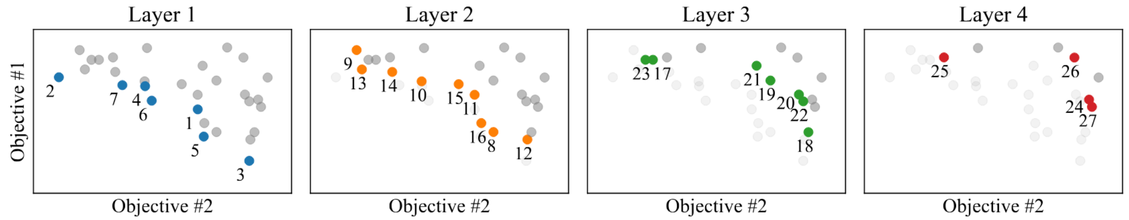
Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations



Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations

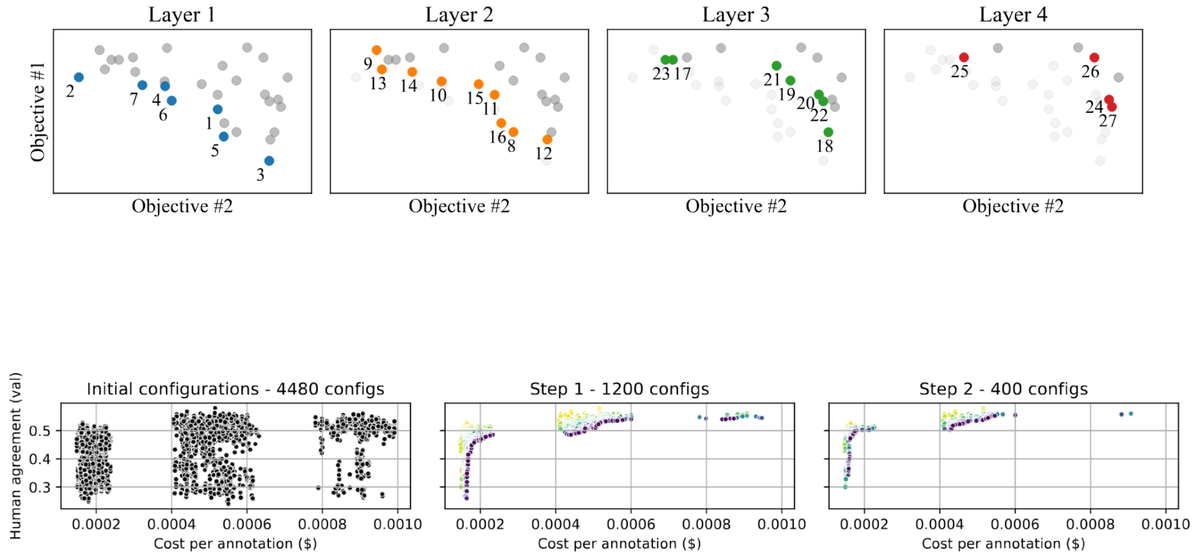


Figure 4. Illustration of the selection process. All 4 480 configurations are first evaluated on 400 instructions (left), the top 1 200 configurations are then evaluated on 1 200 instructions (center) and finally the top 400 configurations are evaluated on 3 548 instructions (right). The color denotes the ranking assigned by the non-dominated sort procedure.

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

LMsys test set

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

Judge	Human agr. (↑)
Random	0.33
Length	0.60
GPT-3.5	0.63
GPT-4	0.67
PandaLM-7B	0.60
PandaLM-70B	0.67
Ours-small	0.67
Ours-medium	0.78
Ours-large	0.76

Table 3. Comparison with PandaLM on PandaLM test set. Note that our method is not fine-tuned as opposed to PandaLM.

LMsys test set

PandaLM test set

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

LMsys test set

Judge	Human agr. (↑)
Random	0.33
Length	0.60
GPT-3.5	0.63
GPT-4	0.67
PandaLM-7B	0.60
PandaLM-70B	0.67
Ours-small	0.67
Ours-medium	0.78
Ours-large	0.76

Table 3. Comparison with PandaLM on PandaLM test set. Note that our method is not fine-tuned as opposed to PandaLM.

PandaLM test set

Judge	Sp. corr. (↑)	Cost per 1K ann. (↓)
Length	0.50 +/- 0.21	-
Arena-hard + Claude	0.82 +/- 0.12	75.0
Arena-hard + GPT4	0.90 +/- 0.06	50.0
Ours-small	0.81 +/- 0.10	0.21
Ours-medium	0.93 +/- 0.05	0.48
Ours-large	0.86 +/- 0.09	0.48

Table 4. For each judge, we compute the Spearman correlation between win-rates using the protocol of Arena-Hard and ELO-ratings computed from human annotations from Chatbot Arena. We report the mean and std over 100 boostraps of the set of models.

ArenaHard

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

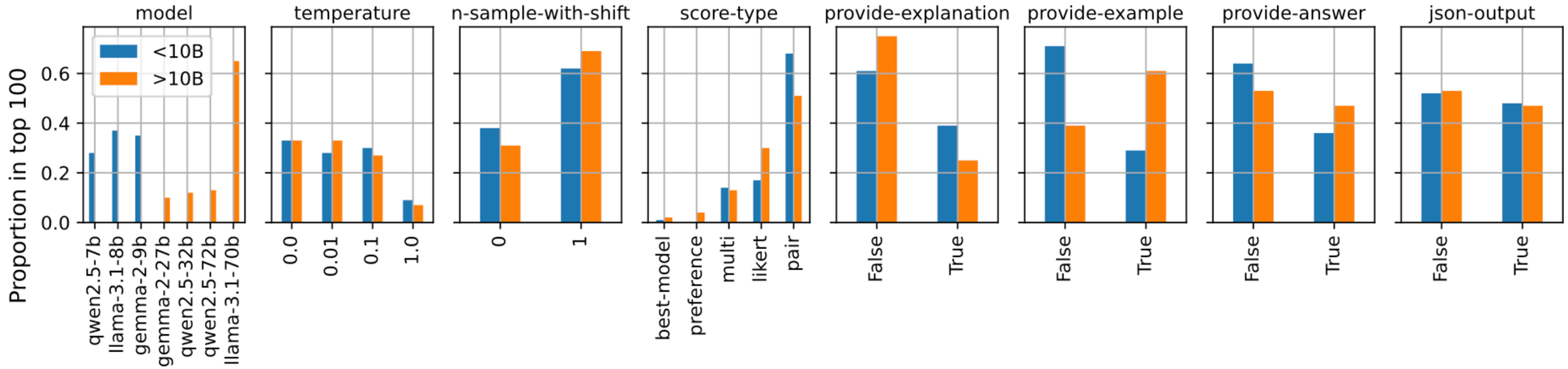


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

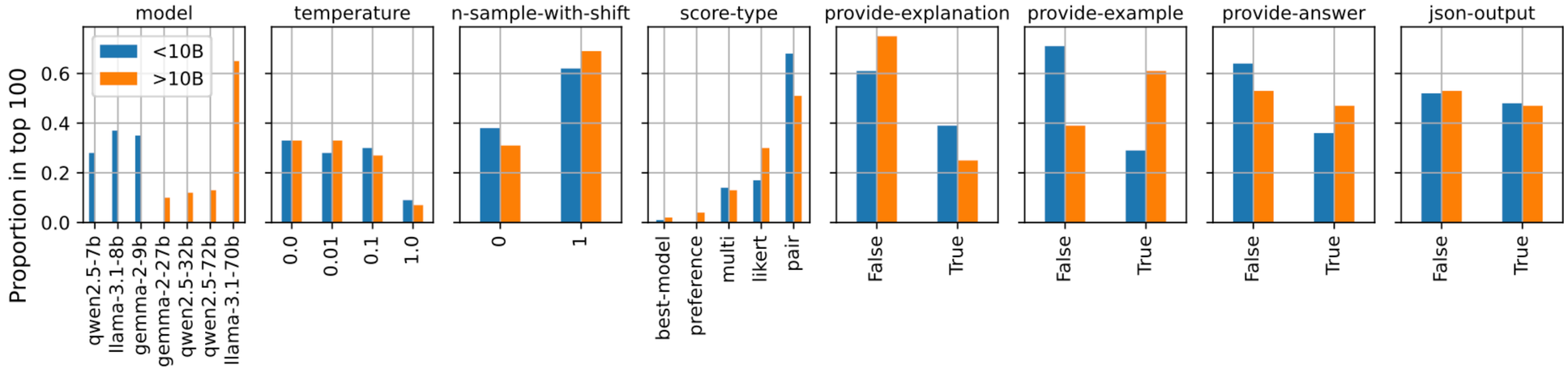


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

Format used for output is very important.

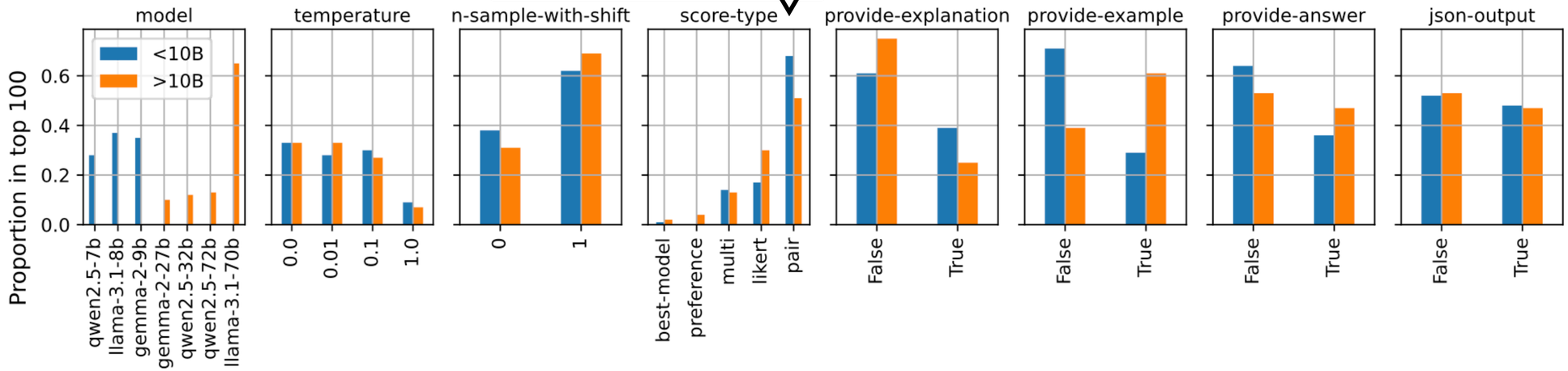


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

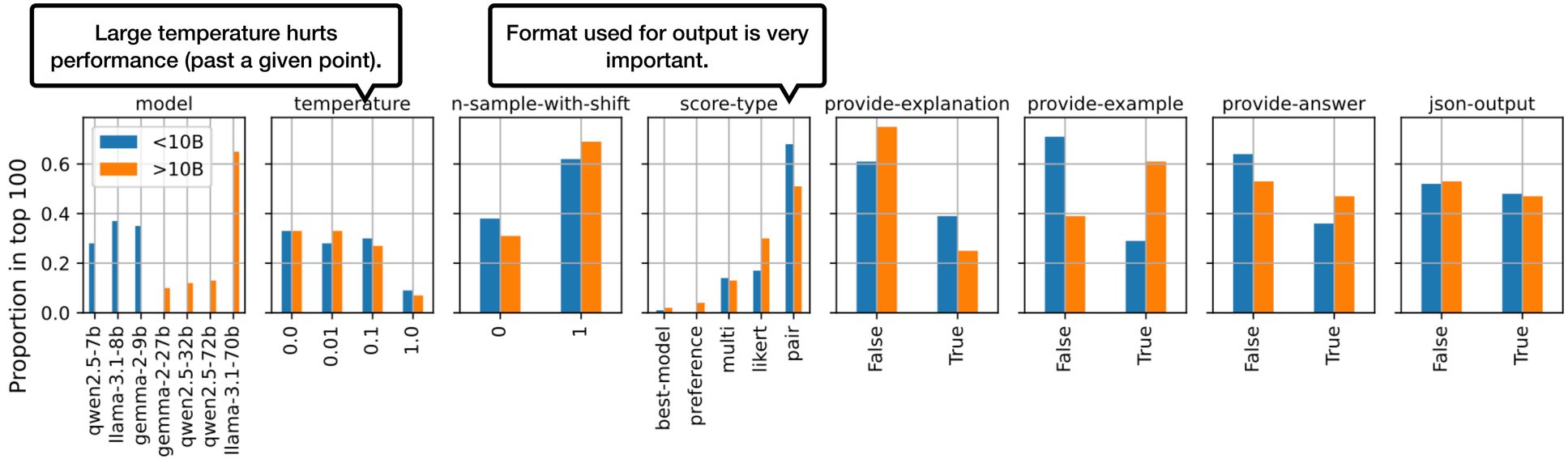


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

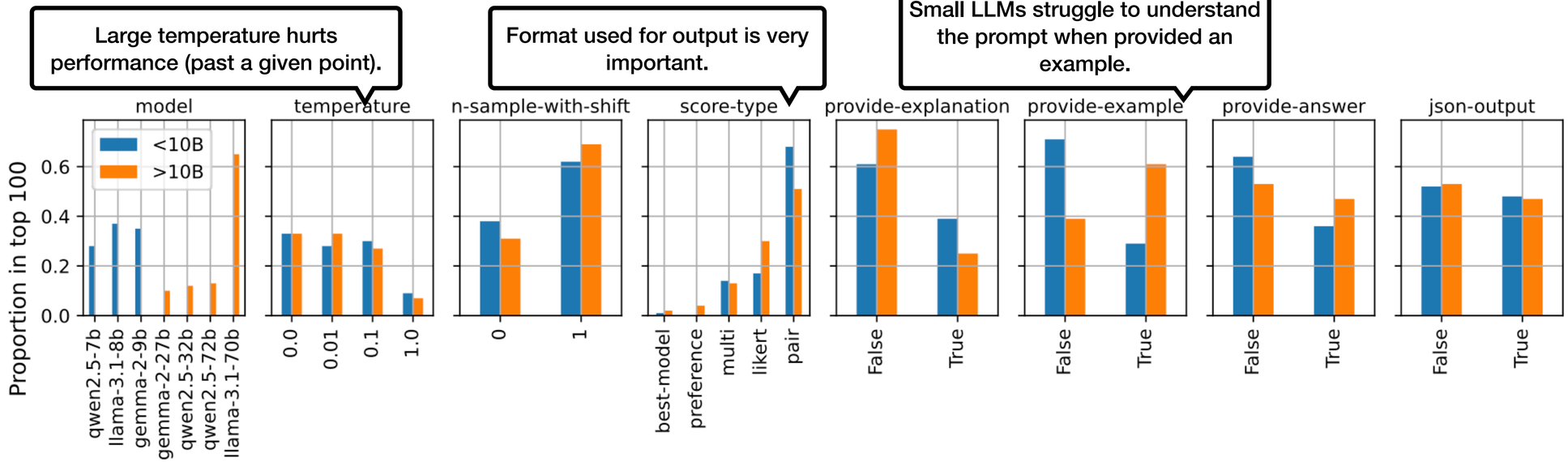


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

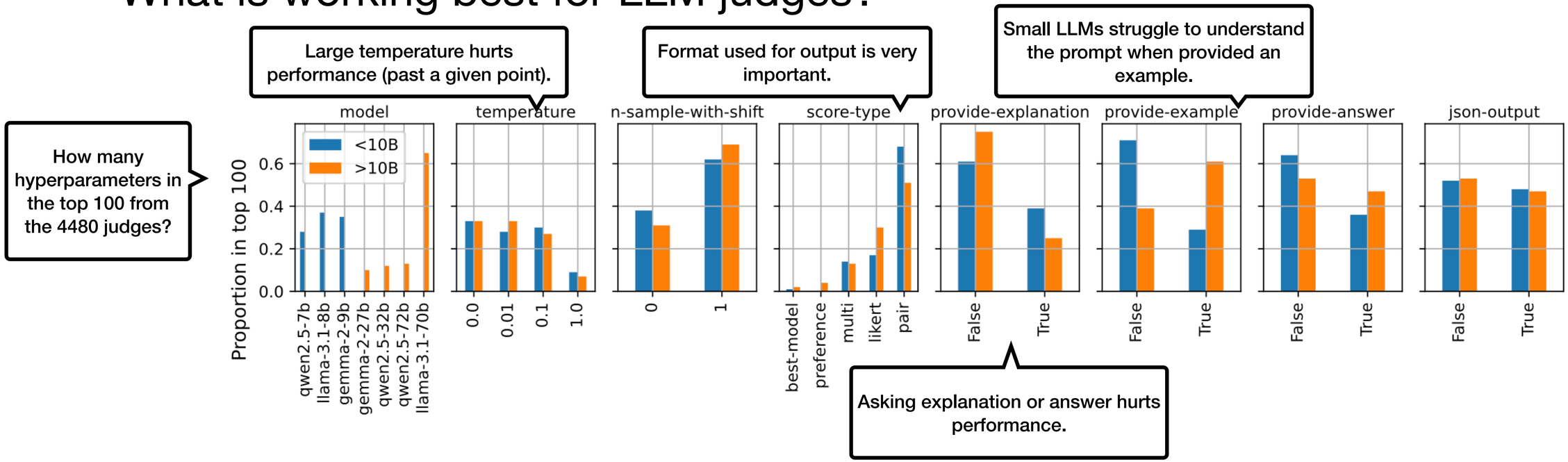


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Prompt generalization

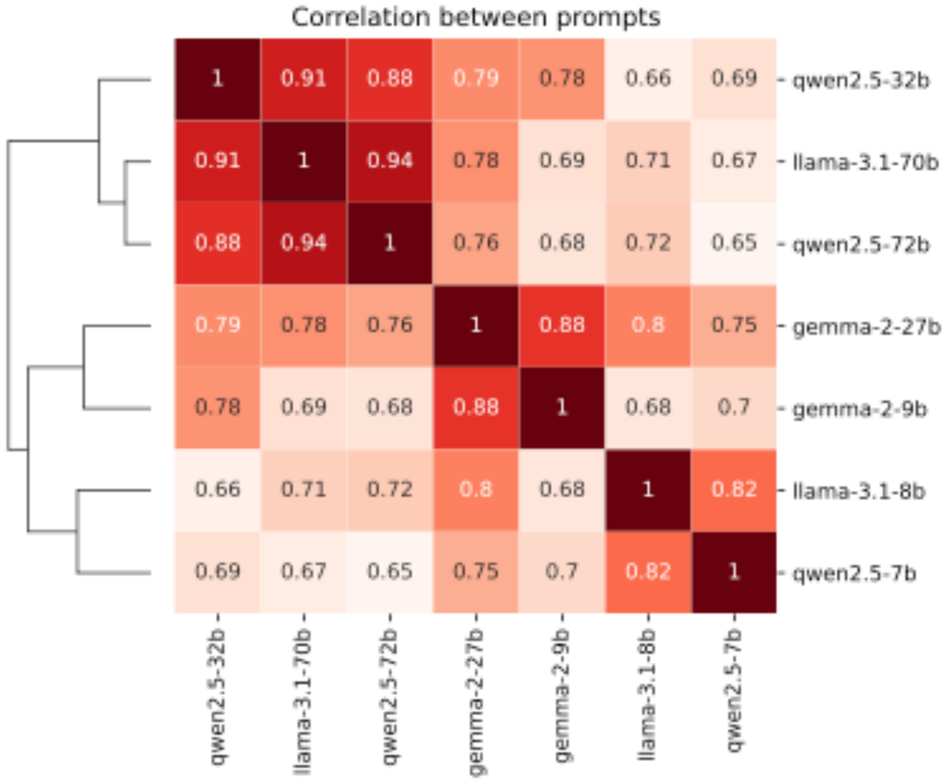
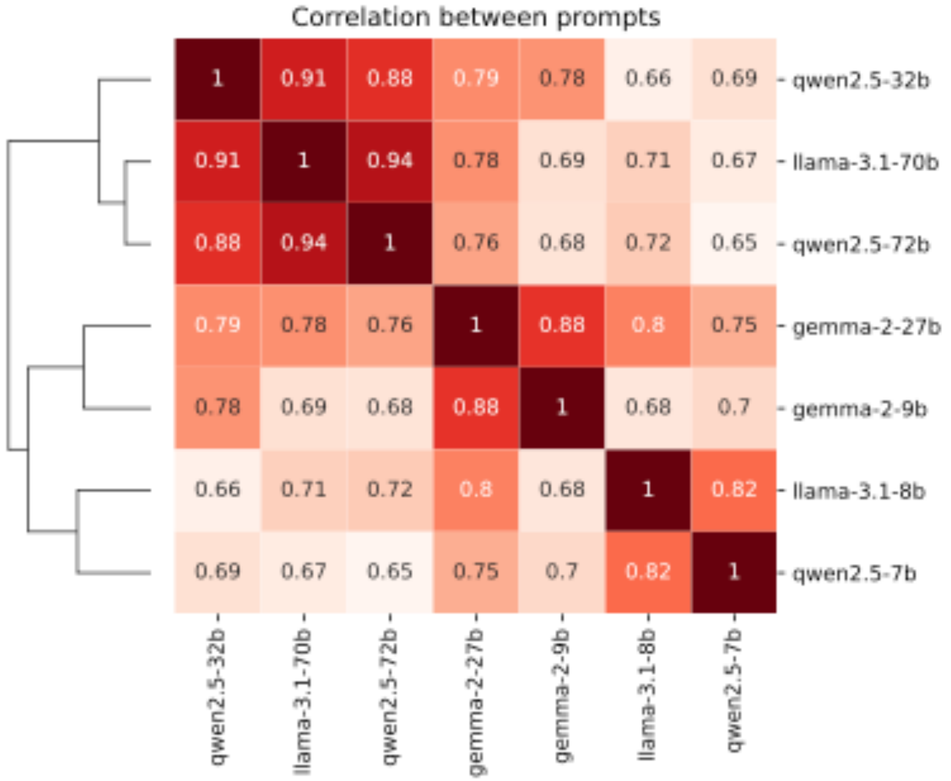


Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

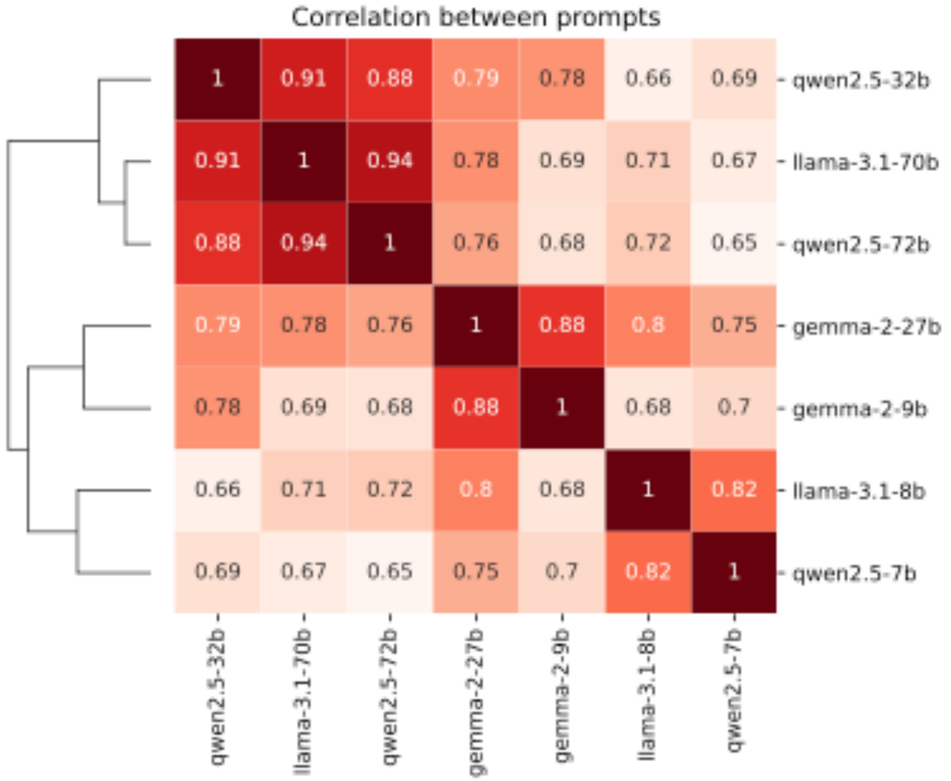
Prompt generalization



Correlation between rank of prompt configurations across models

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

Prompt generalization

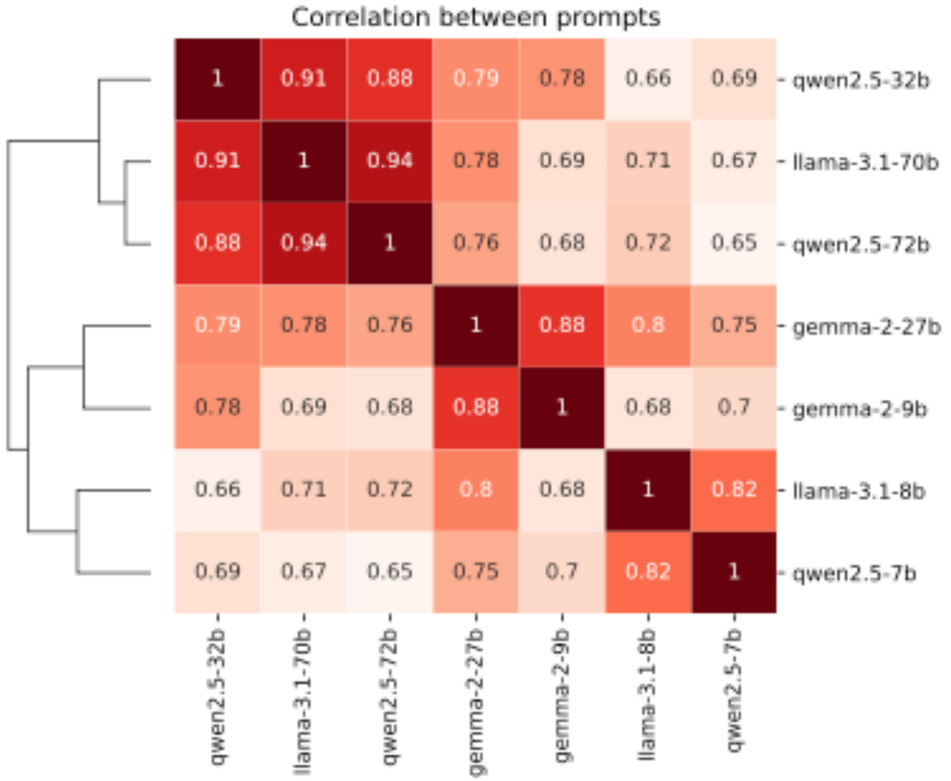


Correlation between rank of prompt configurations across models

High correlation => best prompt relatively stable across models

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

Prompt generalization



Correlation between rank of prompt configurations across models

High correlation => best prompt relatively stable across models

Correlation higher when model sizes and family are close

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

Conclusion



Conclusion



- LLM judges can be tuned at a reasonable cost

Conclusion



- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches

Conclusion



- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly

Conclusion



- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly
  - … and using only open-weight models

Conclusion



- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly
  - … and using only open-weight models
- Code to reproduce figures and all annotations available at

Conclusion



- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly
  - … and using only open-weight models
- Code to reproduce figures and all annotations available at
  - https://github.com/geoalgo/judgetuning

Multilingual LLM Judges

Multilingual LLM judges

How to evaluate on other languages?

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:
  - Translated to arena-hard-auto-v0.1 to 22 languages with Google API

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:
  - Translated to arena-hard-auto-v0.1 to 22 languages with Google API
  - Languages supported: Arabic Chinese Czech Dutch English French German Greek Hebrew Hindi Indonesian Italian Japanese Korean Persian Polish Portuguese Romanian Russian Spanish Turkish Ukrainian Vietnamese

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:
  - Translated to arena-hard-auto-v0.1 to 22 languages with Google API
  - Languages supported: Arabic Chinese Czech Dutch English French German Greek Hebrew Hindi Indonesian Italian Japanese Korean Persian Polish Portuguese Romanian Russian Spanish Turkish Ukrainian Vietnamese
- How to translate instructions like this?

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:
  - Translated to arena-hard-auto-v0.1 to 22 languages with Google API
  - Languages supported: Arabic Chinese Czech Dutch English French German Greek Hebrew Hindi Indonesian Italian Japanese Korean Persian Polish Portuguese Romanian Russian Spanish Turkish Ukrainian Vietnamese
- How to translate instructions like this?
  - “Translate this text in English: Le meilleur petit-déjeuner contient au moins un croissant.”

Multilingual LLM judges

How to evaluate on other languages?

- Alpaca-Eval and Arena-Hard are mostly English…
- Arena-Hard is also very biased toward technical questions
- How to evaluate to other languages?
- m-Arena-Hard:
  - Translated to arena-hard-auto-v0.1 to 22 languages with Google API
  - Languages supported: Arabic Chinese Czech Dutch English French German Greek Hebrew Hindi Indonesian Italian Japanese Korean Persian Polish Portuguese Romanian Russian Spanish Turkish Ukrainian Vietnamese
- How to translate instructions like this?
  - “Translate this text in English: Le meilleur petit-déjeuner contient au moins un croissant.”
  - “Ecris un code en python. ```def calcule_fibonacci(i: int) -> int```

Multilingual LLM judges

Research questions

Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Diacritics and apostrophes messed up…

Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Diacritics and apostrophes messed up…

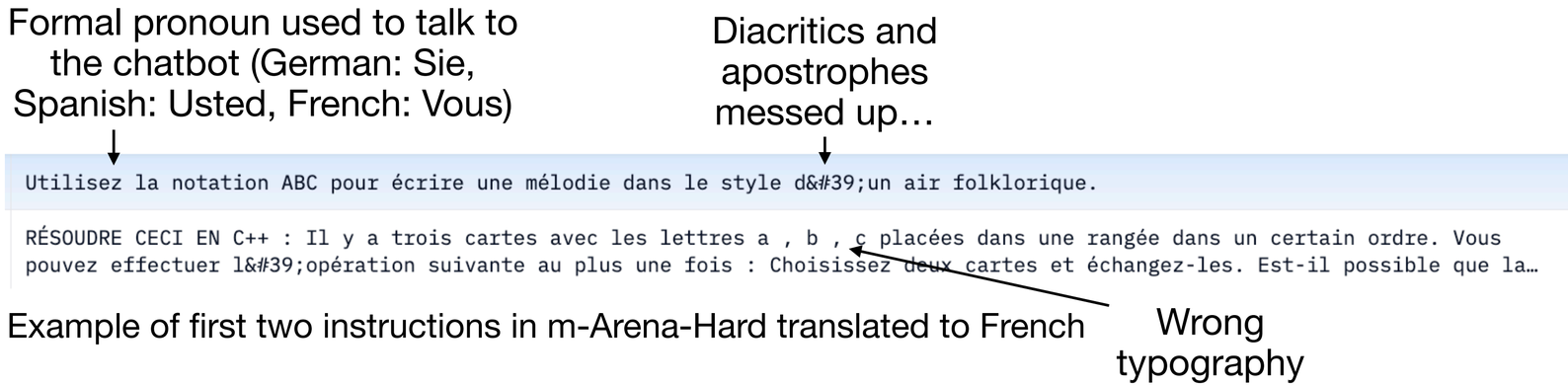
Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Wrong typography

Multilingual LLM judges

Research questions

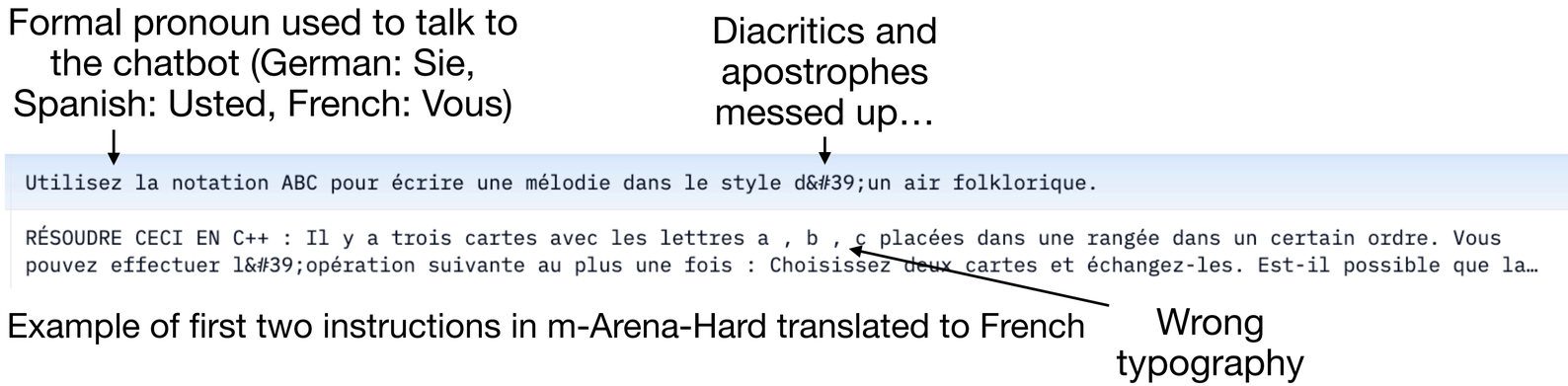


Example of first two instructions in m-Arena-Hard translated to French

- Can we have avoid the low translation quality artifacts?

Multilingual LLM judges

Research questions

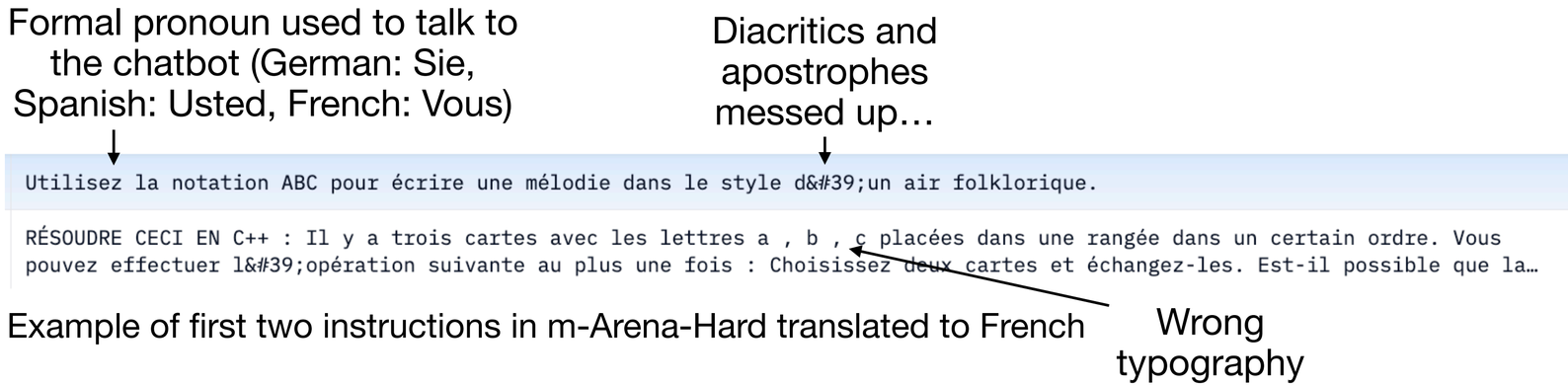


Example of first two instructions in m-Arena-Hard translated to French

- Can we have avoid the low translation quality artifacts?
- Can we evaluate the ability of an LLM judge to perform evaluation at all in low-resource languages?

Multilingual LLM judges

Research questions



- Can we have avoid the low translation quality artifacts?
- Can we evaluate the ability of an LLM judge to perform evaluation at all in low-resource languages?
- Can we use multilingual LLM judges to analyse the strength and weaknesses of a given model?

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Diacritics and apostrophes messed up…

Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Wrong typography

- Can we have avoid the low translation quality artifacts?
- Can we evaluate the ability of an LLM judge to perform evaluation at all in low-resource languages?
- Can we use multilingual LLM judges to analyse the strength and weaknesses of a given model?

How would you do that? 🤔

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Diacritics and apostrophes messed up…

Utilisez la notation ABC pour écrire une mélodie dans le style d'un air folklorique.
RÉSOUDRE CECI EN C++ : Il y a trois cartes avec les lettres a , b , c placées dans une rangée dans un certain ordre. Vous pouvez effectuer l'opération suivante au plus une fois : Choisissez deux cartes et échangez-les. Est-il possible que la…

Example of first two instructions in m-Arena-Hard translated to French

Wrong typography

- Can we have avoid the low translation quality artifacts?
- Can we evaluate the ability of an LLM judge to perform evaluation at all in low-resource languages?
- Can we use multilingual LLM judges to analyse the strength and weaknesses of a given model?

How would you do that? 🤔

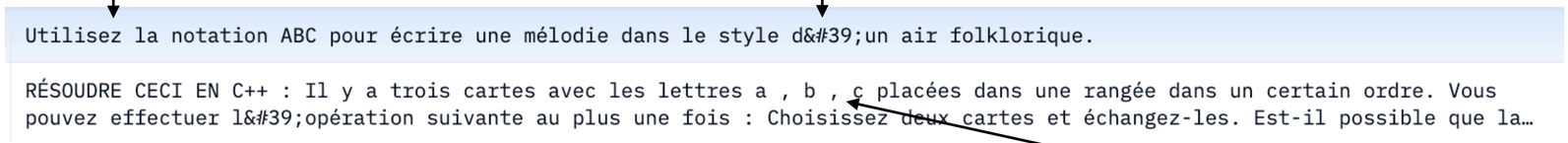
How would you do that? 🤔

Multilingual LLM judges

Research questions

Formal pronoun used to talk to the chatbot (German: Sie, Spanish: Usted, French: Vous)

Diacritics and apostrophes messed up…



Example of first two instructions in m-Arena-Hard translated to French

Wrong typography

- Can we have avoid the low translation quality artifacts?
- Can we evaluate the ability of an LLM judge to perform evaluation at all in low-resource languages?
- Can we use multilingual LLM judges to analyse the strength and weaknesses of a given model?

How would you do that? 🤔

How would you do that? 🤔

Note: ongoing work ⚠️

Multilingual LLM judges

Sourcing native instructions

Multilingual LLM judges

Sourcing native instructions

- Consider instructions from LMSys-55K and ComparIA

Multilingual LLM judges

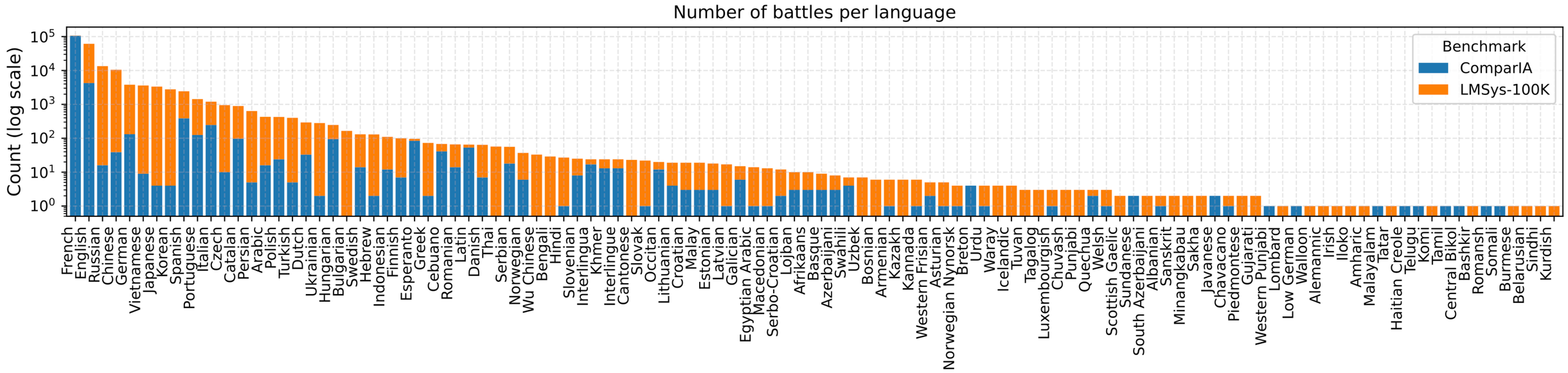
Sourcing native instructions

- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions

Multilingual LLM judges

Sourcing native instructions

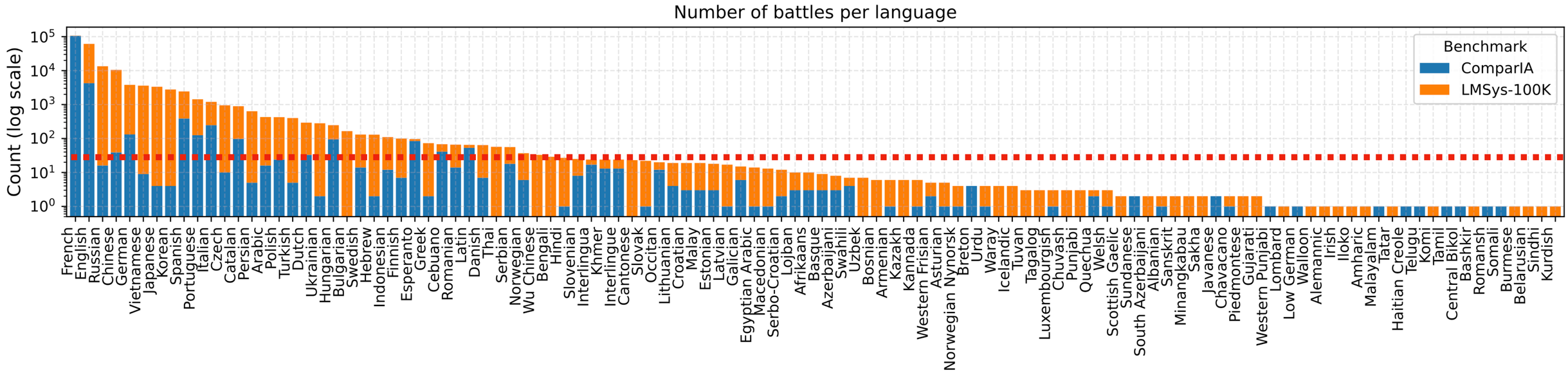
- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions



Multilingual LLM judges

Sourcing native instructions

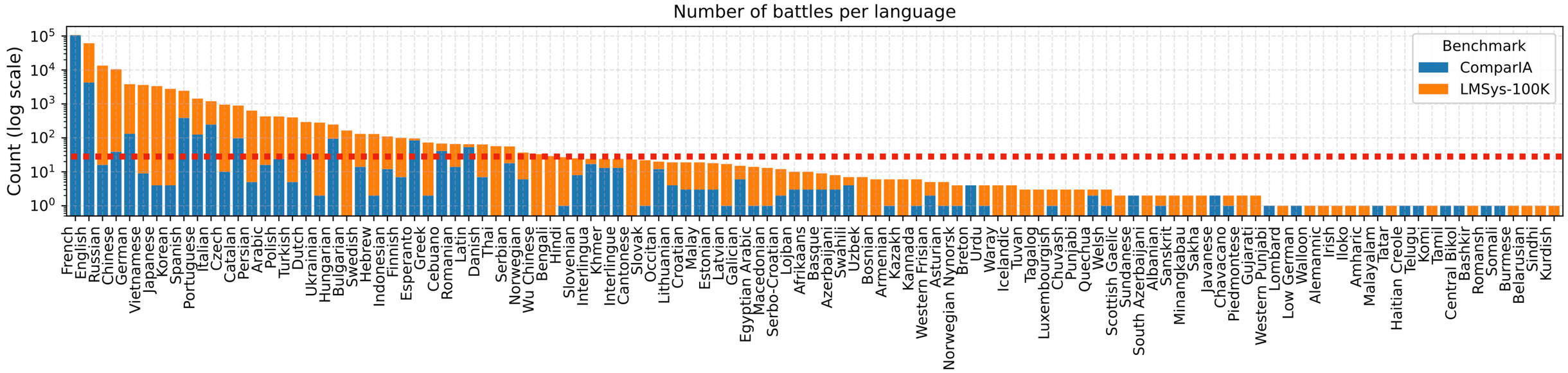
- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions



Multilingual LLM judges

Sourcing native instructions

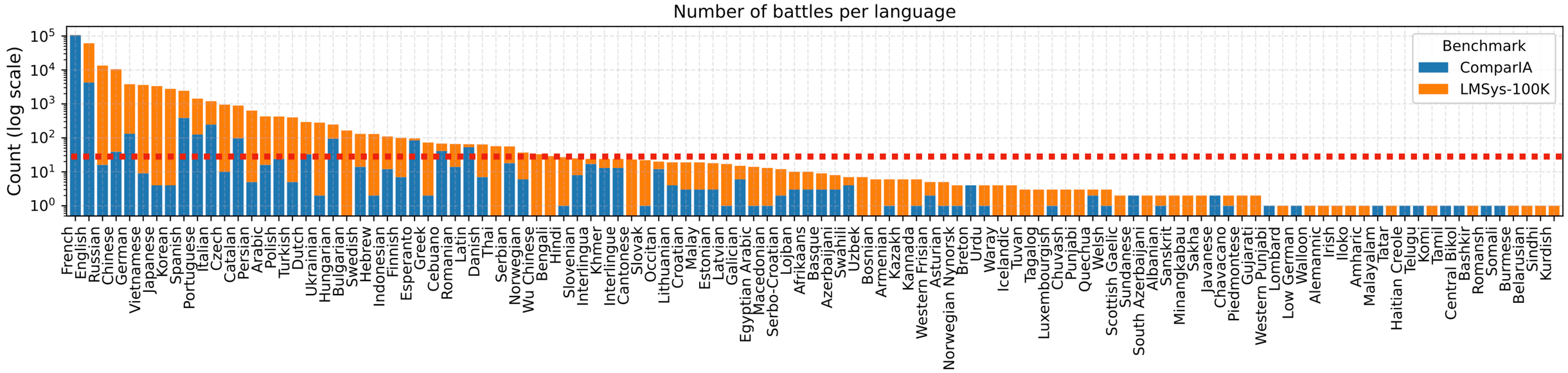
- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions
- European: English, French, German, Spanish, Portuguese, Italian, Czech, Polish, Swedish, Finnish, Catalan, Turkish, Ukrainian, Dutch, Hungarian, Bulgarian, Swedish, Danish, Romanian, Greek, Norwegian, Serbian



Multilingual LLM judges

Sourcing native instructions

- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions
- European: English, French, German, Spanish, Portuguese, Italian, Czech, Polish, Swedish, Finnish, Catalan, Turkish, Ukrainian, Dutch, Hungarian, Bulgarian, Swedish, Danish, Romanian, Greek, Norwegian, Serbian
- Non European: Russian, Chinese, Vietnamese, Japanese, Korean, Persian, Arabic, Indonesian, Hebrew, Esperanto

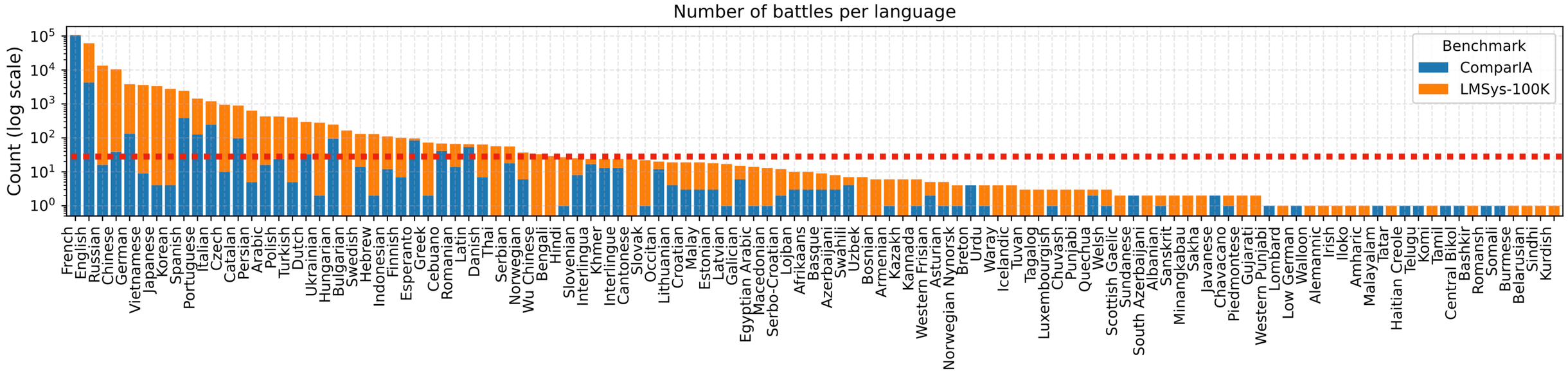


Multilingual LLM judges

Sou

Ok but can we say whether the LLM judge works for those languages?

- Consider instructions from LMSys-55K and ComparIA
- Combination allows to consider 25 languages with 100 native instructions and 7 with 50-100 instructions
- European: English, French, German, Spanish, Portuguese, Italian, Czech, Polish, Swedish, Finnish, Catalan, Turkish, Ukrainian, Dutch, Hungarian, Bulgarian, Swedish, Danish, Romanian, Greek, Norwegian, Serbian
- Non European: Russian, Chinese, Vietnamese, Japanese, Korean, Persian, Arabic, Indonesian, Hebrew, Esperanto



Multilingual LLM judges

Evaluating LLM judge capability in other languages

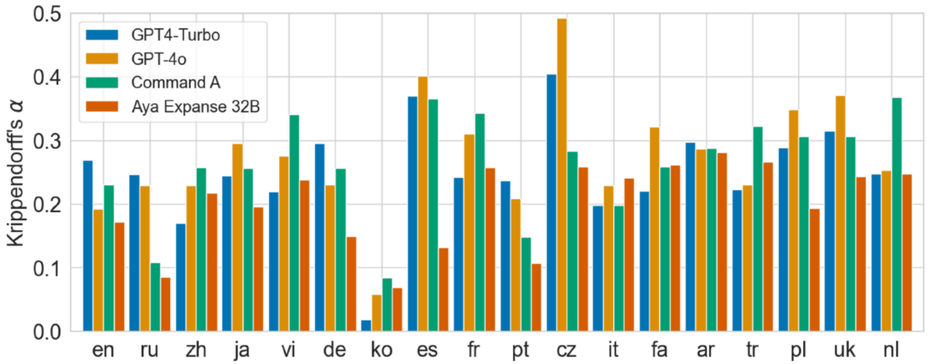


Figure 10: LLM-as-a-judge agreement with humans on arena.

	α
GPT-4-Turbo	0.25
GPT-4o	**0.28**
Command A	0.26
Aya Expanse 32B	0.20
	# Best
GPT-4-Turbo	5
GPT-4o	**6**
Command A	**6**
Aya Expanse 32B	1

Table 4: Summary across languages.

Déjà Vu: Multilingual LLM Evaluation through the Lens of Machine Translation Evaluation

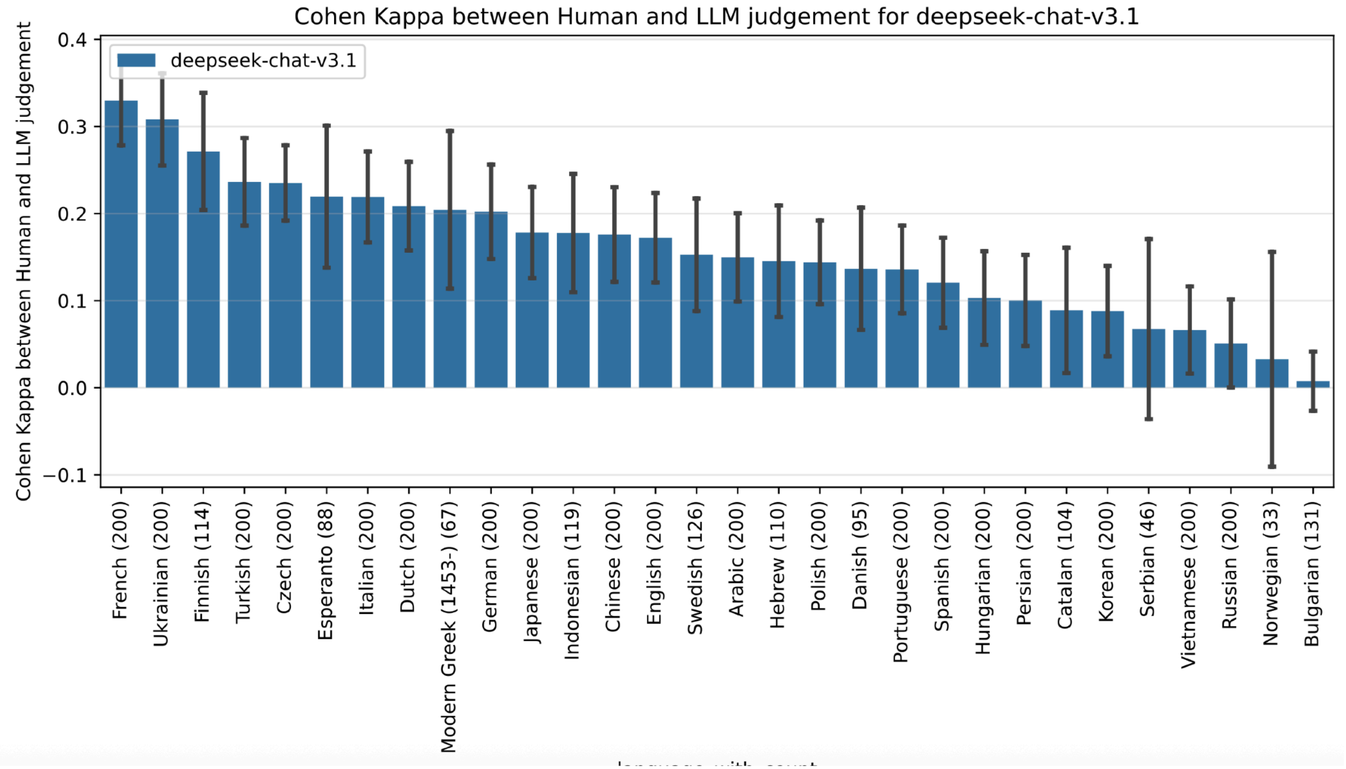
Julia Kreutzer[1], Eleftheria Briakou[2], Sweta Agrawal[2], Marzieh Fadaee[1], and Kocmi Tom[3]

[1]Cohere Labs, [2]Google, [3]Cohere

- Score a subset of LMSys 100K public battles for 18 language, 200 generation pairs each
- Use two open and two proprietary LLMs as judges
- Report Krippendorff's: i.e. agreement rate between humans and LLMs
- Higher is better, zero indicates randomness
- Confidence intervals? Other languages?

Multilingual LLM judges

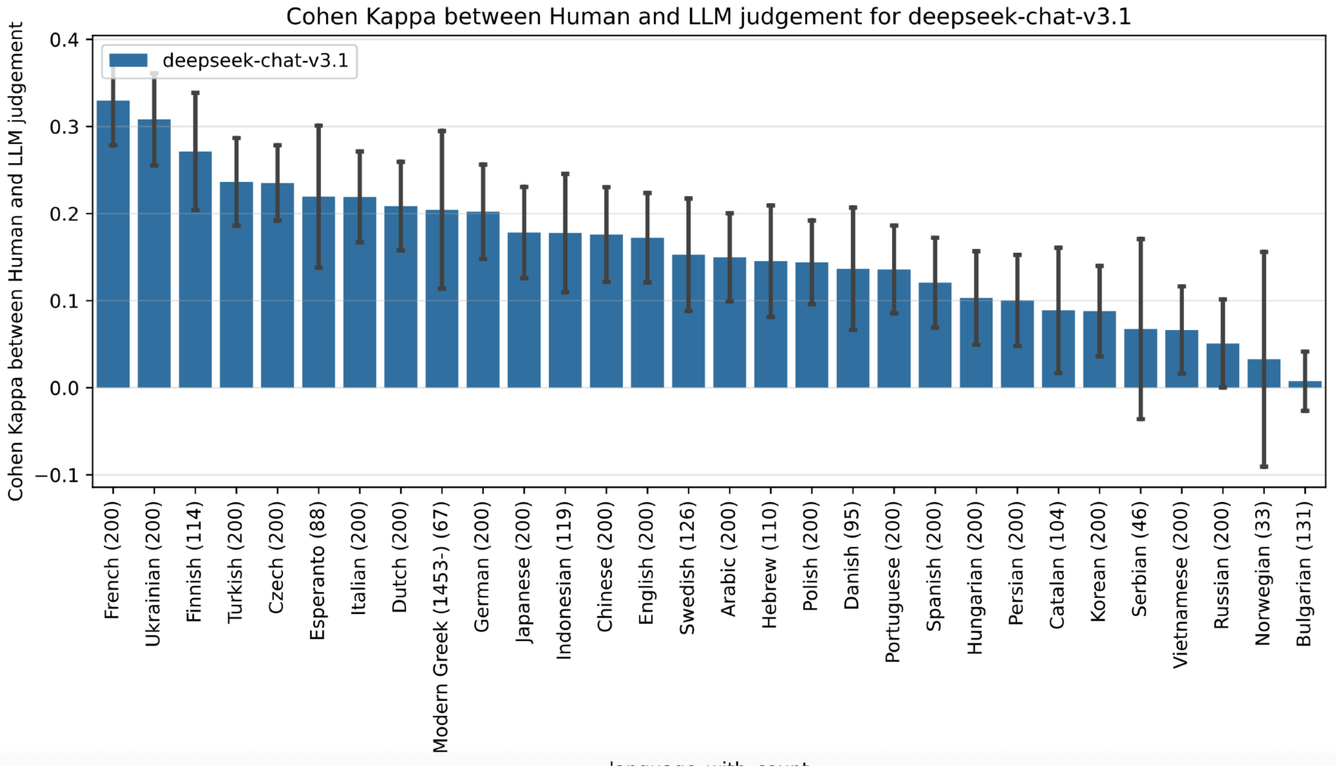
Evaluating LLM judge capability in other languages



Multilingual LLM judges

Evaluating LLM judge capability in other languages

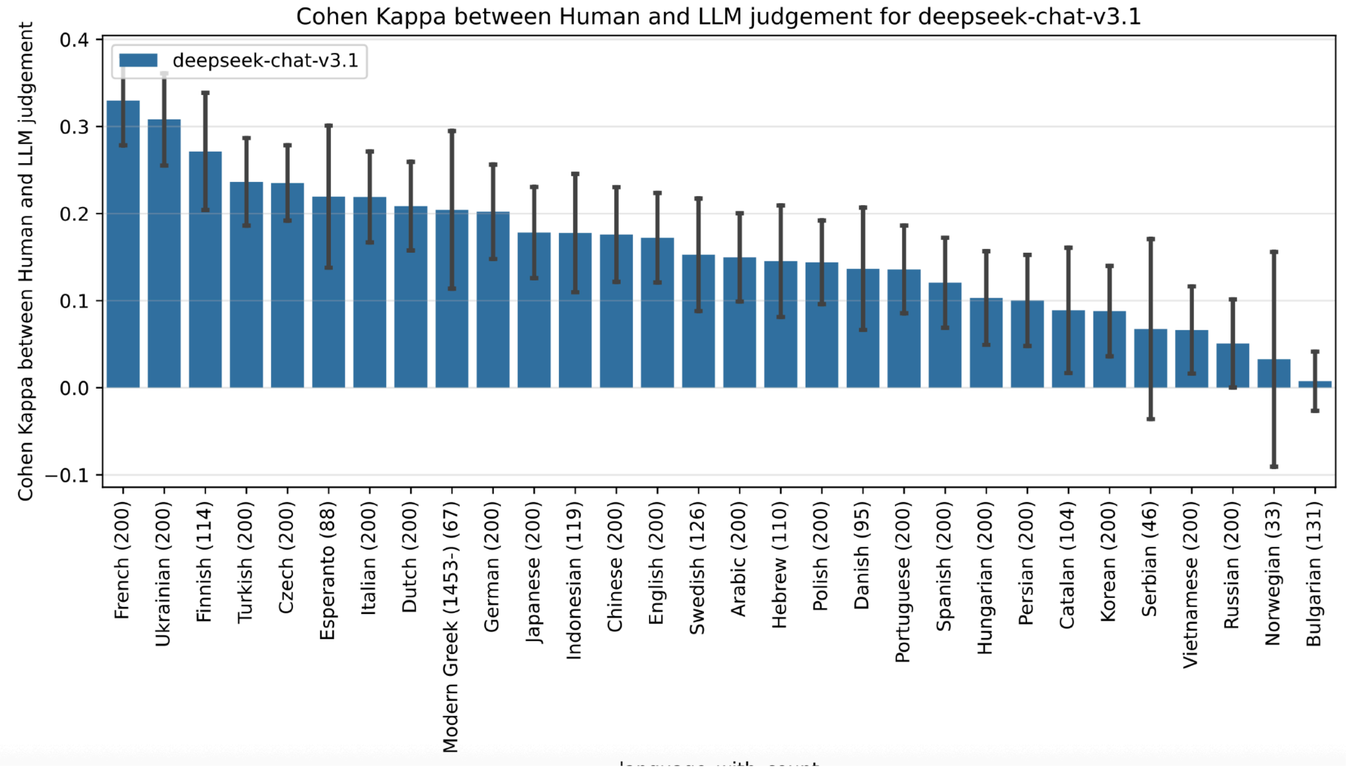
- Highlight languages where LLM judges…



Multilingual LLM judges

Evaluating LLM judge capability in other languages

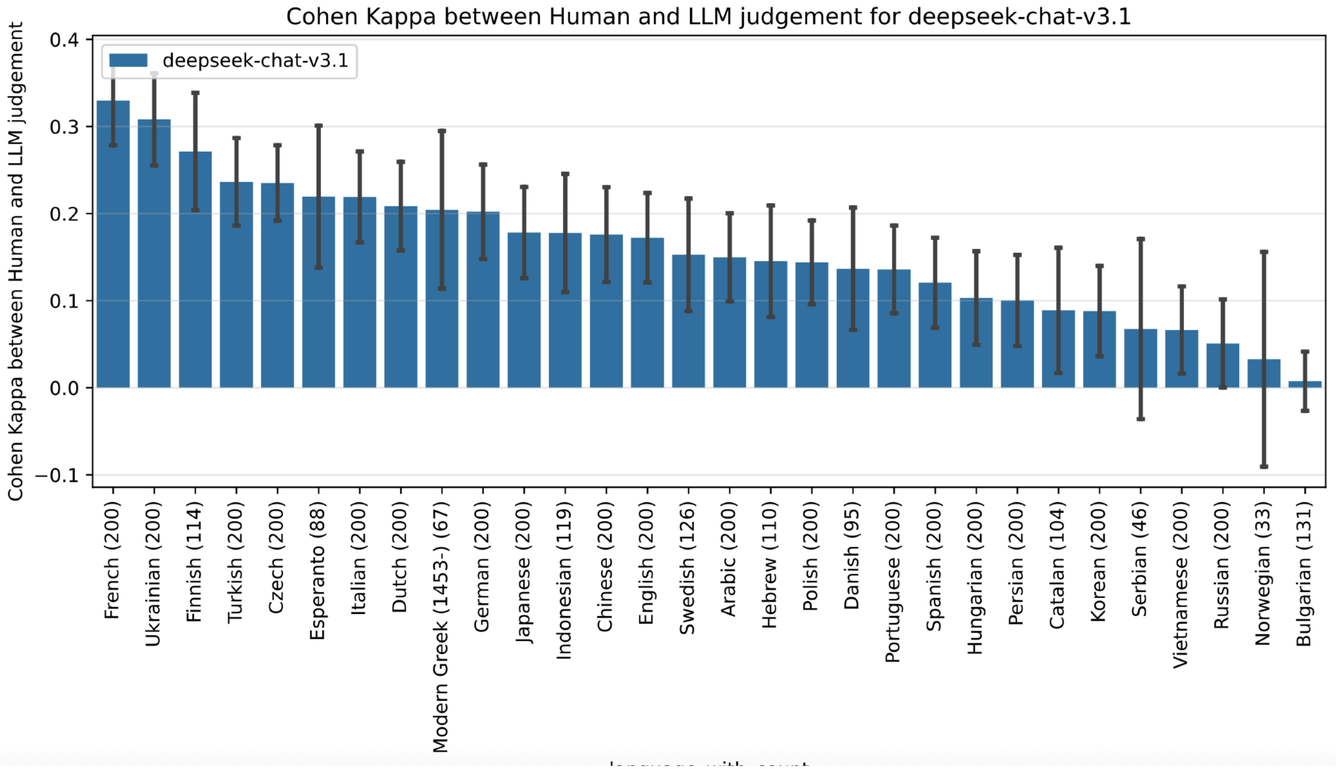
- Highlight languages where LLM judges…
  - **can be used:** Finnish, Ukrainian, Catalan, Polish, Hungarian, …



Multilingual LLM judges

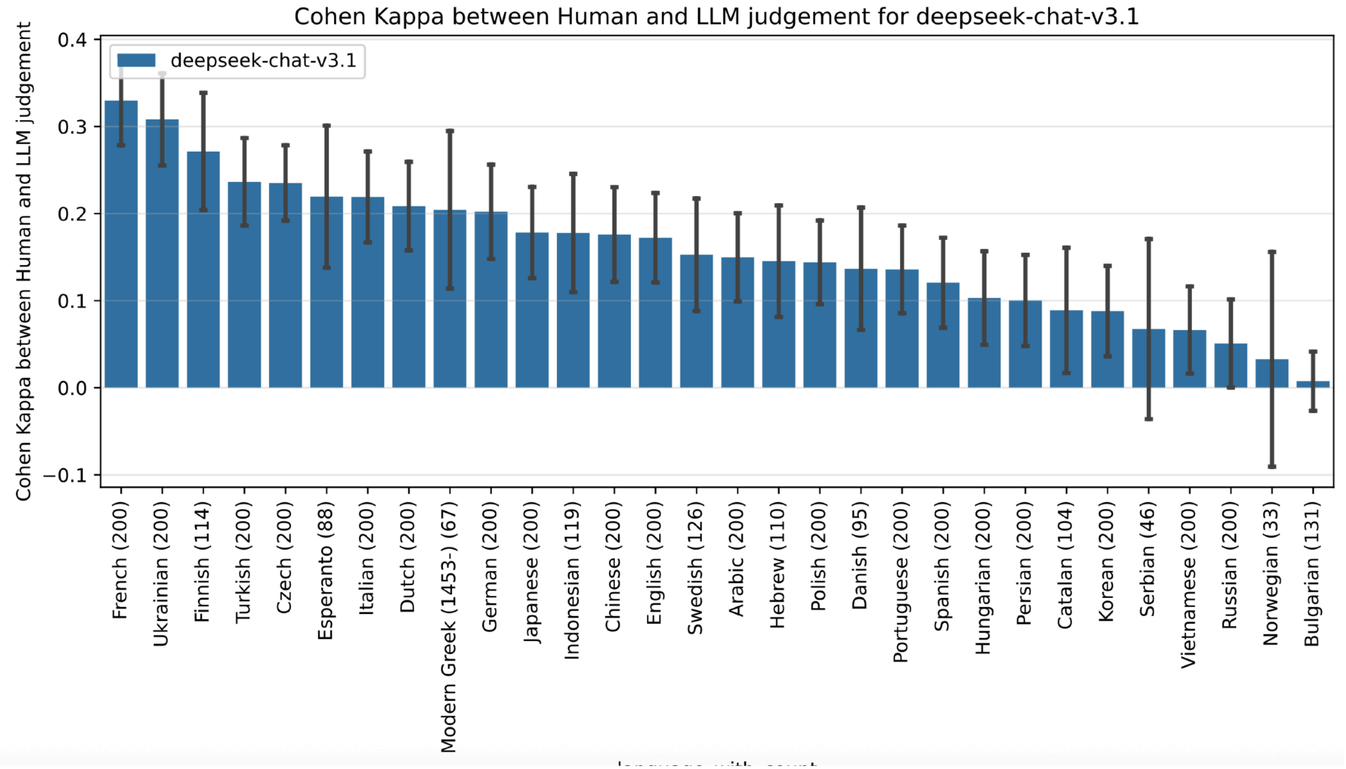
Evaluating LLM judge capability in other languages

- Highlight languages where LLM judges…
  - **can be used:** Finnish, Ukrainian, Catalan, Polish, Hungarian, …
  - **cannot be used** (at the moment)**:** Bulgarian, Norwegian, Russian, Vietnamese, Serbian, Catalan



Multilingual LLM judges

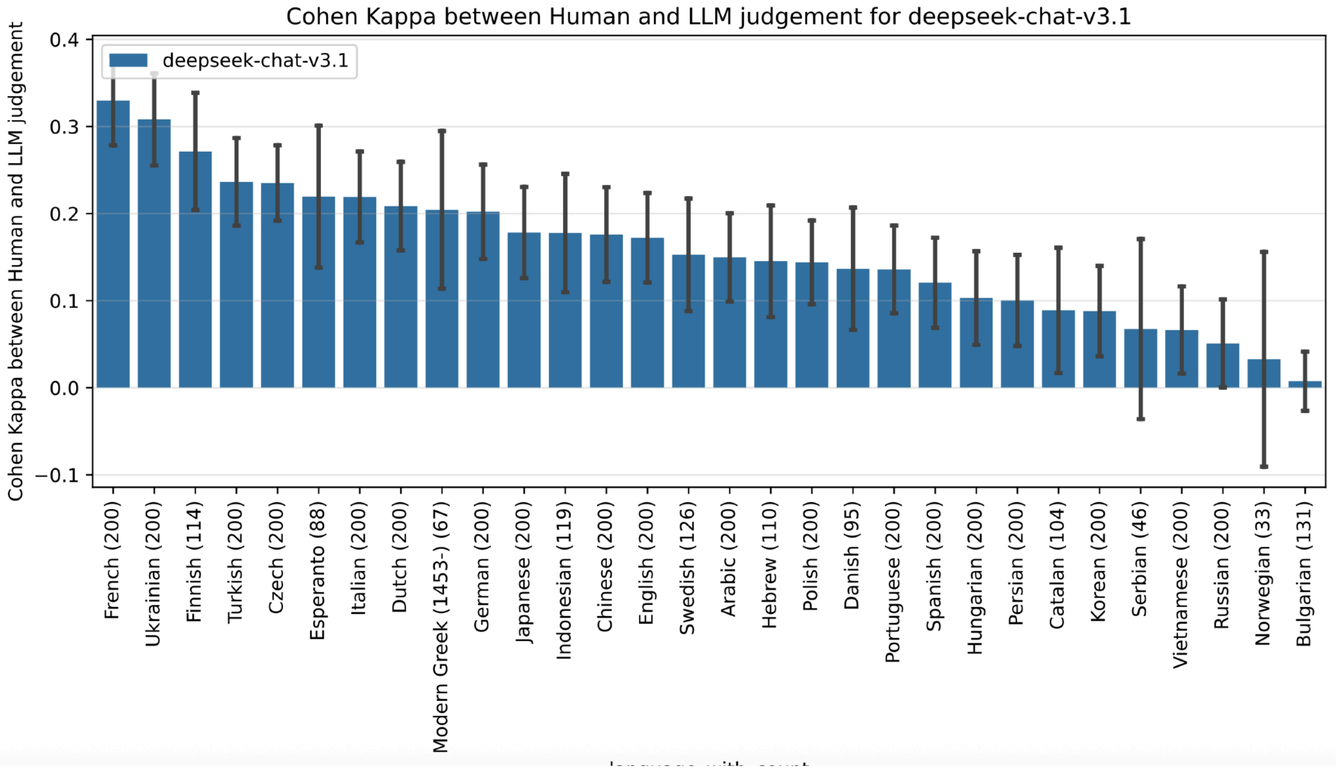
Evaluating LLM judge capability in other languages



Multilingual LLM judges

Evaluating LLM judge capability in other languages

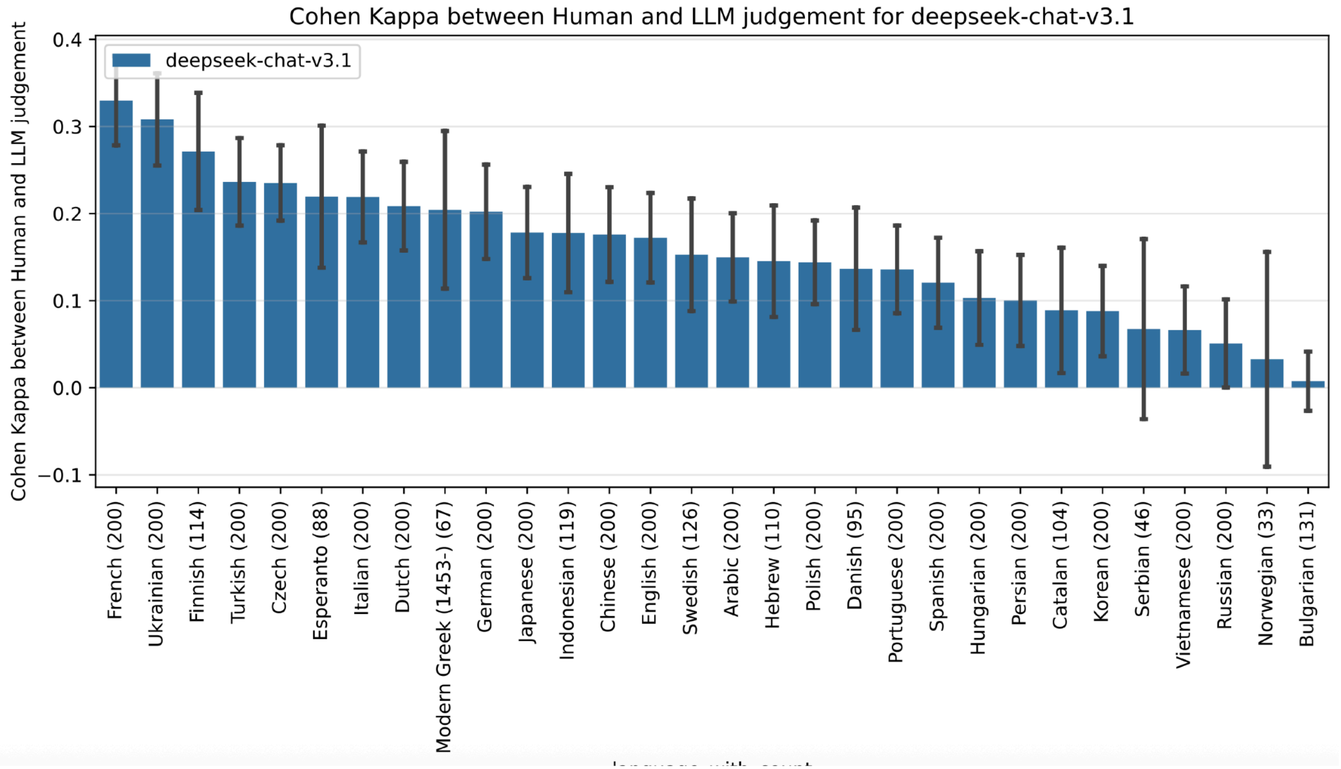
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:



Multilingual LLM judges

Evaluating LLM judge capability in other languages

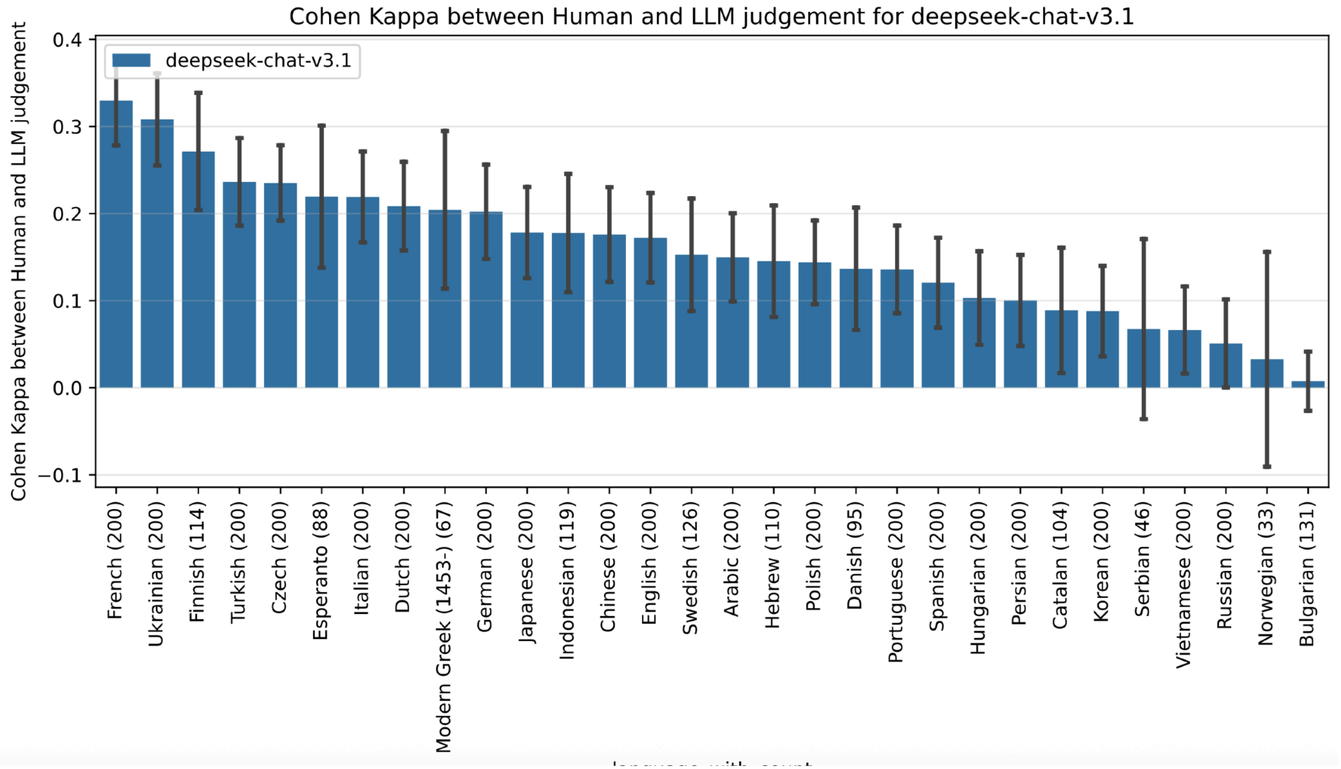
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12



Multilingual LLM judges

Evaluating LLM judge capability in other languages

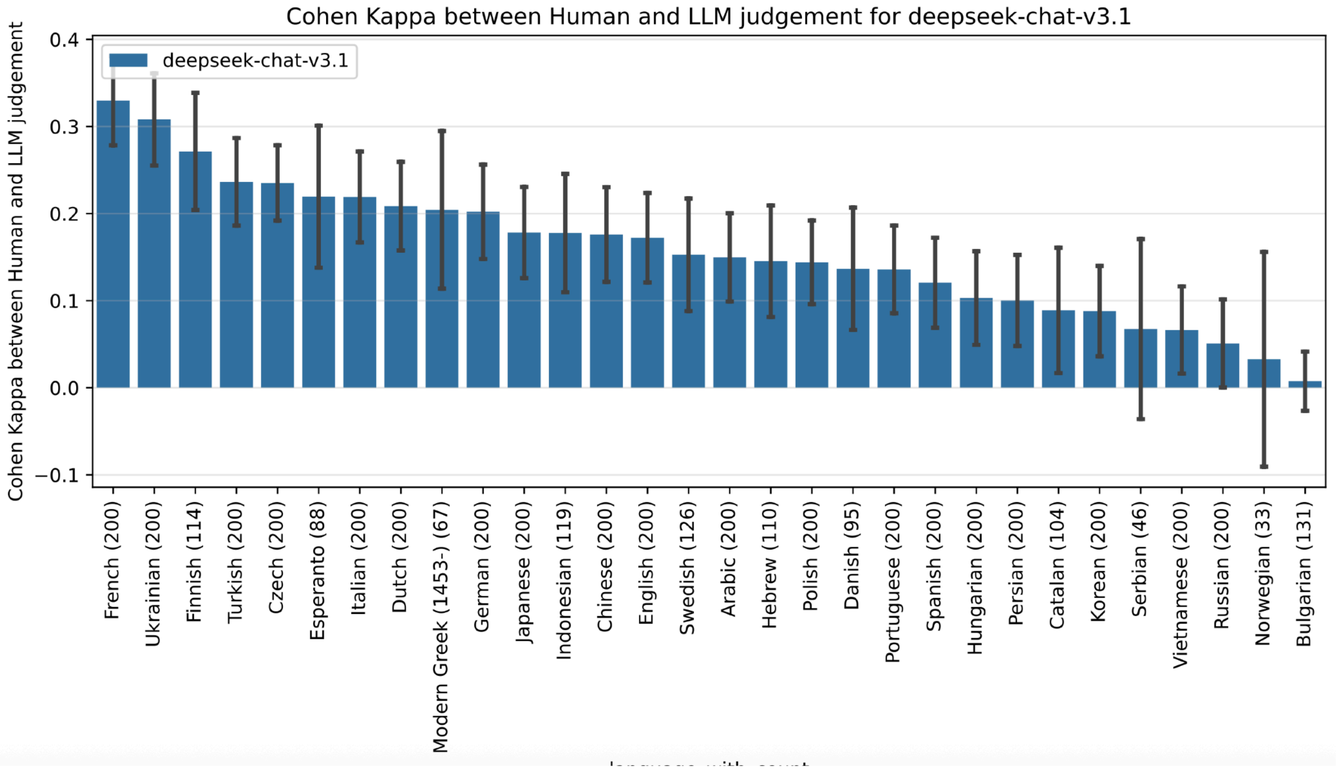
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12
  - French: 0.16 +/- 0.15



Multilingual LLM judges

Evaluating LLM judge capability in other languages

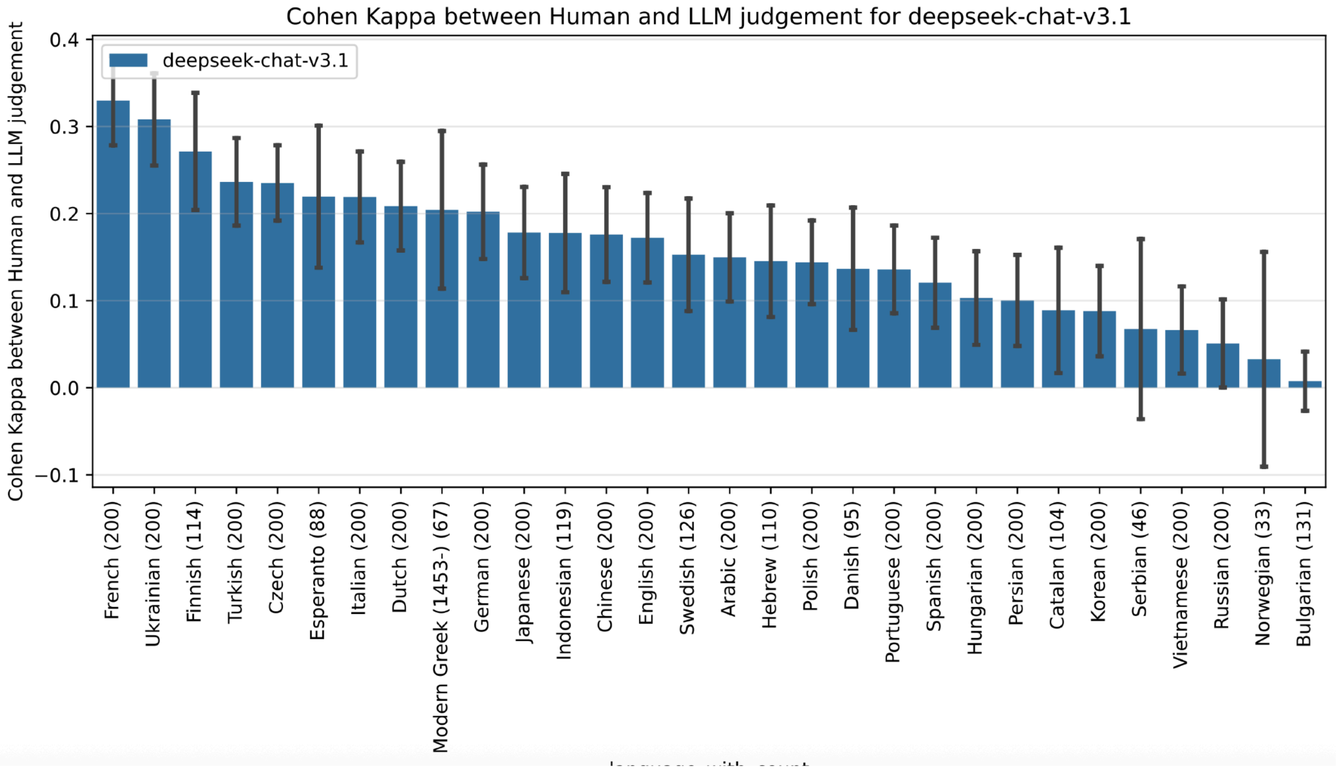
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12
  - French: 0.16 +/- 0.15
  - LLM judges in the same ballpark (perhaps better?)



Multilingual LLM judges

Evaluating LLM judge capability in other languages

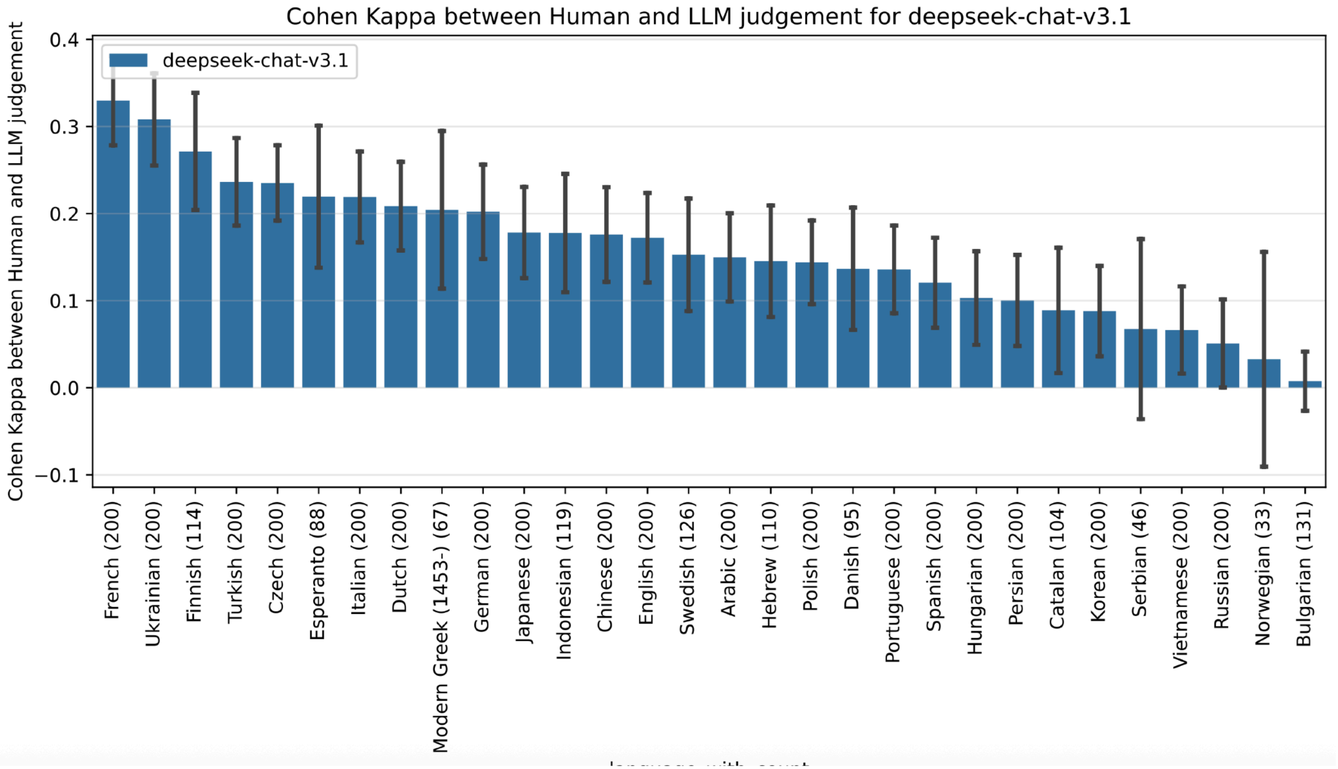
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12
  - French: 0.16 +/- 0.15
  - LLM judges in the same ballpark (perhaps better?)
- Code for annotations: https://huggingface.co/datasets/openeurollm/battle-annotations/tree/main



Multilingual LLM judges

Evaluating LLM judge capability in other languages

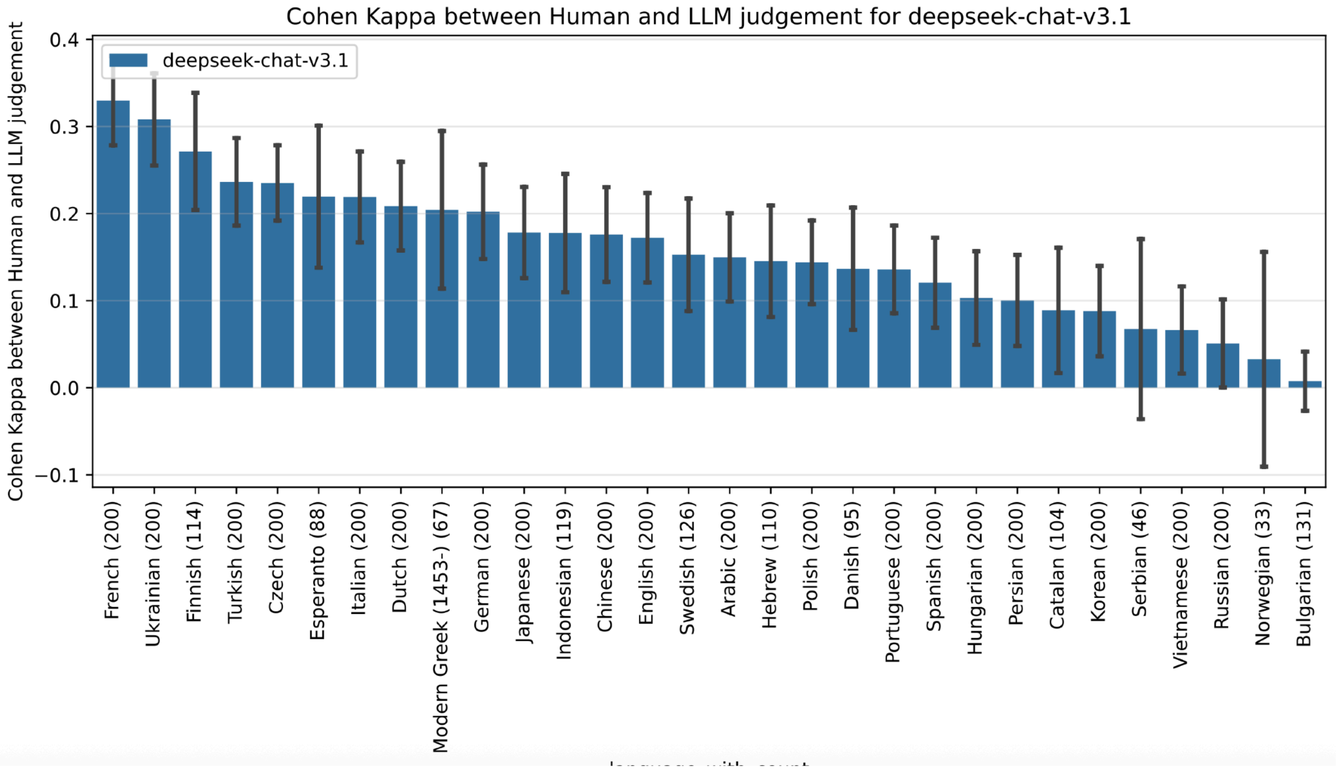
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12
  - French: 0.16 +/- 0.15
  - LLM judges in the same ballpark (perhaps better?)
- Code for annotations: https://huggingface.co/datasets/openeurollm/battle-annotations/tree/main
- Dataset: https://huggingface.co/datasets/openeurollm/battle-annotations/tree/main



Multilingual LLM judges

Evaluating LLM judge capability in other languages

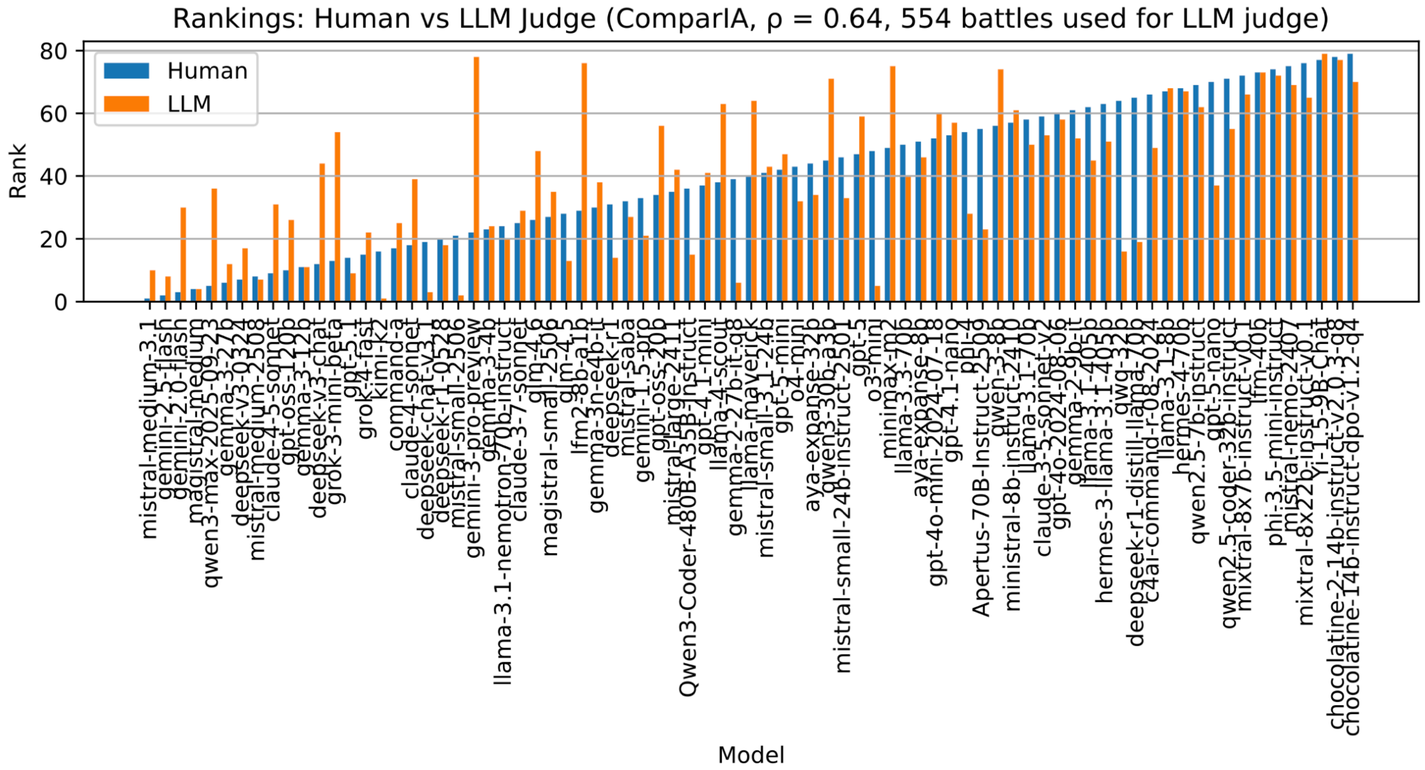
- I have annotated 100 instructions in English and French, my Cohen-Kappa was:
  - English: 0.13 +/- 0.12
  - French: 0.16 +/- 0.15
  - LLM judges in the same ballpark (perhaps better?)
- Code for annotations: https://huggingface.co/datasets/openeurollm/battle-annotations/tree/main
- Dataset: https://huggingface.co/datasets/openeurollm/battle-annotations/tree/main
- Your annotations are welcome 🤗, I highly recommend to take a look at random instructions to see what the data looks like

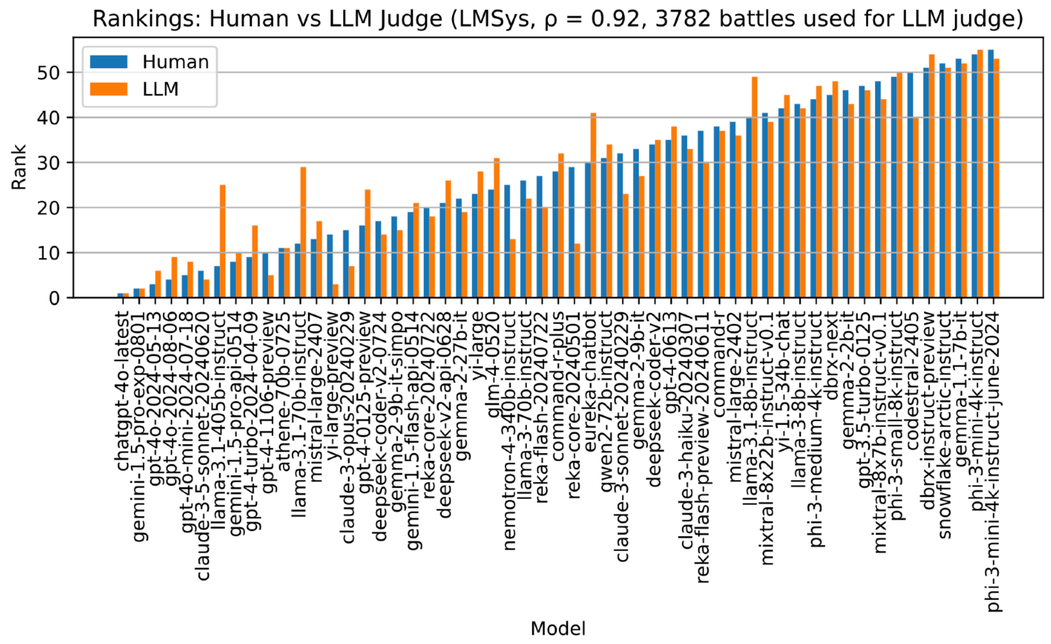


Multilingual LLM judges

Comparing LLM and Human ranking on LMSys on multilingual

- How this local agreement translates to agreement for “global” rankings for multilingual?
- Compute rankings with 31 languages with at most 200 instructions
- Only include languages where LLM agreement rate clearly exceeds random chance





Multilingual LLM judges

Multilingual LLM judges

- Ok so:

Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level

Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)

Multilingual LLM judges

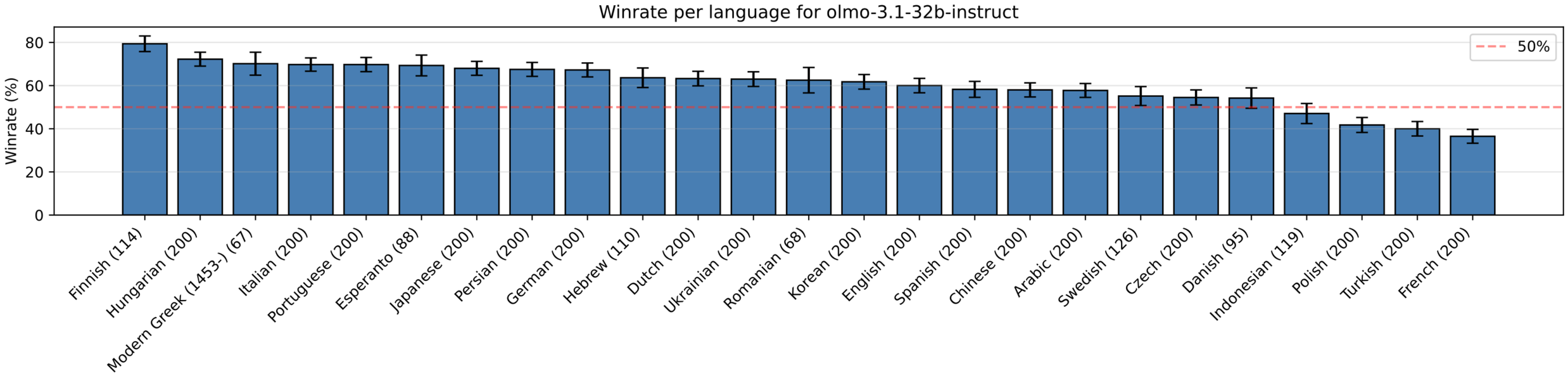
- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model

Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model
- What results for Olmo3-32B?

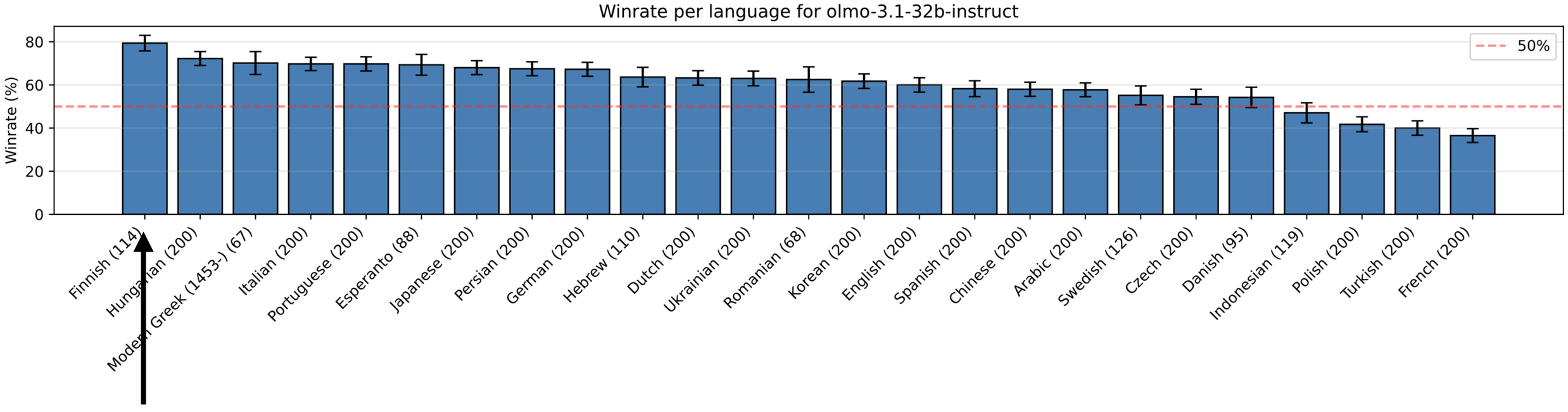
Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model
- What results for Olmo3-32B?



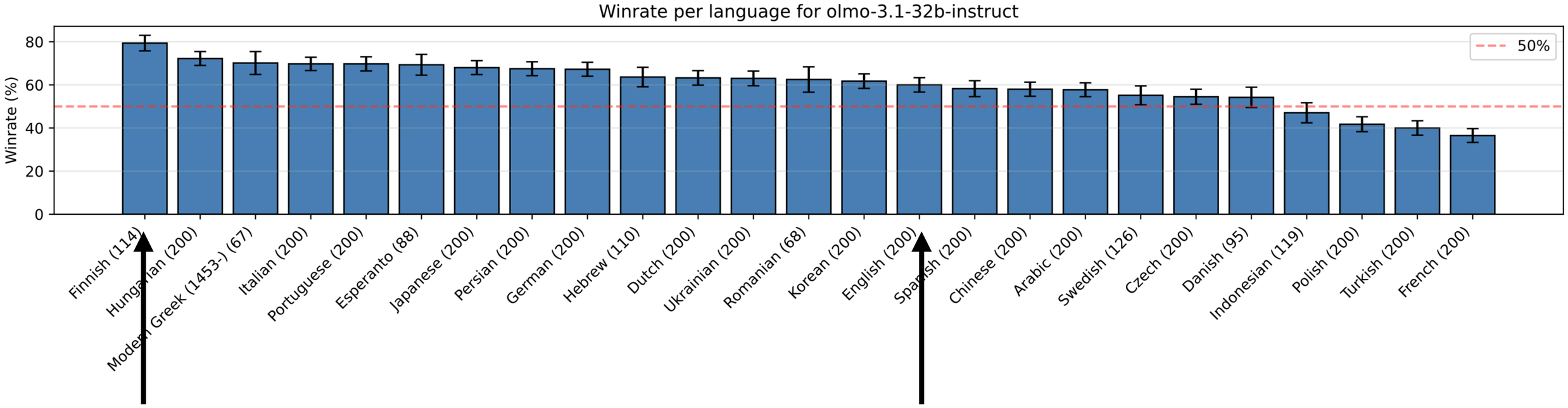
Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model
- What results for Olmo3-32B?



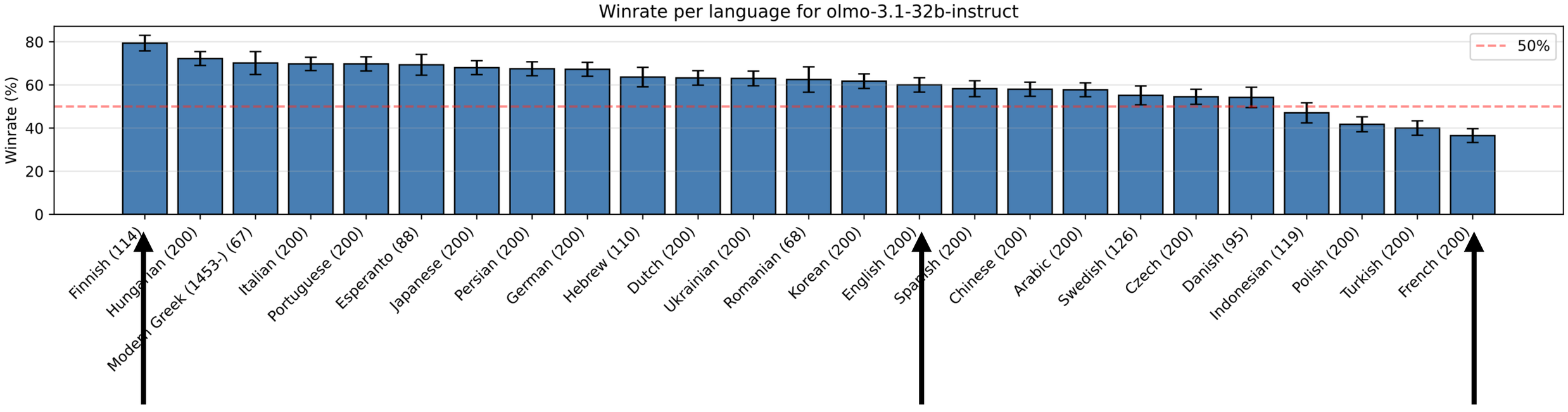
Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model
- What results for Olmo3-32B?



Multilingual LLM judges

- Ok so:
  - we can trust annotations to match human annotations above random level
  - we have native instructions instead of translated ones (could still complement with translated instructions)
- From a model development perspective, we want to understand what are strengths and weaknesses of our model
- What results for Olmo3-32B?



Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

135	122 ◂–▸ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ◂–▸ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ◂–▸ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ◂–▸ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ◂–▸ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ◂–▸ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ◂–▸ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ◂–▸ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ◂–▸ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ◂–▸ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ◂–▸ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ◂–▸ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

- Since we use random baselines, we can compute ELO ratings and compare scores with LMArena or ComparIA

135	122 ↔ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ↔ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ↔ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ↔ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ↔ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ↔ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ↔ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ↔ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ↔ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ↔ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ↔ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ↔ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

- Since we use random baselines, we can compute ELO ratings and compare scores with LMArena or ComparIA
- ELO rating computed on LLM judge annotations: **1306 ± 23**

135	122 ↔ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ↔ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ↔ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ↔ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ↔ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ↔ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ↔ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ↔ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ↔ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ↔ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ↔ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ↔ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

- Since we use random baselines, we can compute ELO ratings and compare scores with LMArena or ComparIA
- ELO rating computed on LLM judge annotations: **1306 ± 23**
- vs **1324** on LMArena

135	122 ↔ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ↔ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ↔ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ↔ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ↔ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ↔ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ↔ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ↔ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ↔ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ↔ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ↔ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ↔ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

- Since we use random baselines, we can compute ELO ratings and compare scores with LMArena or ComparIA
- ELO rating computed on LLM judge annotations: **1306 ± 23**
- vs **1324** on LMArena
- Useful because it allows to quantify winning probabilities against baselines

135	122 ◂▸ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ◂▸ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ◂▸ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ◂▸ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ◂▸ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ◂▸ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ◂▸ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ◂▸ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ◂▸ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ◂▸ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ◂▸ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ◂▸ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Multilingual LLM judges

Estimating ELO ratings directly (instead of winrates)

- Since we use random baselines, we can compute ELO ratings and compare scores with LMArena or ComparIA
- ELO rating computed on LLM judge annotations: **1306 ± 23**
- vs **1324** on LMArena
- Useful because it allows to quantify winning probabilities against baselines
- Careful analysis required, it is unlikely that we will be able to estimate ELO ratings this well for all models

135	122 ↔ 154	qwen3-30b-a3b	1328	±5	27428	Alibaba	Apache 2.0
136	123 ↔ 153	llama-4-maverick-17b-128e-instruct	1328	±4	41136	Meta	Llama 4
137	113 ↔ 162	llama-3.3-nemotron-49b-super-v1	1327	±12	2230	Nvidia	Nvidia
138	118 ↔ 162	hunyuan-large-2025-02-10	1327	±10	3738	Tencent	Proprietary
139	133 ↔ 158	gpt-4-turbo-2024-04-09	1325	±4	98130	OpenAI	Proprietary
140	133 ↔ 158	claude-3-5-haiku-20241022	1324	±3	71179	Anthropic	Proprietary
141	128 ↔ 162	olmo-3.1-32b-instruct	1324	±7	8310	Allen AI	Apache 2.0
142	124 ↔ 162	deepseek-v2.5-1210	1323	±8	6793	DeepSeek	DeepSeek
143	133 ↔ 158	gemini-1.5-pro-001	1323	±4	79132	Google	Proprietary
144	133 ↔ 160	llama-4-scout-17b-16e-instruct	1323	±5	31231	Meta	Llama
145	133 ↔ 158	claude-3-opus-20240229	1323	±3	194904	Anthropic	Proprietary
146	132 ↔ 163	gpt-4.1-nano-2025-04-14	1322	±8	6107	OpenAI	Proprietary

Comparing with previous LLM judges setup

Feature	m-ArenaHard	Ours
Baseline	Fixed	Random
Instructions	Selected	Random
# Languages	22	31
Source	Translated 🤖	Sourced 🙋
LLM judge check	❌	✅
Can approximate ELO scores	❌	✅

- Future work:
  - Filter by instruction quality
  - Report performance per topic
  - Check ELO ratings gaps for other models

OpenJury

An open library to run LLM judges benchmarks

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …

Framework	MT-Bench	AlpacaEval	Arena-Hard	M-Arena-Hard	Tuned judge configuration	Support vLLM Judges
FastChat	✅	❌	❌	❌	❌	❌
AlpacaEval	❌	✅	❌	❌	❌	❌
Arena-Hard-Auto	❌	❌	✅	❌	❌	❌
Lighteval	✅	❌	❌	❌	❌	❌
Evalchemy	✅	✅	❌	❌	❌	❌

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

Framework	MT-Bench	AlpacaEval	Arena-Hard	M-Arena-Hard	Tuned judge configuration	Support vLLM Judges
FastChat	✅	❌	❌	❌	❌	❌
AlpacaEval	❌	✅	❌	❌	❌	❌
Arena-Hard-Auto	❌	❌	✅	❌	❌	❌
Lighteval	✅	❌	❌	❌	❌	❌
Evalchemy	✅	✅	❌	❌	❌	❌

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

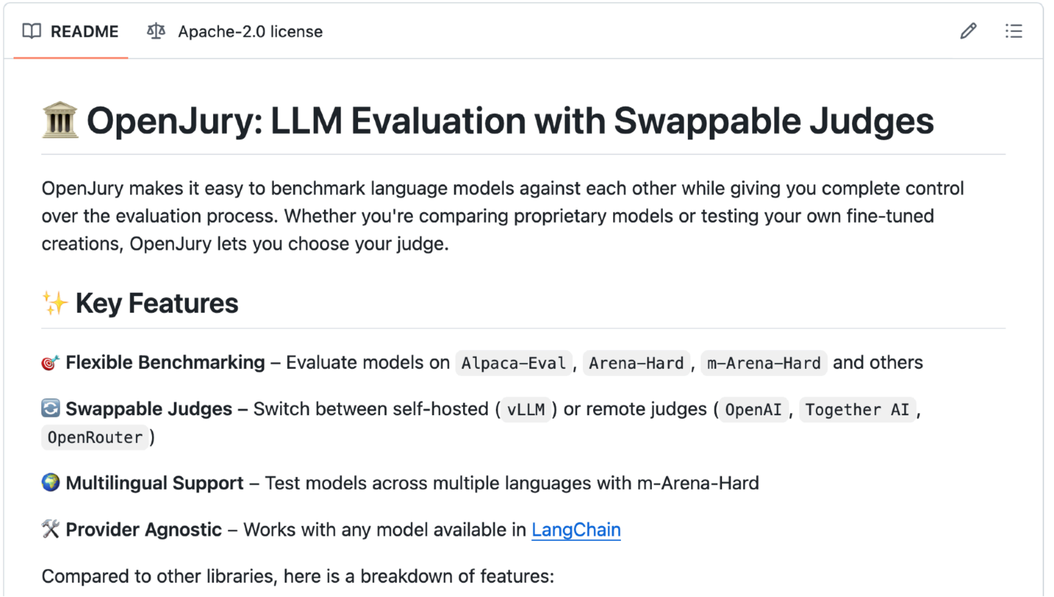
https://github.com/OpenEuroLLM/OpenJury

Framework	MT-Bench	AlpacaEval	Arena-Hard	M-Arena-Hard	Tuned judge configuration	Support vLLM Judges
FastChat	✅	❌	❌	❌	❌	❌
AlpacaEval	❌	✅	❌	❌	❌	❌
Arena-Hard-Auto	❌	❌	✅	❌	❌	❌
Lighteval	✅	❌	❌	❌	❌	❌
Evalchemy	✅	✅	❌	❌	❌	❌
OpenJury	➡️ SOON	✅	✅	✅	✅	✅

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …
- With local judges, OpenRouter, vLLM …



README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

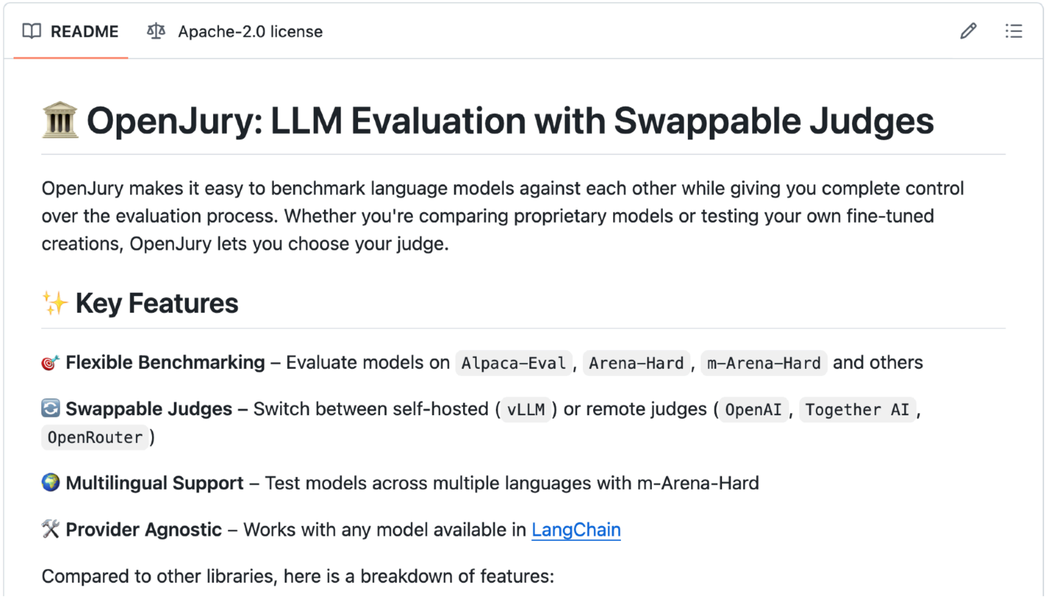
https://github.com/OpenEuroLLM/OpenJury

Framework	MT-Bench	AlpacaEval	Arena-Hard	M-Arena-Hard	Tuned judge configuration	Support vLLM Judges
FastChat	✅	❌	❌	❌	❌	❌
AlpacaEval	❌	✅	❌	❌	❌	❌
Arena-Hard-Auto	❌	❌	✅	❌	❌	❌
Lighteval	✅	❌	❌	❌	❌	❌
Evalchemy	✅	✅	❌	❌	❌	❌
OpenJury	➡️ SOON	✅	✅	✅	✅	✅

OpenJury

An open library to run LLM judges benchmarks

- How to run LLM judges for AlpacaEval, Arena-Hard …
- With local judges, OpenRouter, vLLM …
- With tuned LLM judge prompt



https://github.com/OpenEuroLLM/OpenJury

Framework	MT-Bench	AlpacaEval	Arena-Hard	M-Arena-Hard	Tuned judge configuration	Support vLLM Judges
FastChat	✅	❌	❌	❌	❌	❌
AlpacaEval	❌	✅	❌	❌	❌	❌
Arena-Hard-Auto	❌	❌	✅	❌	❌	❌
Lighteval	✅	❌	❌	❌	❌	❌
Evalchemy	✅	✅	❌	❌	❌	❌
OpenJury	➡️ SOON	✅	✅	✅	✅	✅

OpenJury

Contributing & helping

README | Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

OpenJury

Contributing & helping

- OpenJury goal:

README | Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

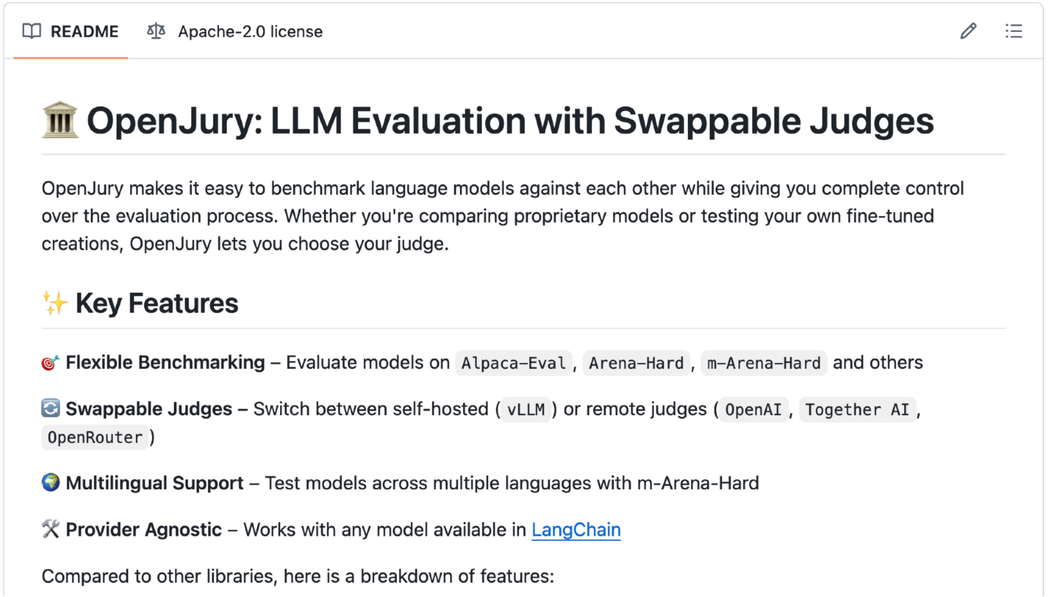
Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:


README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

Notifications Fork 2 Star 4

OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:
  - Start the project if you think its relevant for you => help visibility with Github 🌟

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

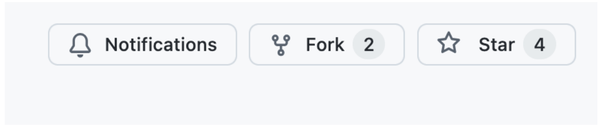
🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

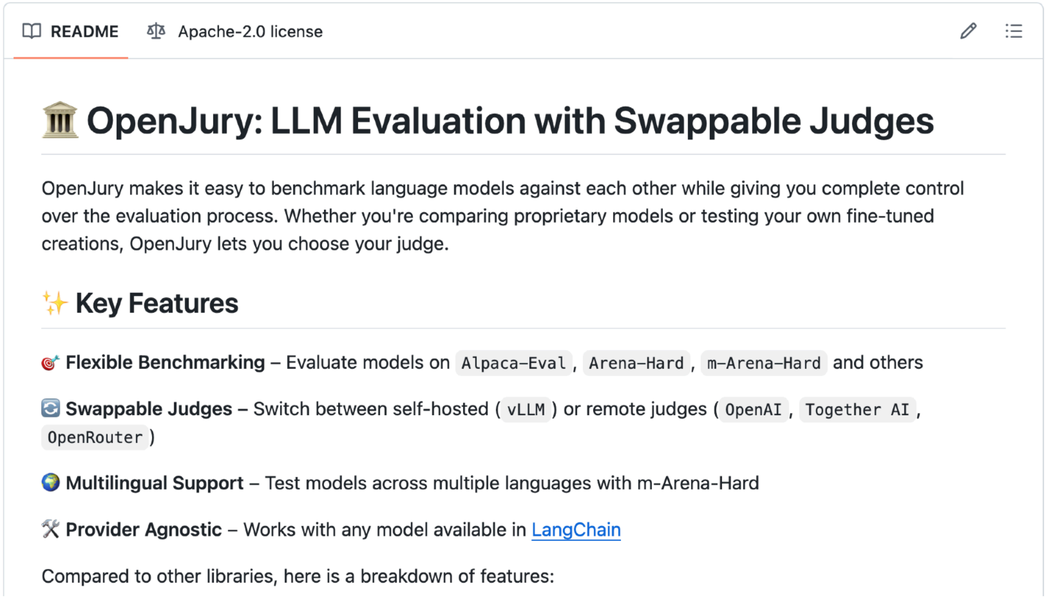
https://github.com/OpenEuroLLM/OpenJury



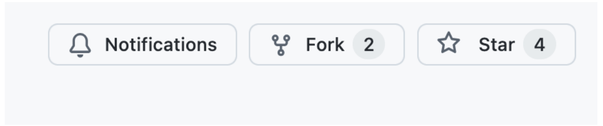
OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:
  - Start the project if you think its relevant for you => help visibility with Github 🌟
  - Try it out and give feedback 🤗, open an issue if something does not work, …



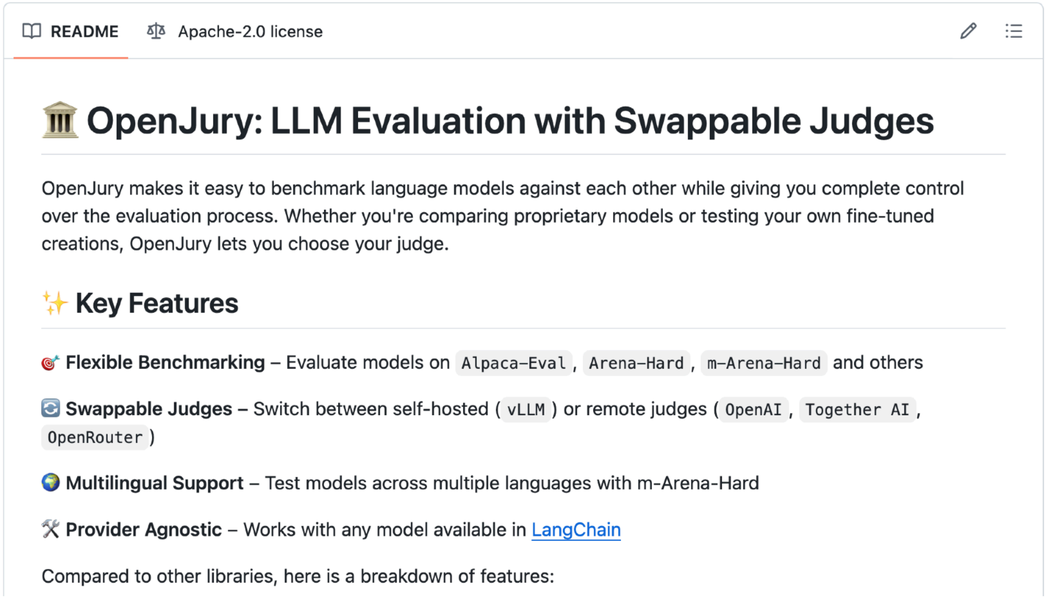
https://github.com/OpenEuroLLM/OpenJury



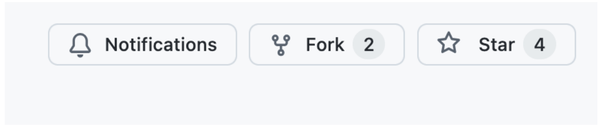
OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:
  - Start the project if you think its relevant for you => help visibility with Github 🌟
  - Try it out and give feedback 🤗, open an issue if something does not work, …
  - Contribute documentations, features, PRs welcome 🥰



https://github.com/OpenEuroLLM/OpenJury



OpenJury

Contributing & helping

- OpenJury goal:
  - Fully open & reproducible LLM judges pipelines for the community
  - With first class citizen for open-weight models!
- You can help & contribute in many ways:
  - Start the project if you think its relevant for you => help visibility with Github 🌟
  - Try it out and give feedback 🤗, open an issue if something does not work, …
  - Contribute documentations, features, PRs welcome 🥰

README Apache-2.0 license

🏛️ OpenJury: LLM Evaluation with Swappable Judges

OpenJury makes it easy to benchmark language models against each other while giving you complete control over the evaluation process. Whether you're comparing proprietary models or testing your own fine-tuned creations, OpenJury lets you choose your judge.

✨ Key Features

🎯 **Flexible Benchmarking** – Evaluate models on `Alpaca-Eval`, `Arena-Hard`, `m-Arena-Hard` and others

🔄 **Swappable Judges** – Switch between self-hosted (`vLLM`) or remote judges (`OpenAI`, `Together AI`, `OpenRouter`)

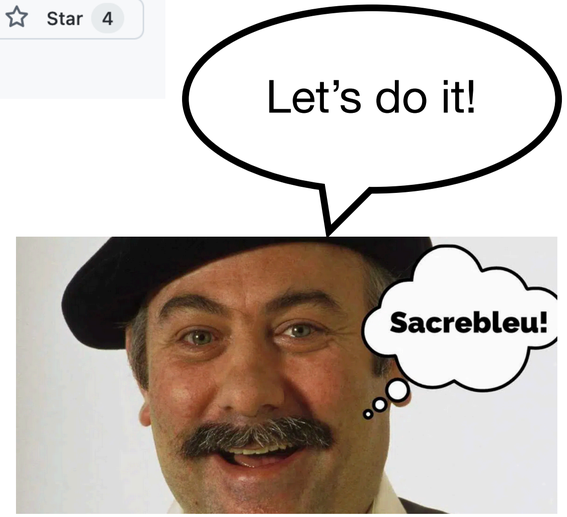
🌍 **Multilingual Support** – Test models across multiple languages with m-Arena-Hard

🛠️ **Provider Agnostic** – Works with any model available in LangChain

Compared to other libraries, here is a breakdown of features:

https://github.com/OpenEuroLLM/OpenJury

Notifications Fork 2 Star 4



Evaluation

Wrapping up

Evaluation

Wrapping up

- Evaluating LLM is hard:

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high
- Tradeoff must be made, for instance cost vs accuracy

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high
- Tradeoff must be made, for instance cost vs accuracy
- Still early days, lot of low hanging fruits

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high
- Tradeoff must be made, for instance cost vs accuracy
- Still early days, lot of low hanging fruits
  - How to achieve good tradeoffs between objectives

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high
- Tradeoff must be made, for instance cost vs accuracy
- Still early days, lot of low hanging fruits
  - How to achieve good tradeoffs between objectives
  - More sample efficient evaluations

Evaluation

Wrapping up

- Evaluating LLM is hard:
  - Many languages & objectives
  - Fundamental hardness to evaluate powerful generative models
  - Cost can be high
- Tradeoff must be made, for instance cost vs accuracy
- Still early days, lot of low hanging fruits
  - How to achieve good tradeoffs between objectives
  - More sample efficient evaluations
  - Better ways to improve low-resource languages & provide less culturally biased benchmarks

Thank you! Time for questions :-)







Previous slides

Multiobjective and multi fidelity optimization

Preamble

Multiobjective & multifidelity optimization

Preamble

Multiobjective & multifidelity optimization

- Who is familiar with

Preamble

Multiobjective & multifidelity optimization

- Who is familiar with
  - multiobjective optimization? 🤔

Preamble

Multiobjective & multifidelity optimization

- Who is familiar with
  - multiobjective optimization? 🤔
  - multifidelity optimization? 🤔

Multiobjective optimization

Multiobjective optimization

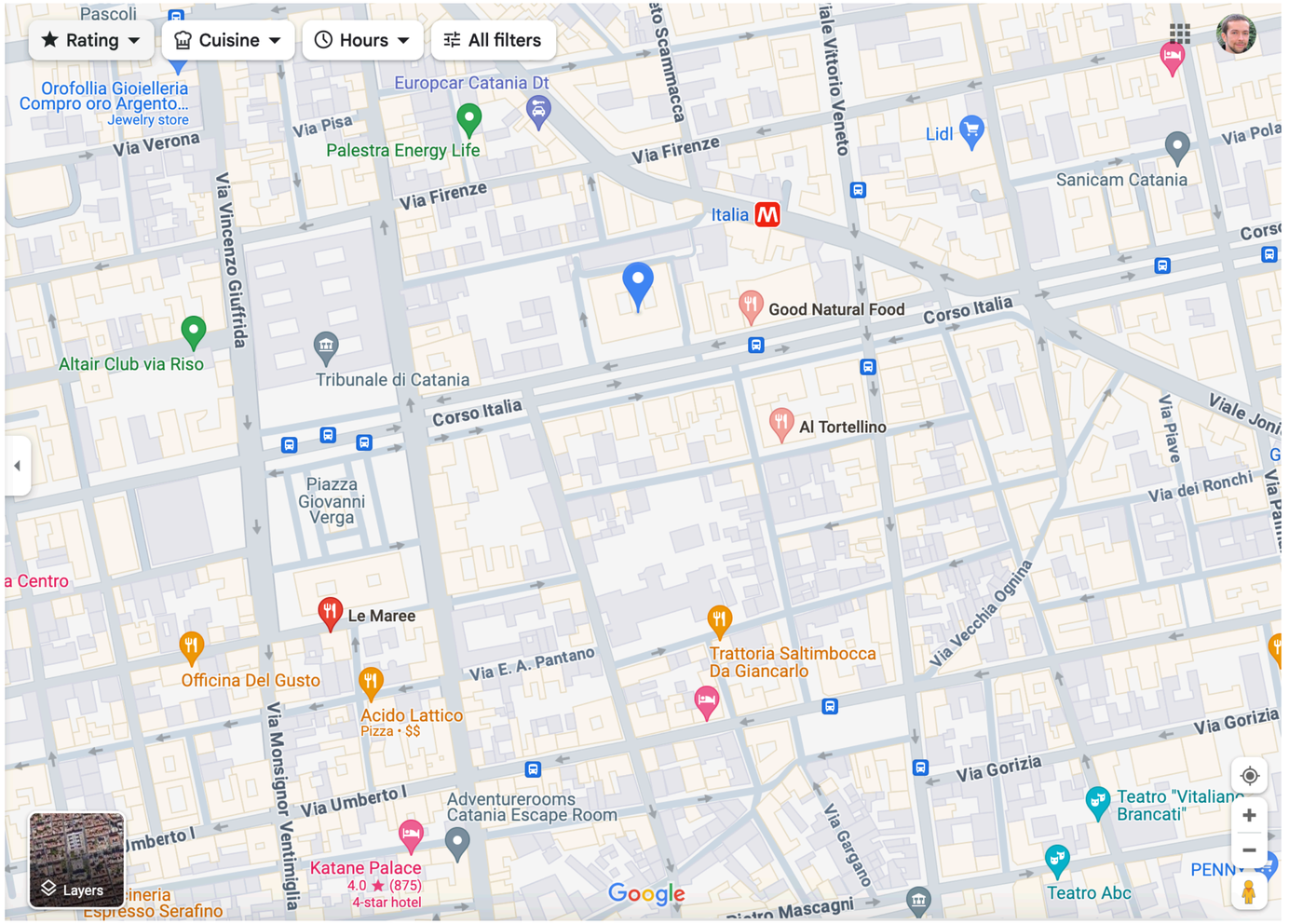
- Optimization is often about optimising a single objective but what if we have many?

Multiobjective optimization

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…

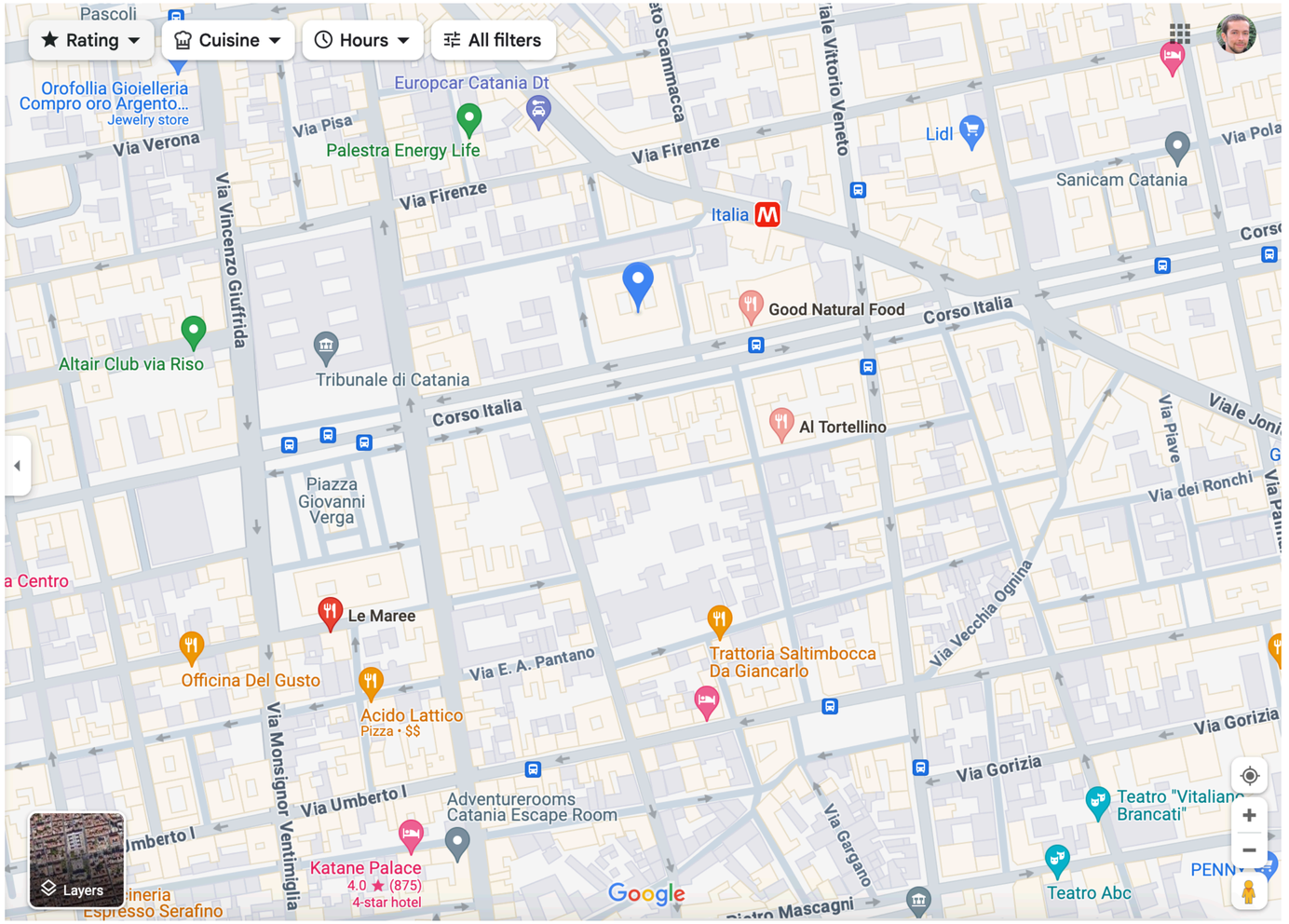
Multiobjective optimization

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…



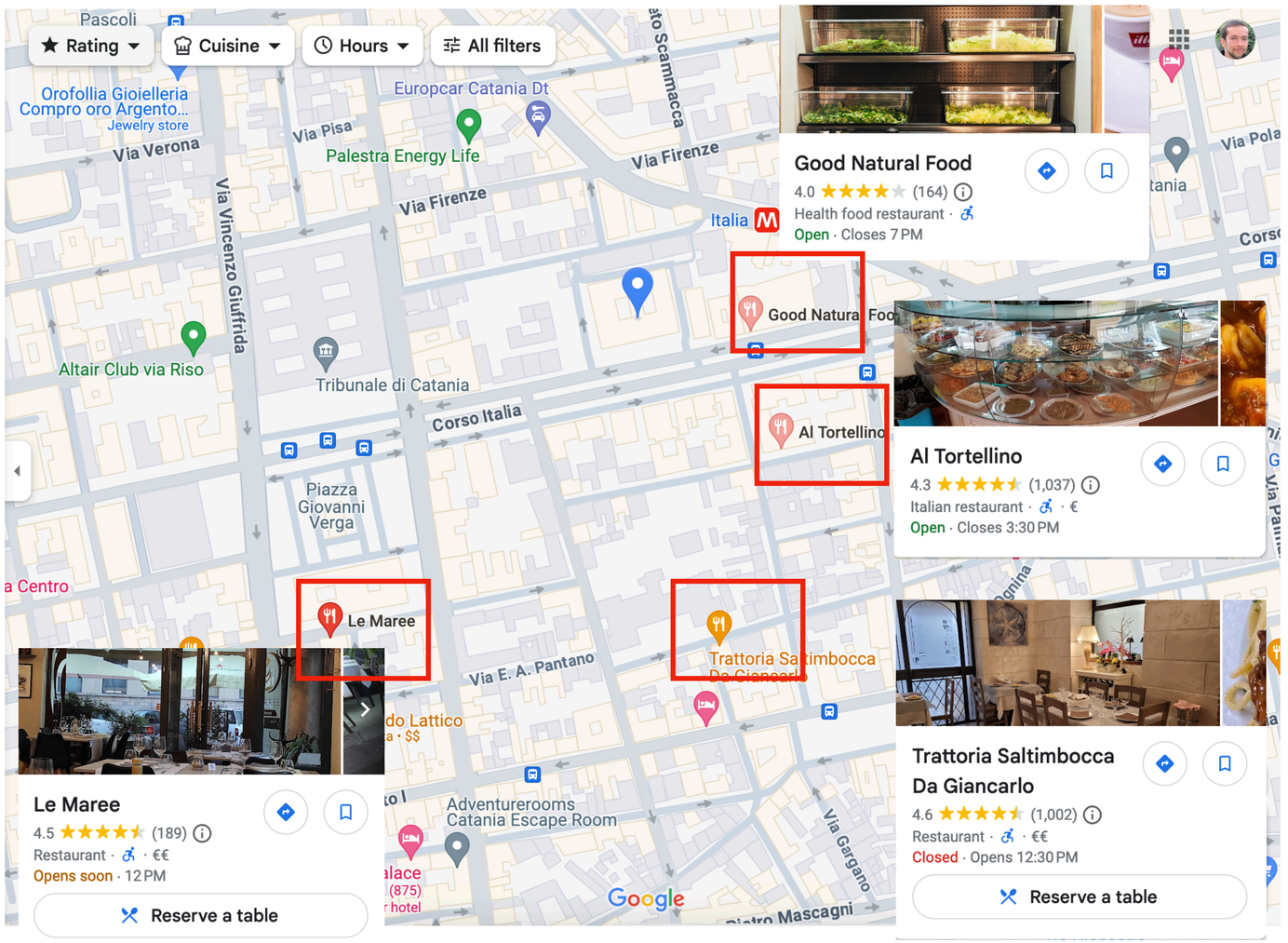
Multiobjective optimization

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance



Multiobjective optimization

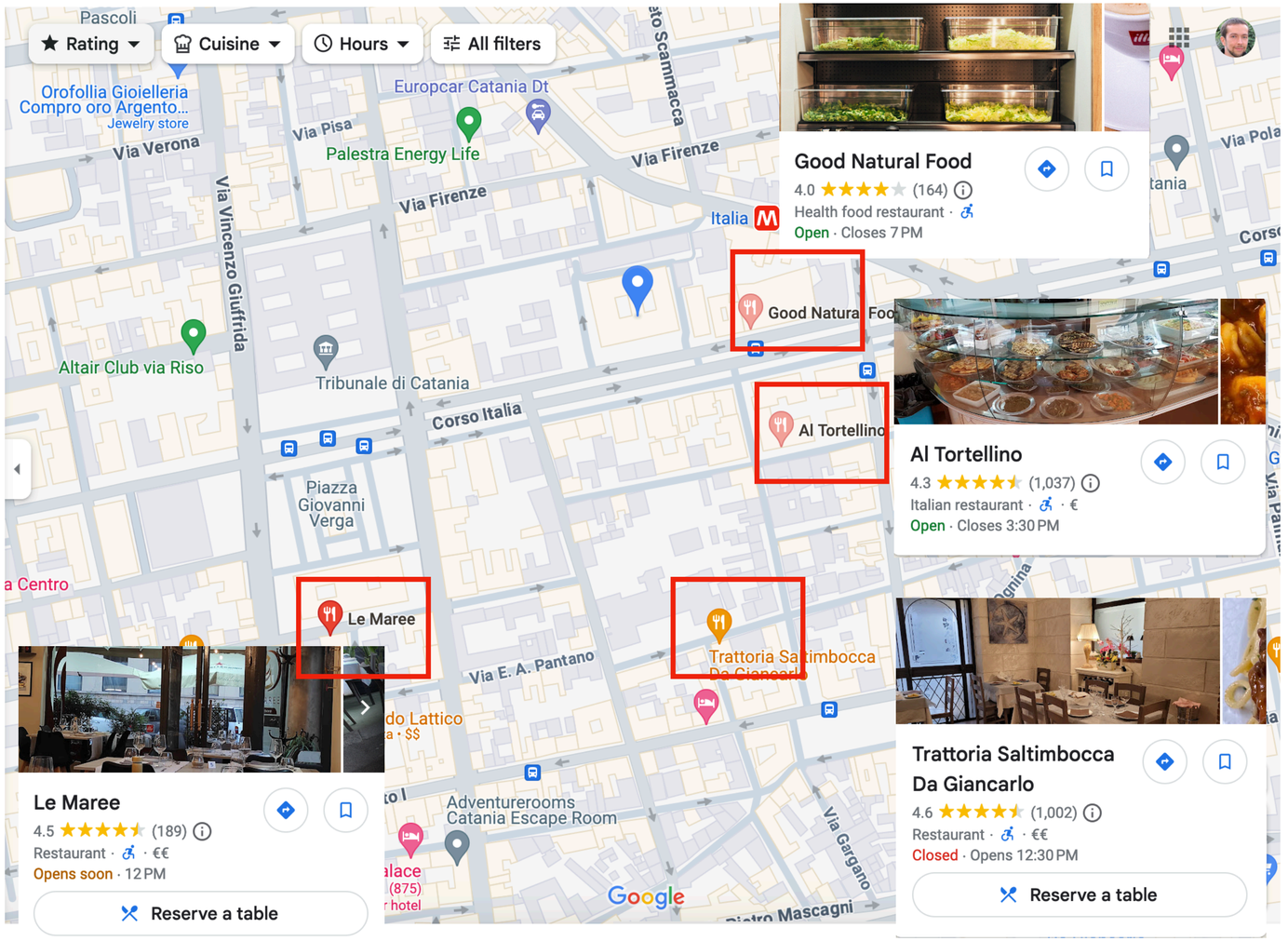
- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance



Multiobjective optimization

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance

🤔 How to select which restaurant to go?



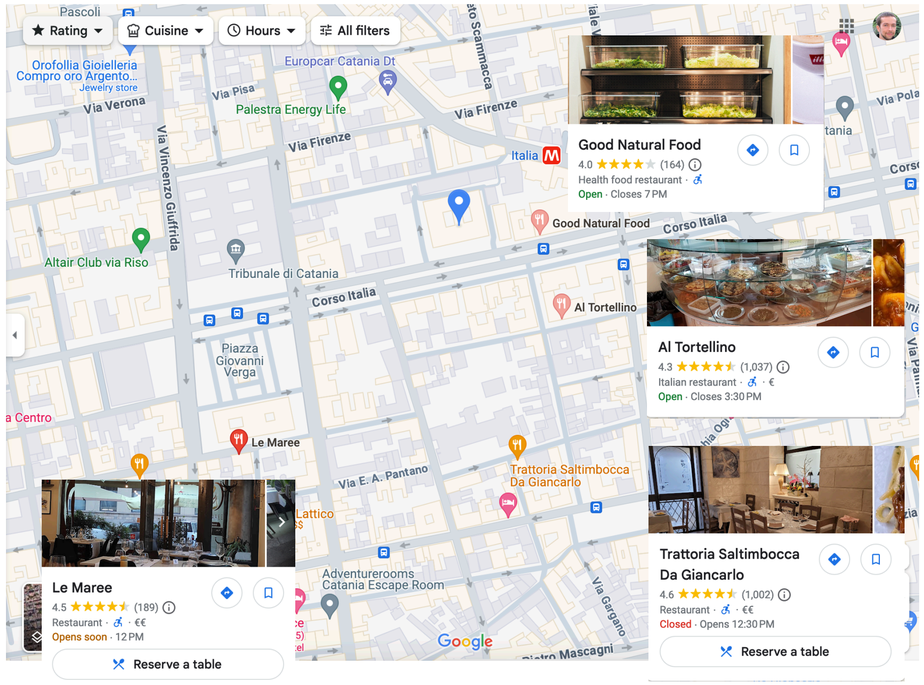
How to select which restaurant to go?

How to select which restaurant to go?

- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives

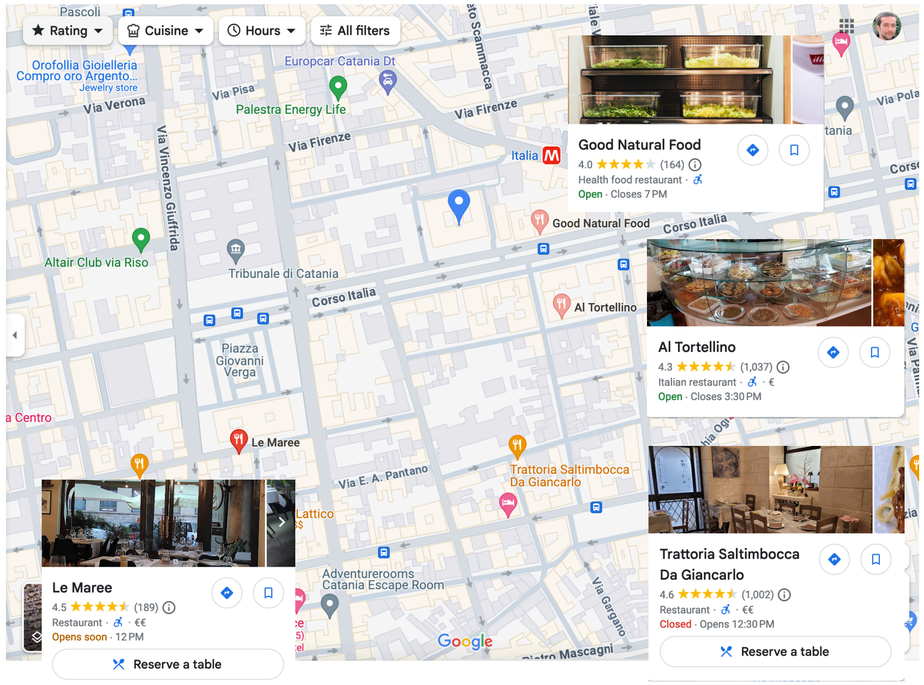
How to select which restaurant to go?

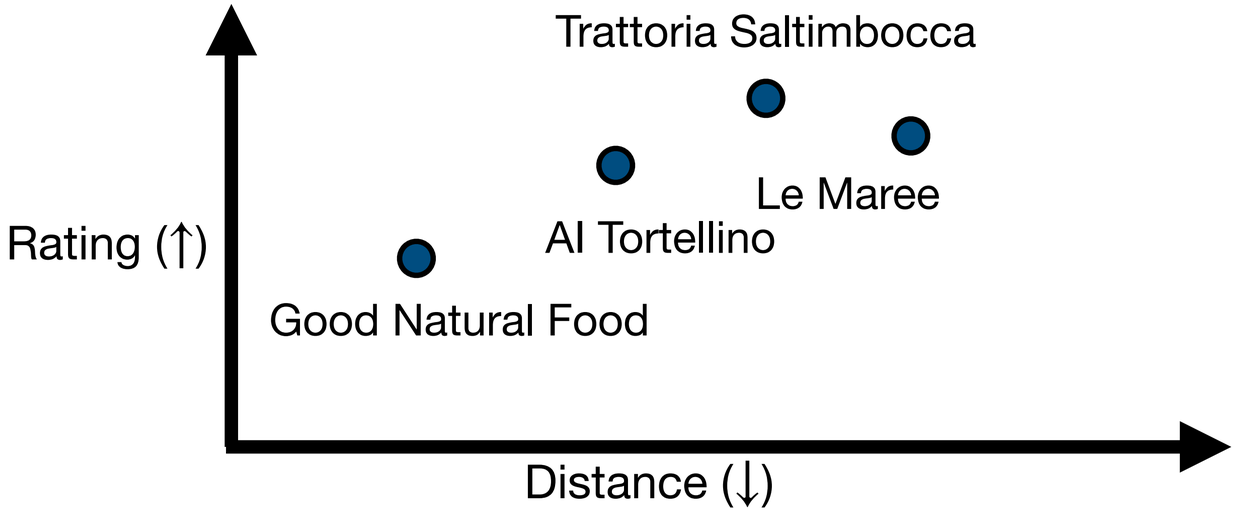
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives



How to select which restaurant to go?

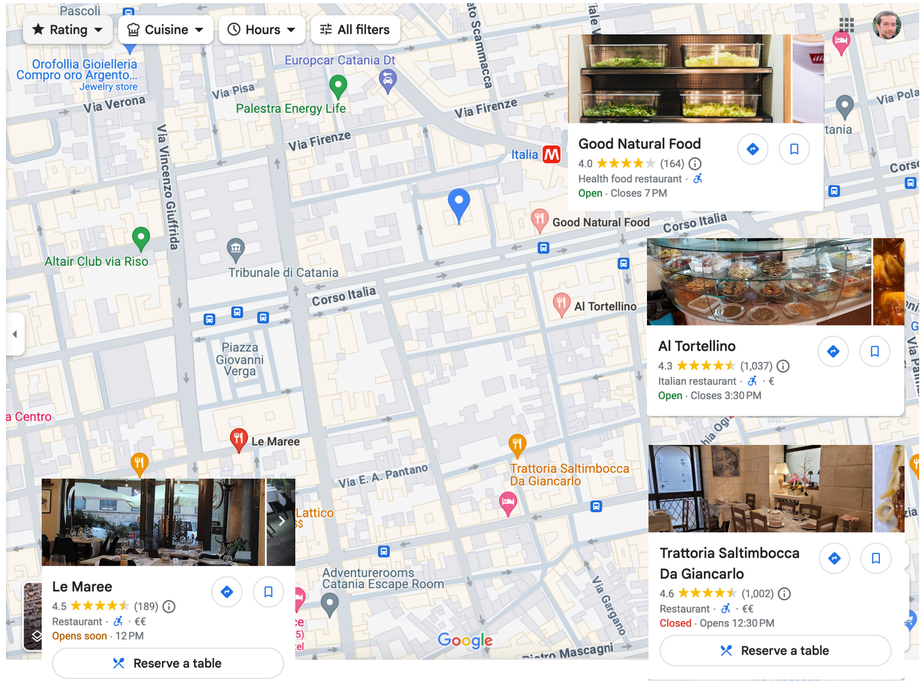
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives

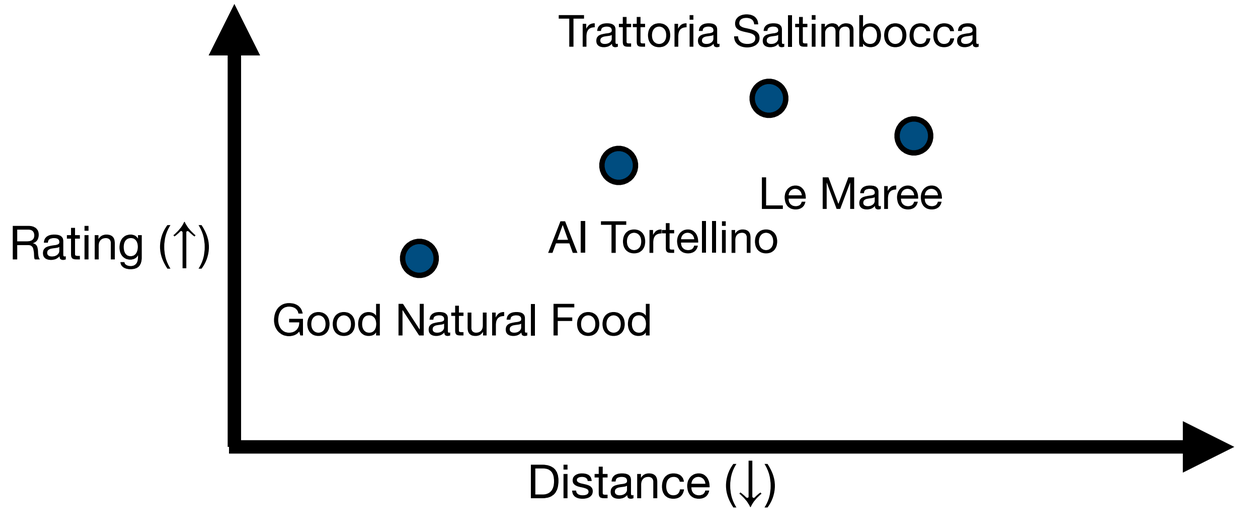




How to select which restaurant to go?

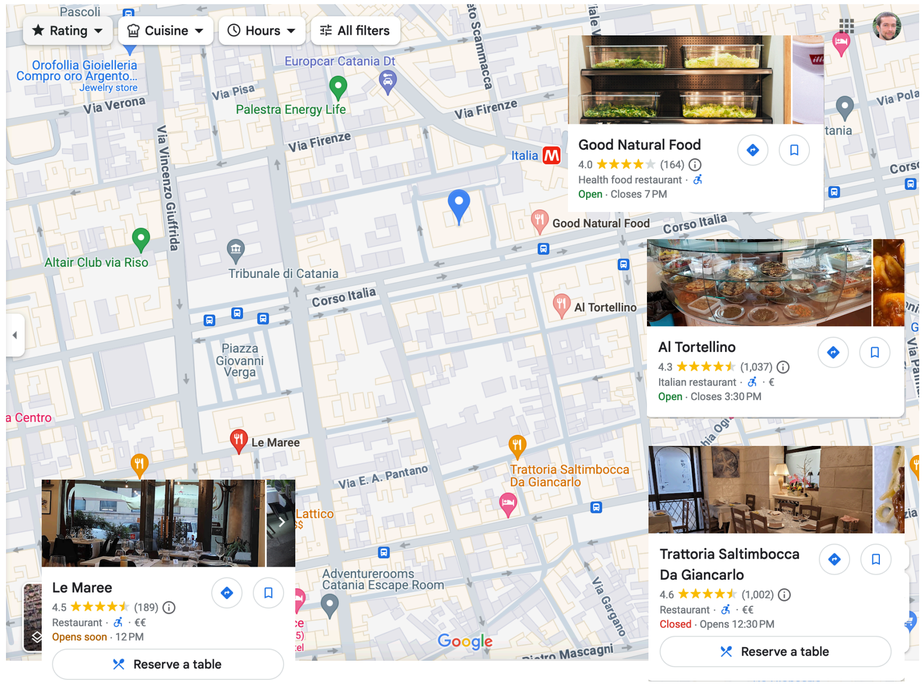
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!

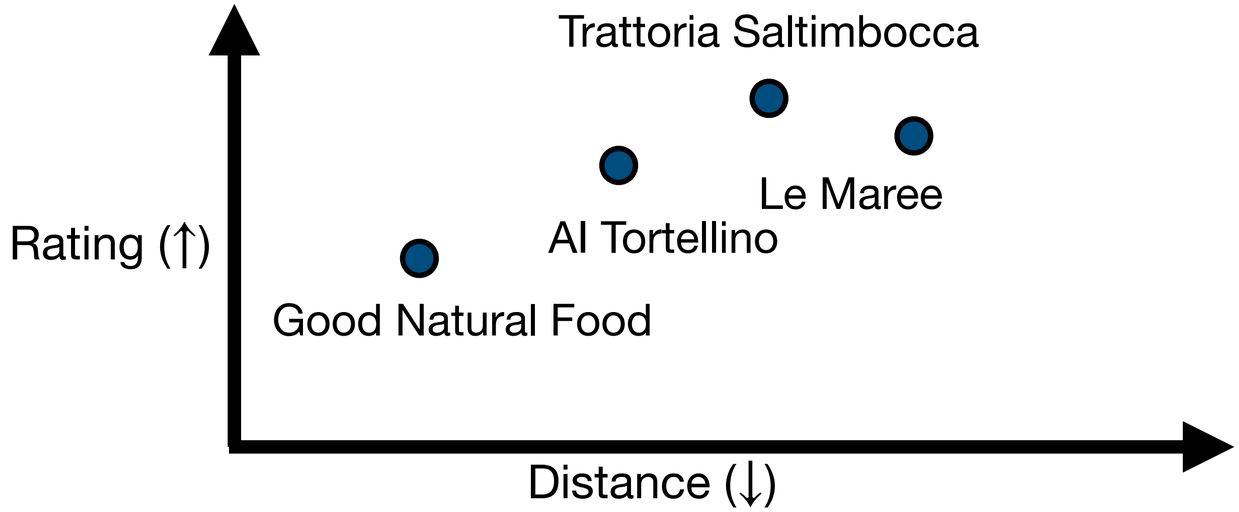




How to select which restaurant to go?

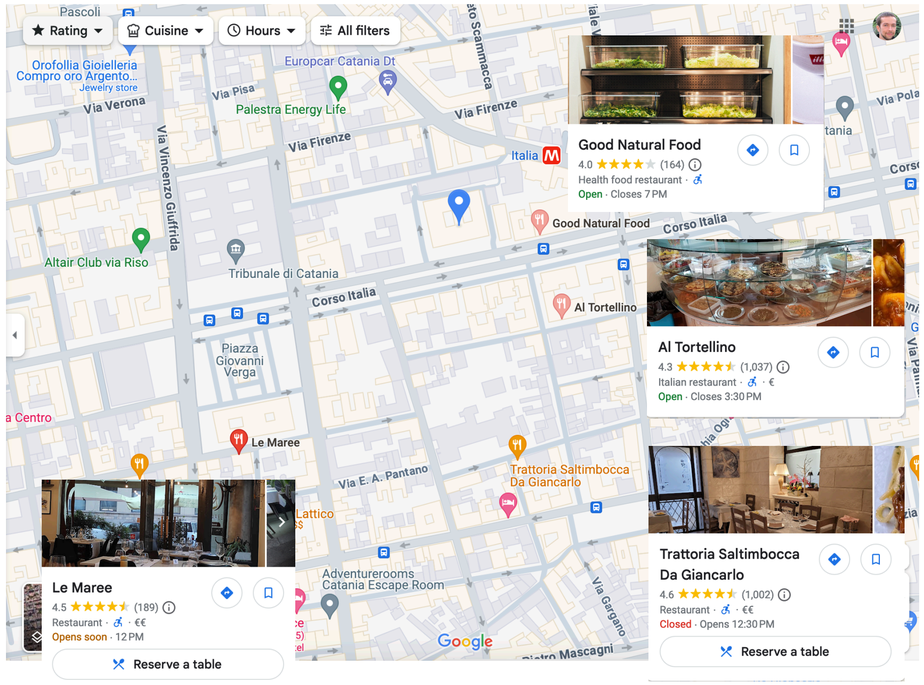
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?

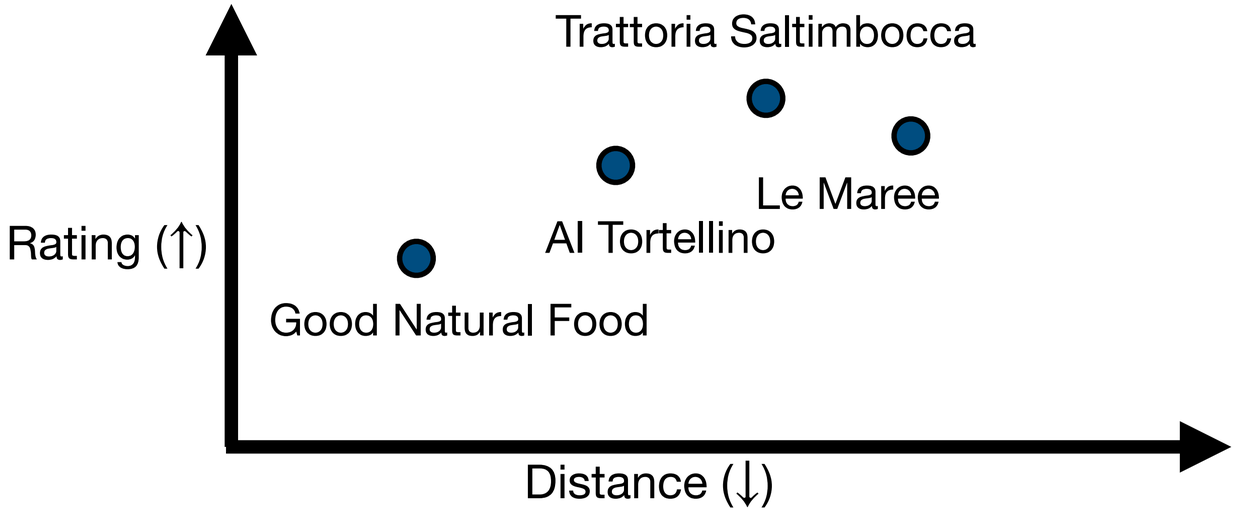




How to select which restaurant to go?

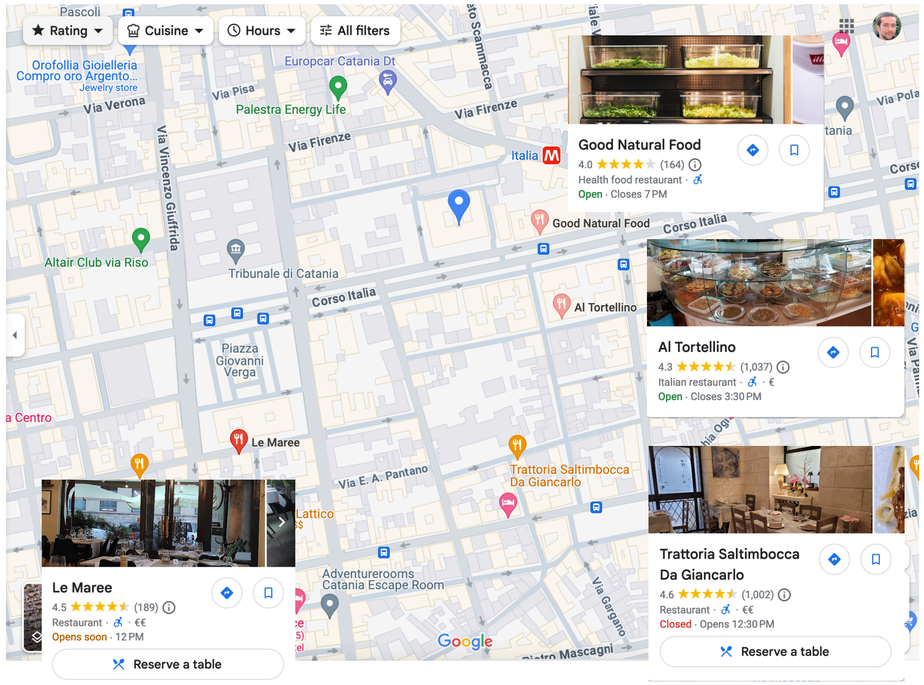
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

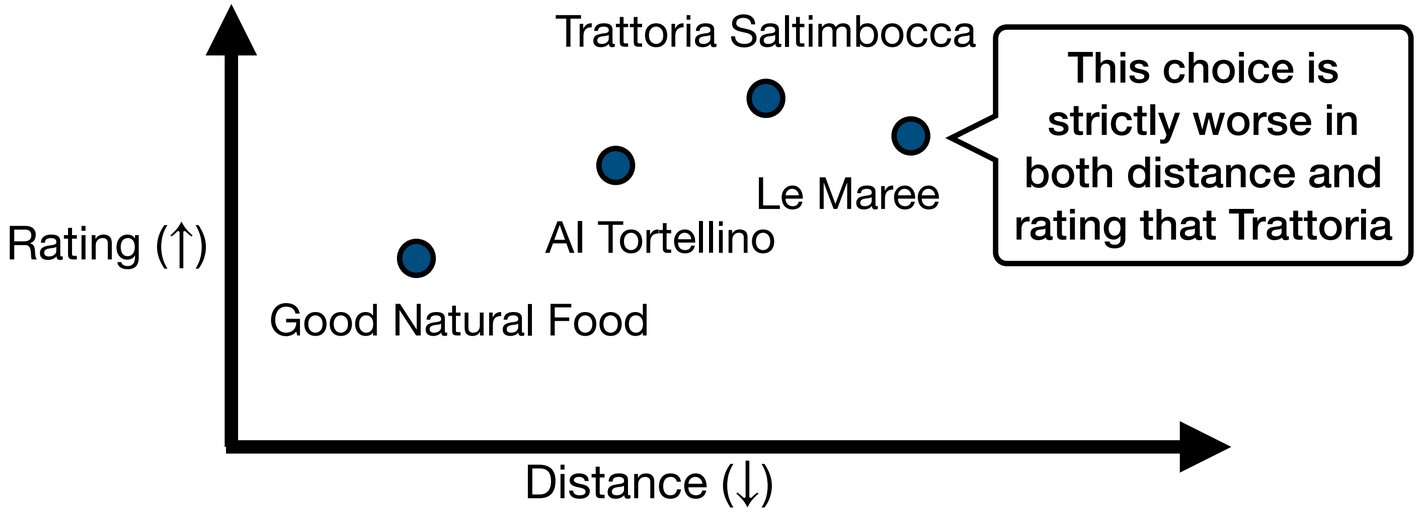




How to select which restaurant to go?

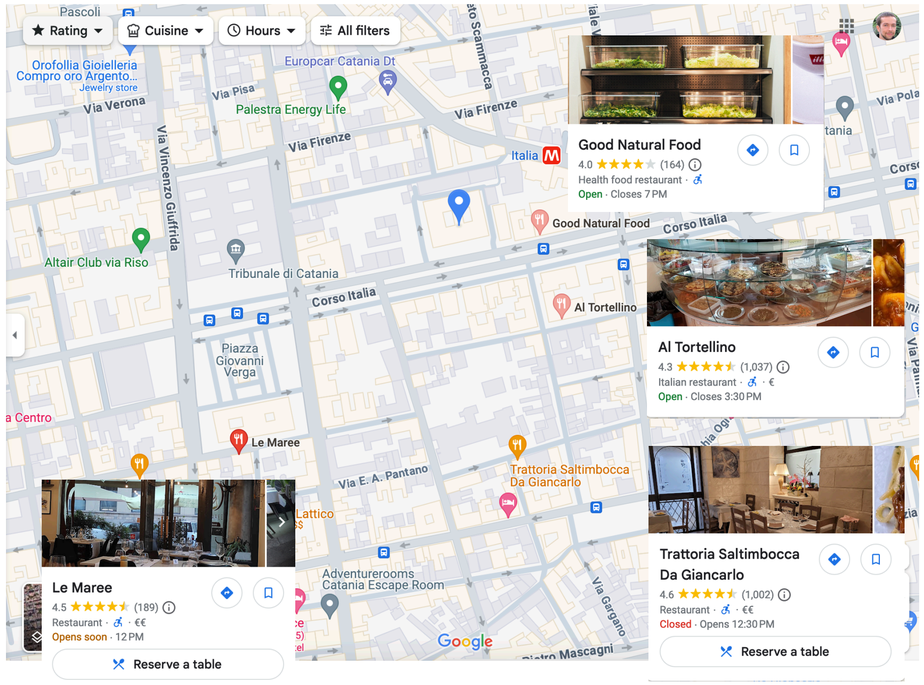
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

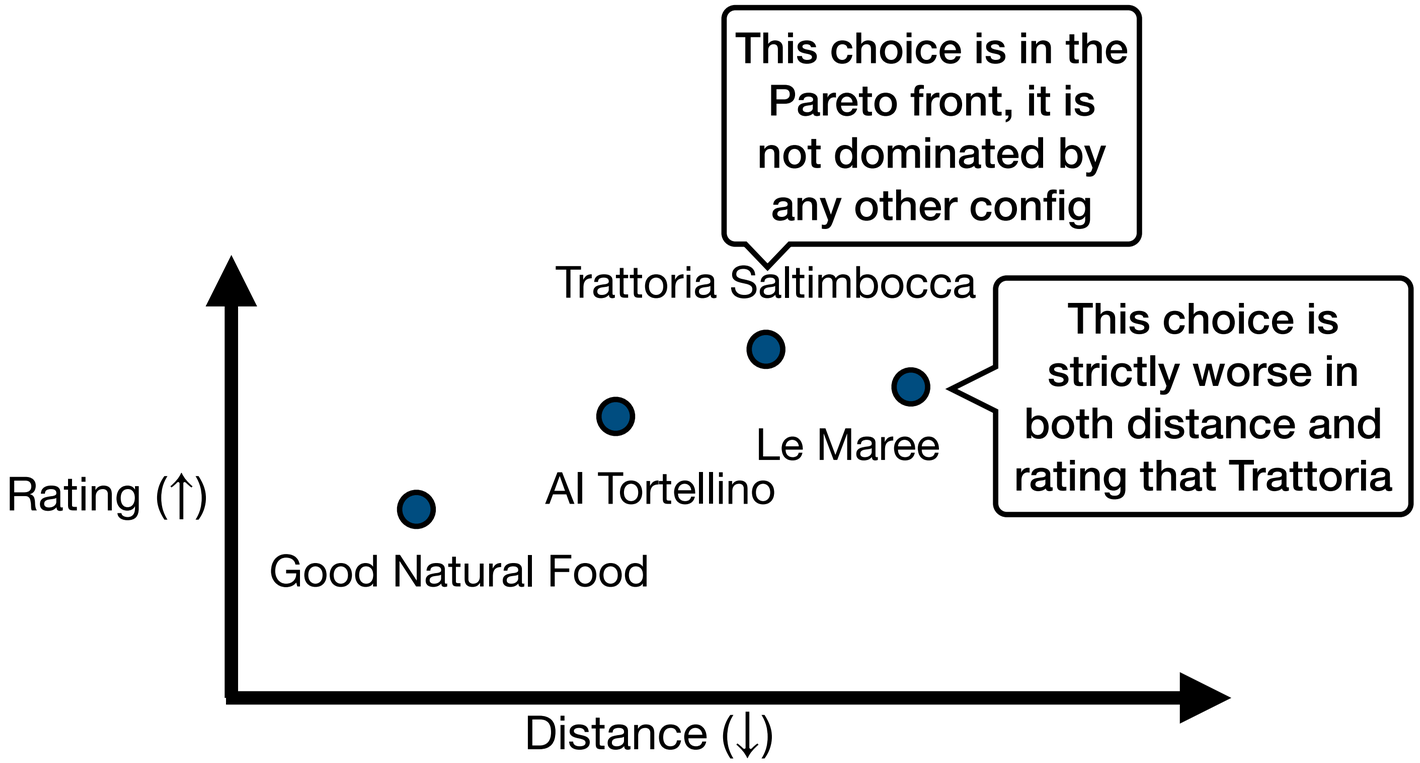




How to select which restaurant to go?

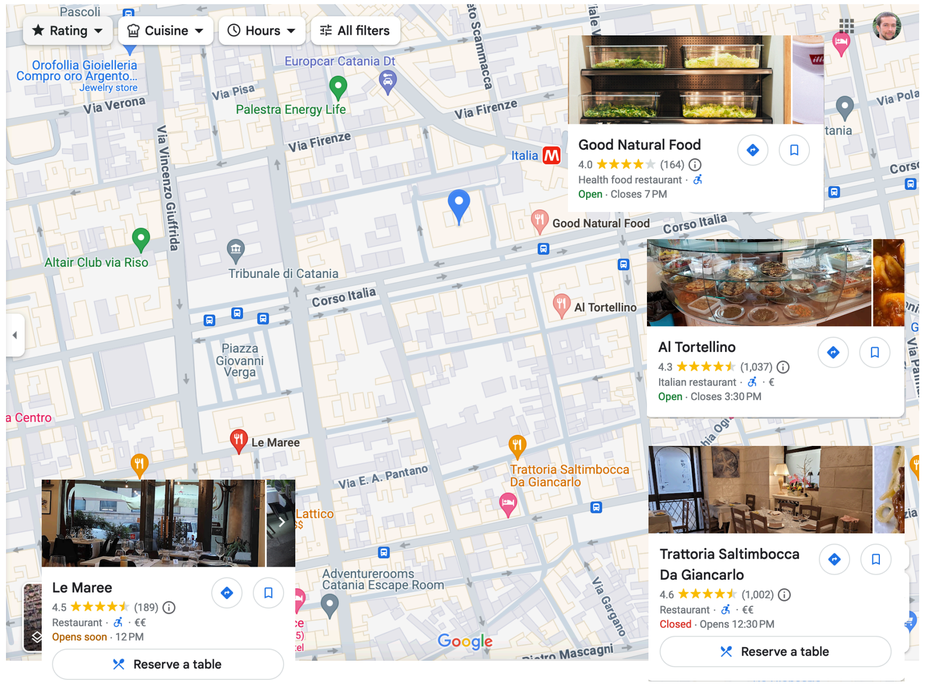
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

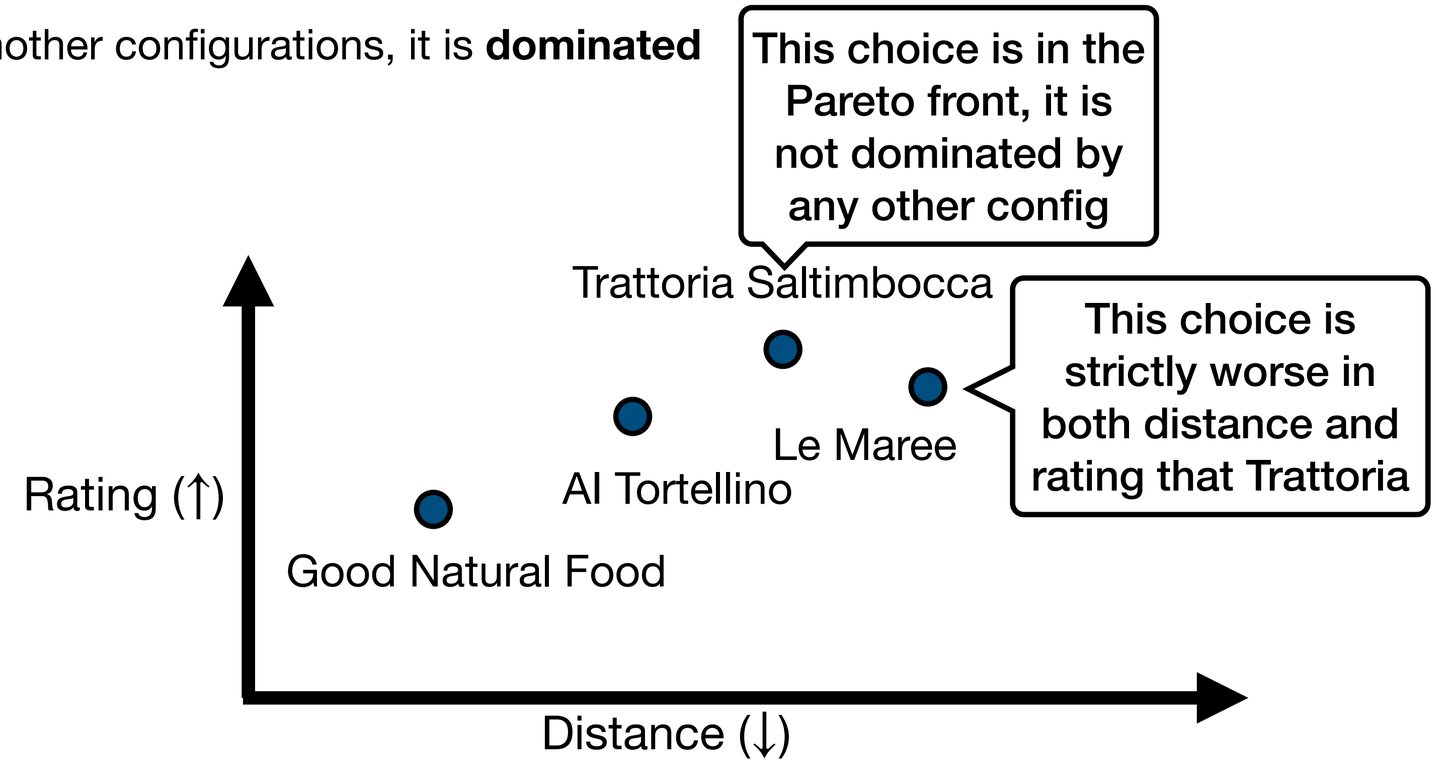




How to select which restaurant to go?

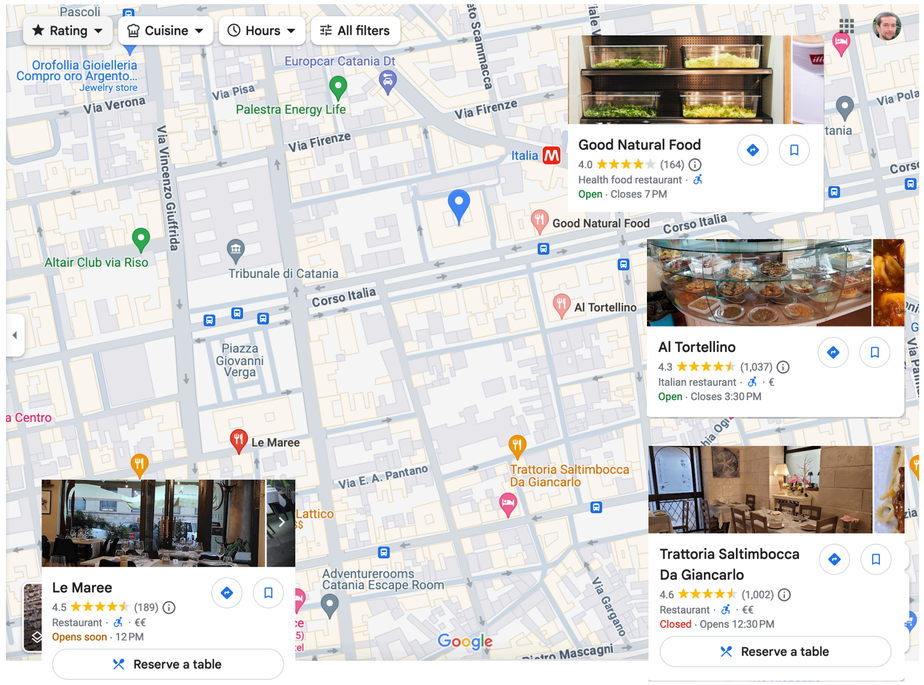
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal
- If one configuration is worse for all metrics than another configurations, it is **dominated**

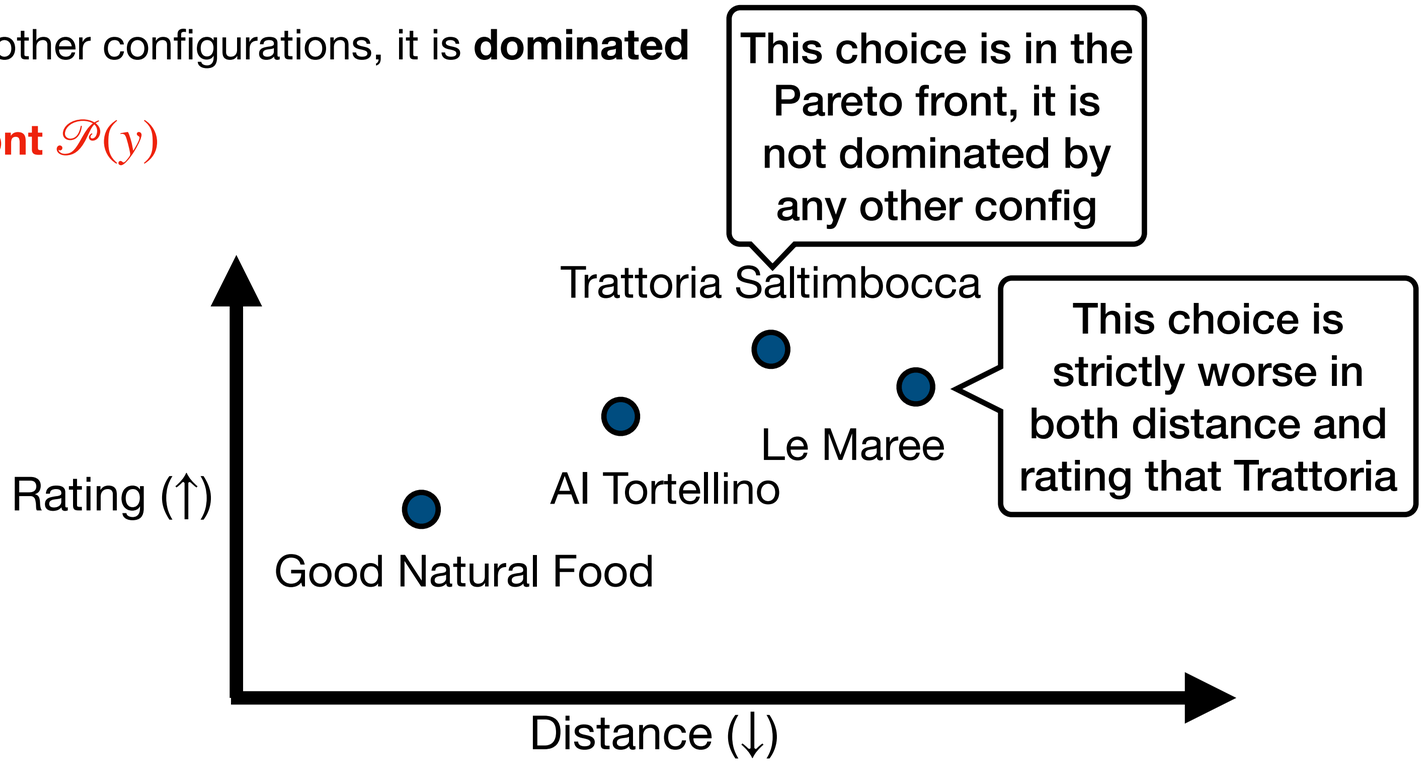




How to select which restaurant to go?

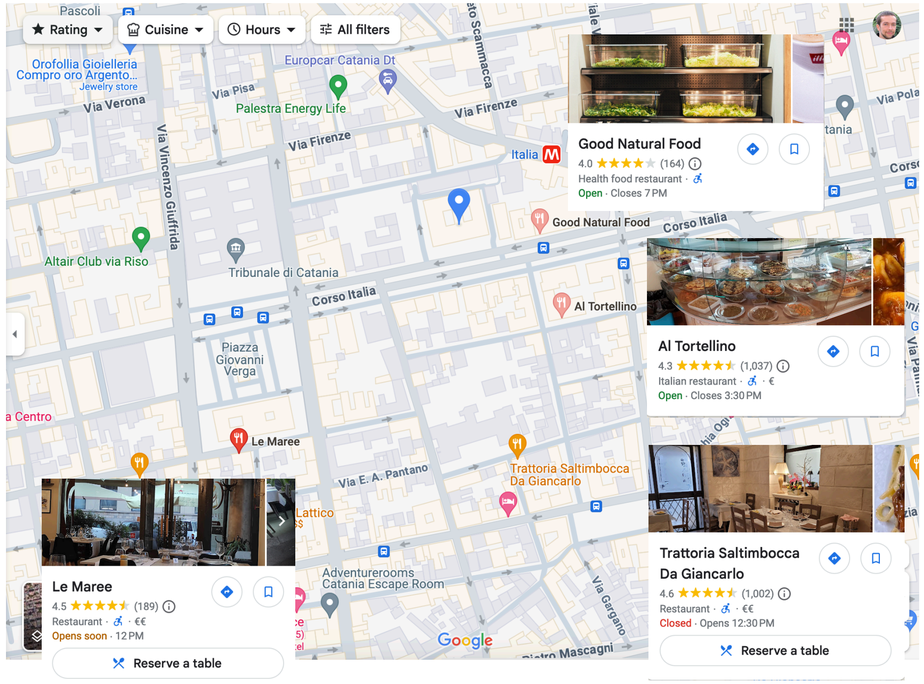
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal
- If one configuration is worse for all metrics than another configurations, it is **dominated**
- The set of non dominated options is the **Pareto front** $\mathcal{P}(y)$

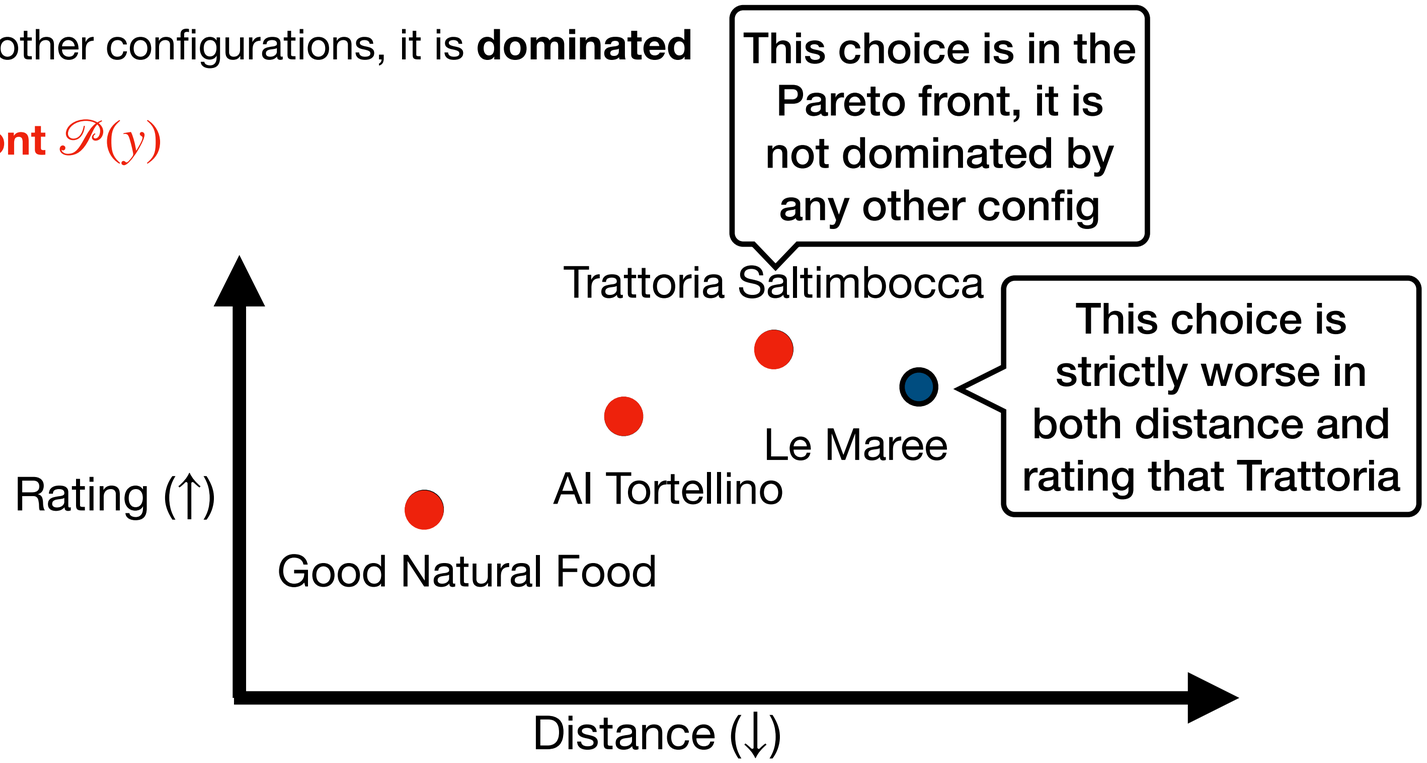




How to select which restaurant to go?

- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal
- If one configuration is worse for all metrics than another configurations, it is **dominated**
- The set of non dominated options is the **Pareto front** $\mathscr{P}(y)$





Pareto front

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.

Pareto front

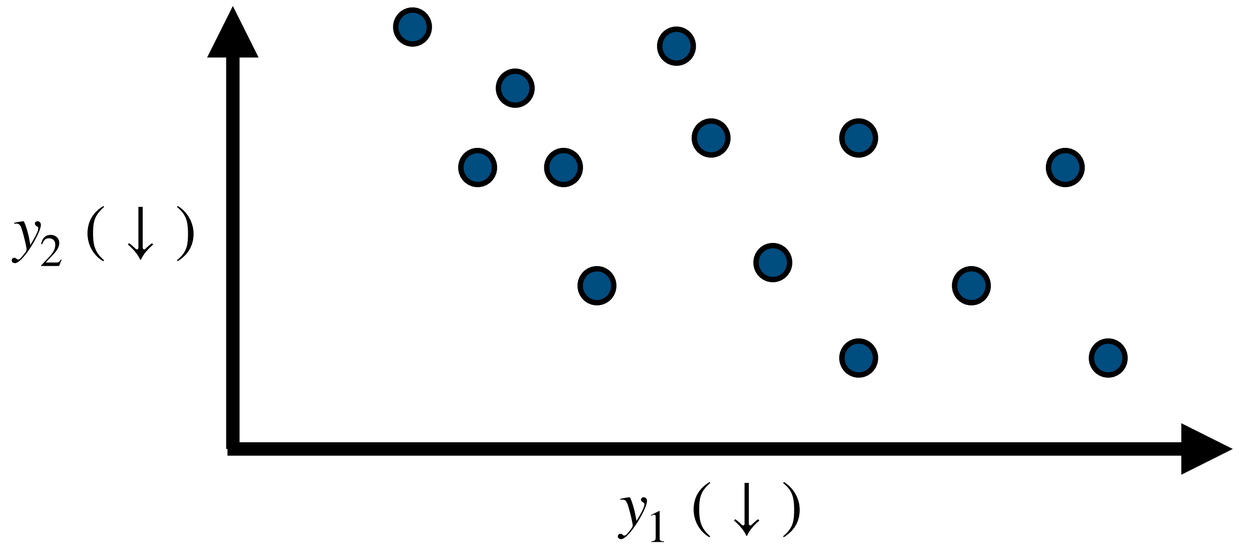
- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔

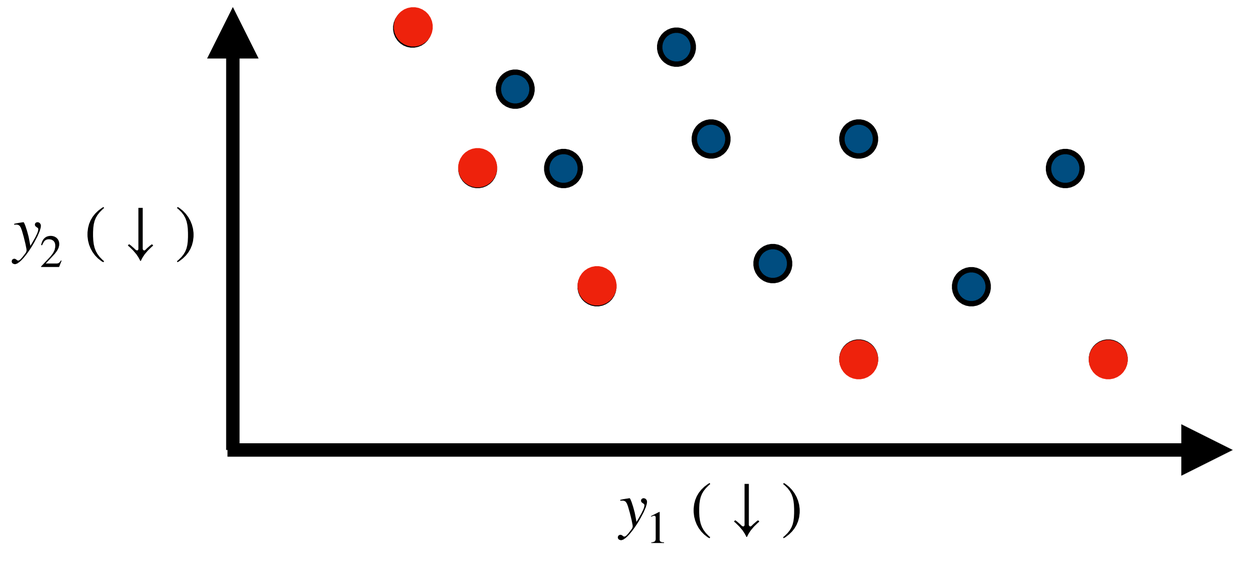
Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔



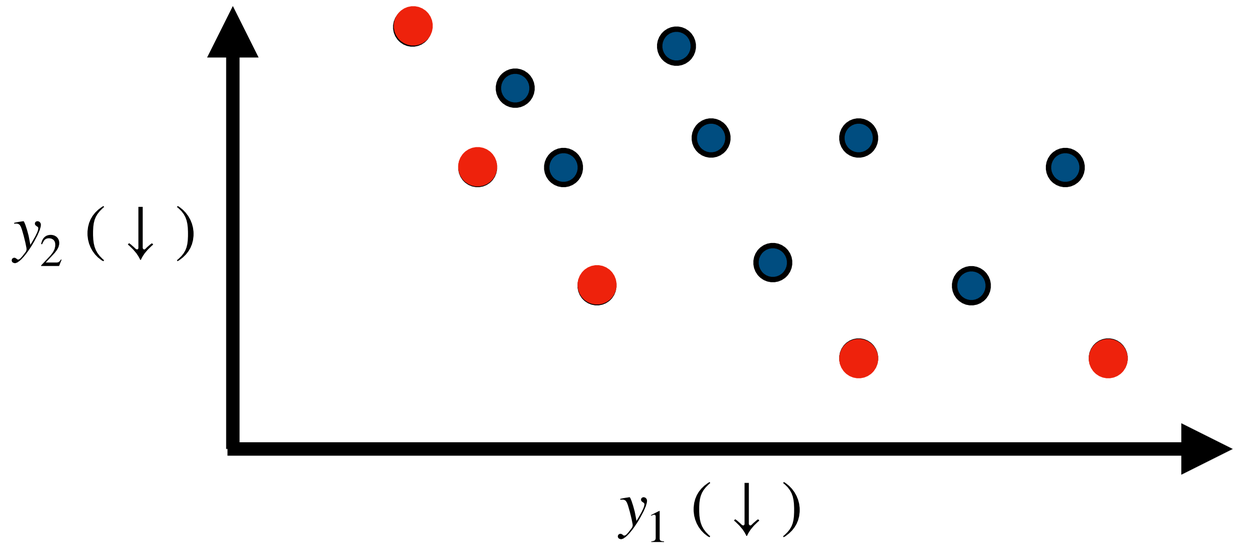
Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔



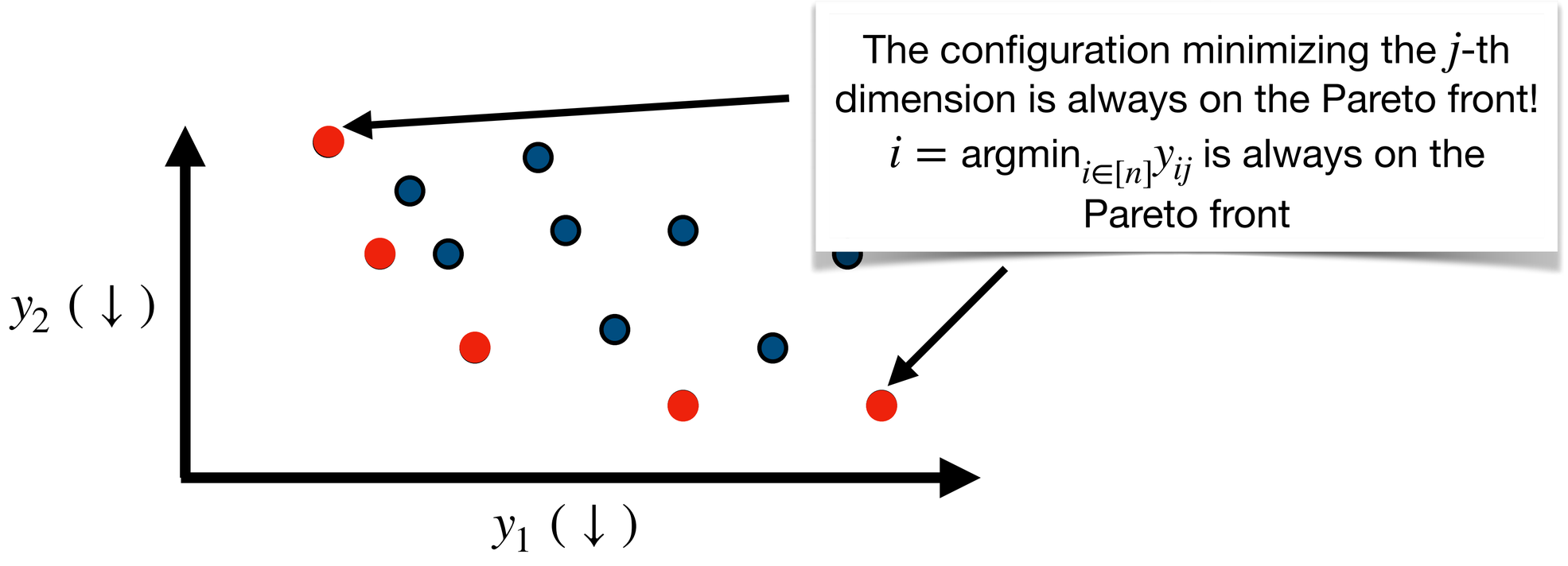
Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathcal{P}(y)$ is the set of non dominated options $\mathcal{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathcal{P}(y)$? 🤔
- 🤔 Given objectives $y \in \mathbb{R}^{n \times d}$, can you give a procedure that finds d configurations that are in the Pareto front $\mathcal{P}(y)$ in $\mathcal{O}(d)$?



Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathcal{P}(y)$ is the set of non dominated options $\mathcal{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathcal{P}(y)$? 🤔
- 🤔 Given objectives $y \in \mathbb{R}^{n \times d}$, can you give a procedure that finds d configurations that are in the Pareto front $\mathcal{P}(y)$ in $\mathcal{O}(d)$?



Multifidelity

Asynchronous Succesful halving (ASHA)

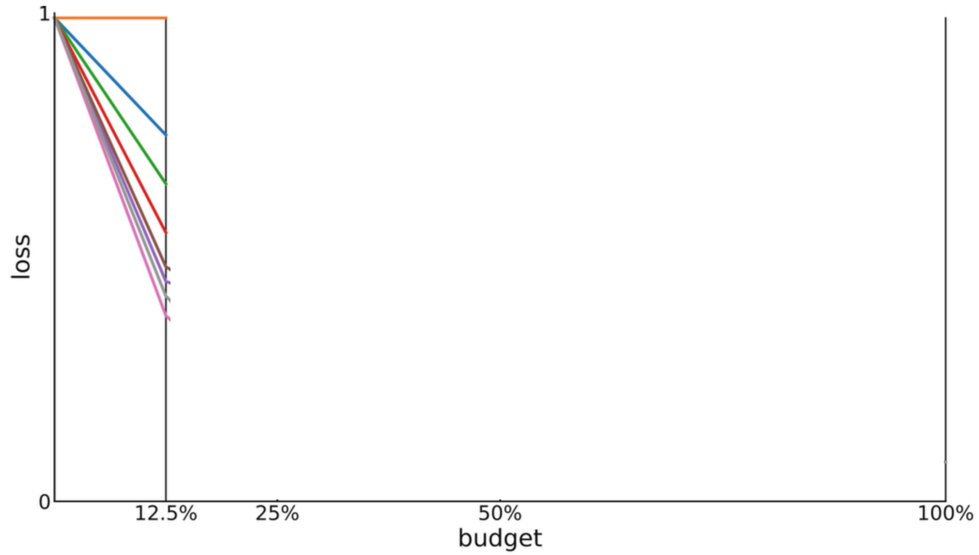


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?

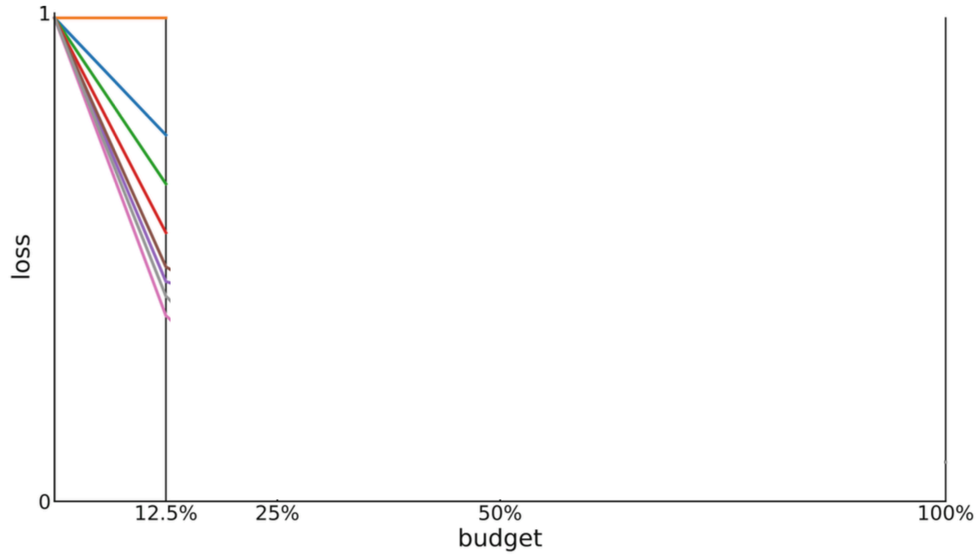


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

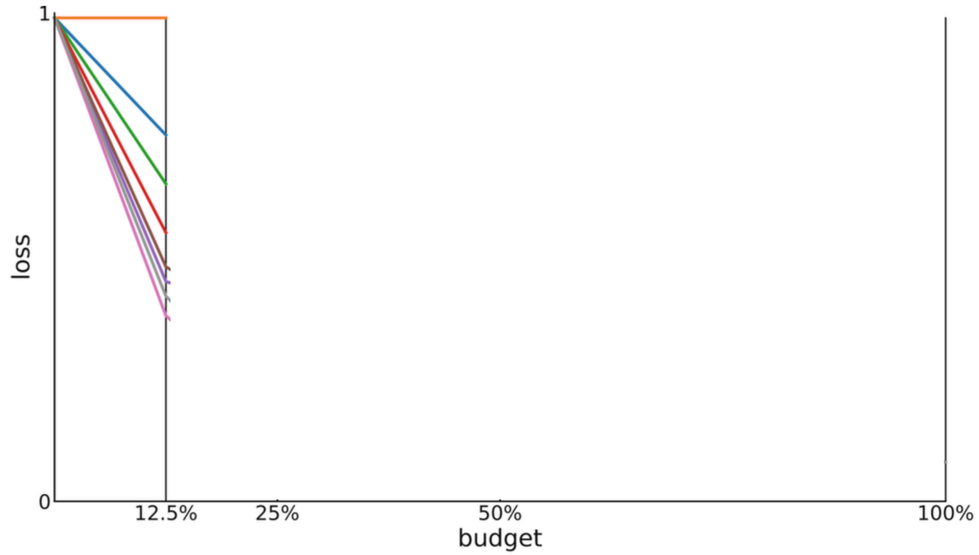


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

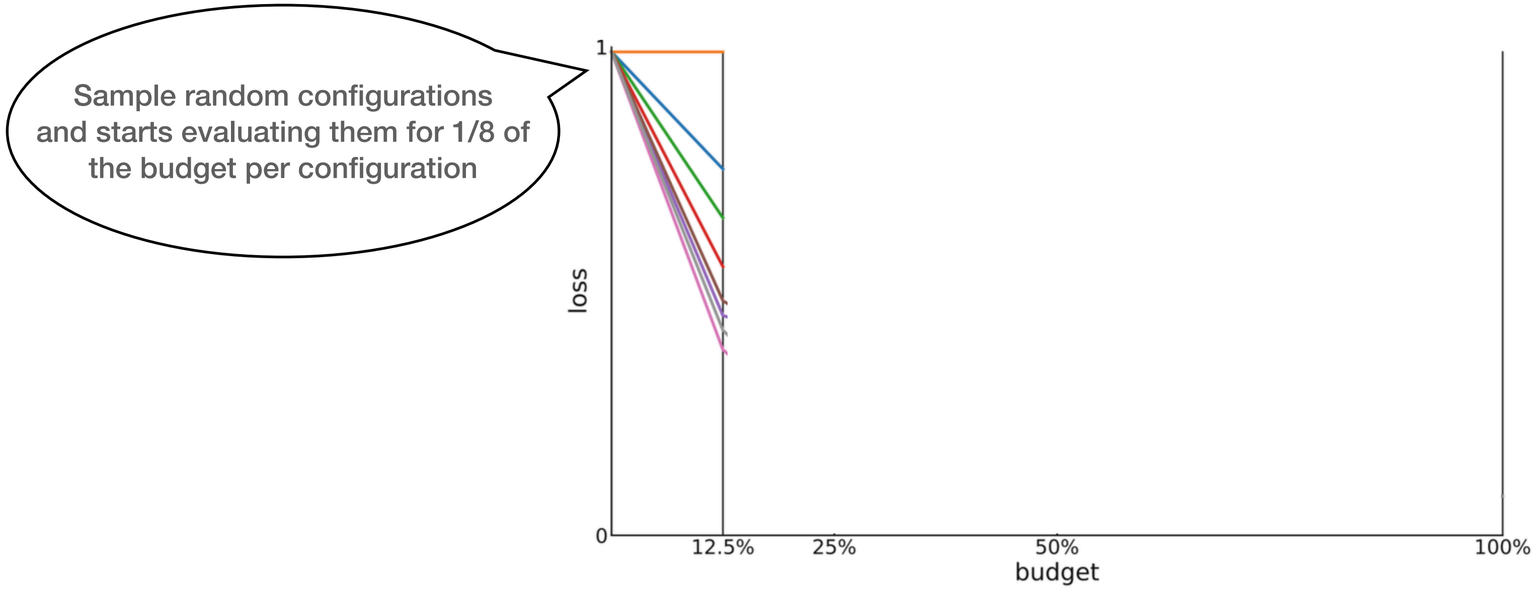


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

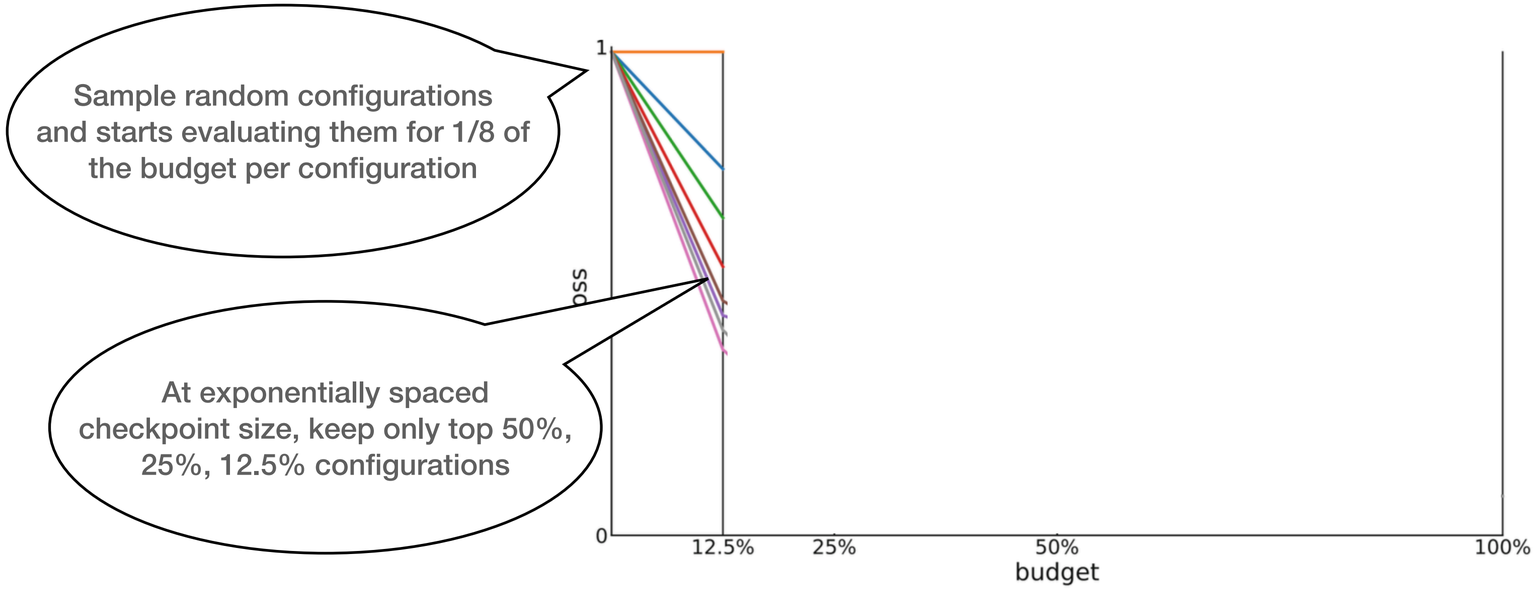


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

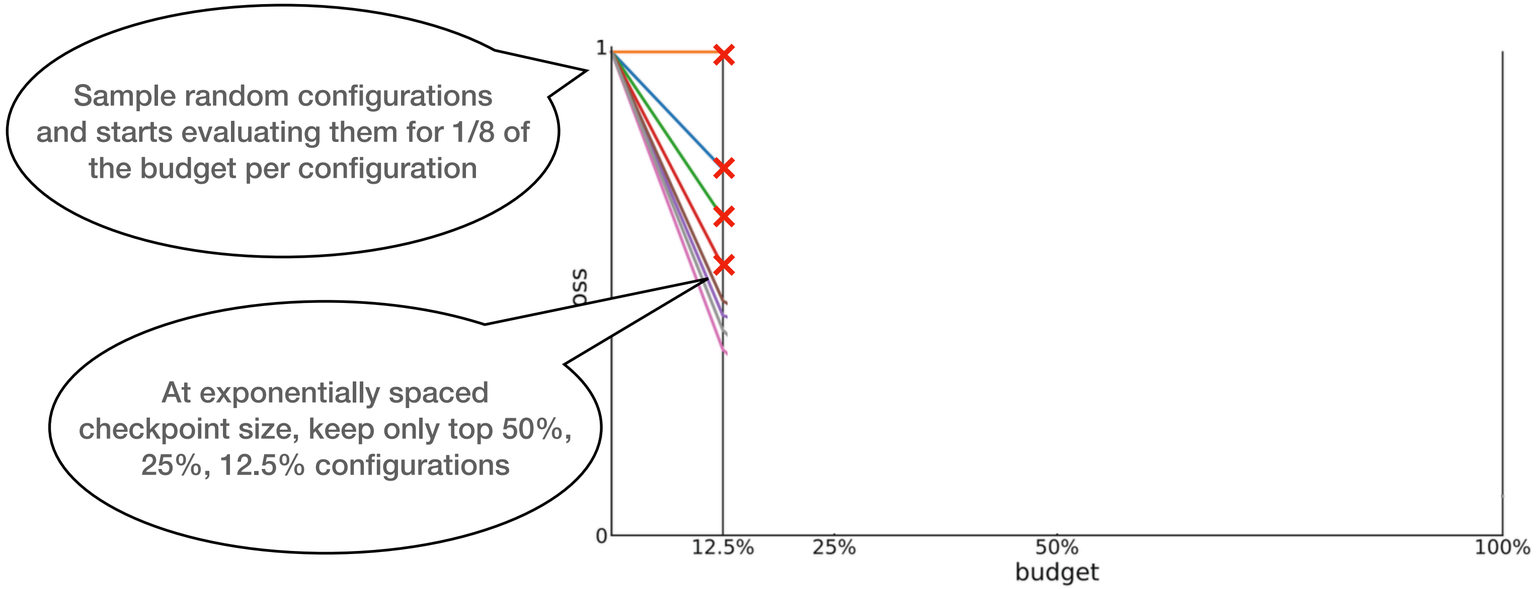


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

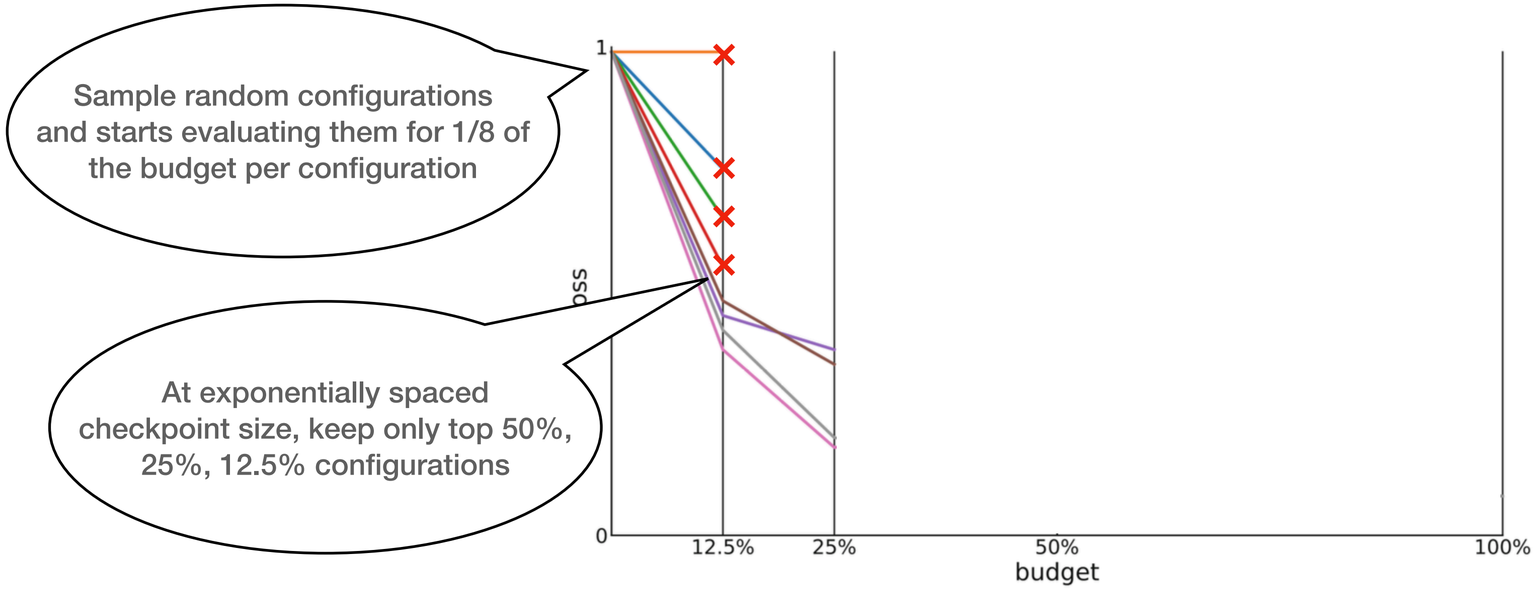


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

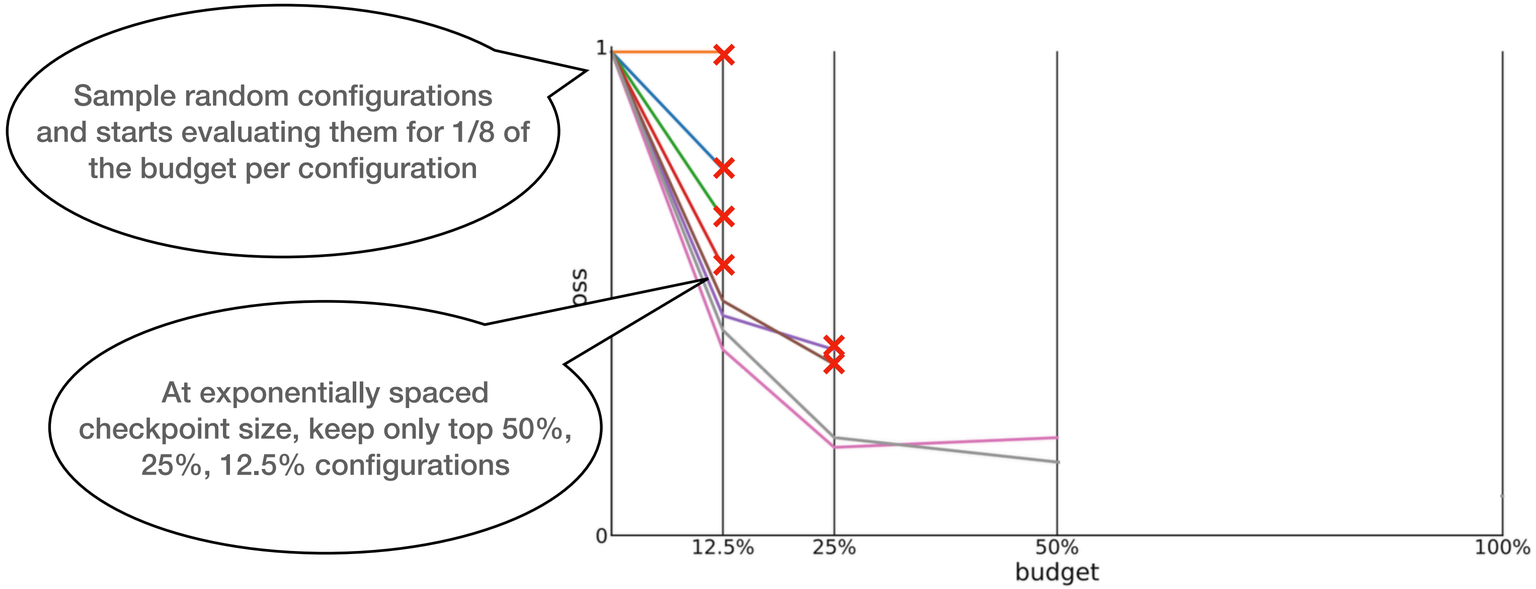


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

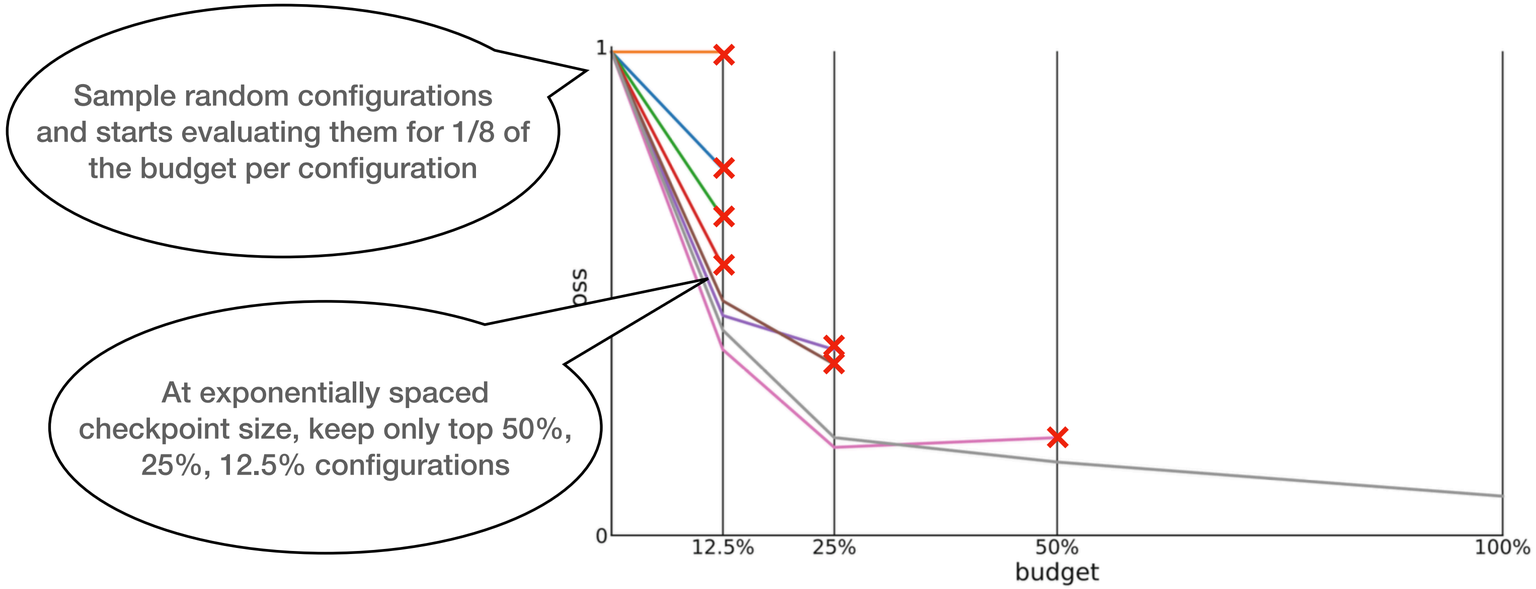


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

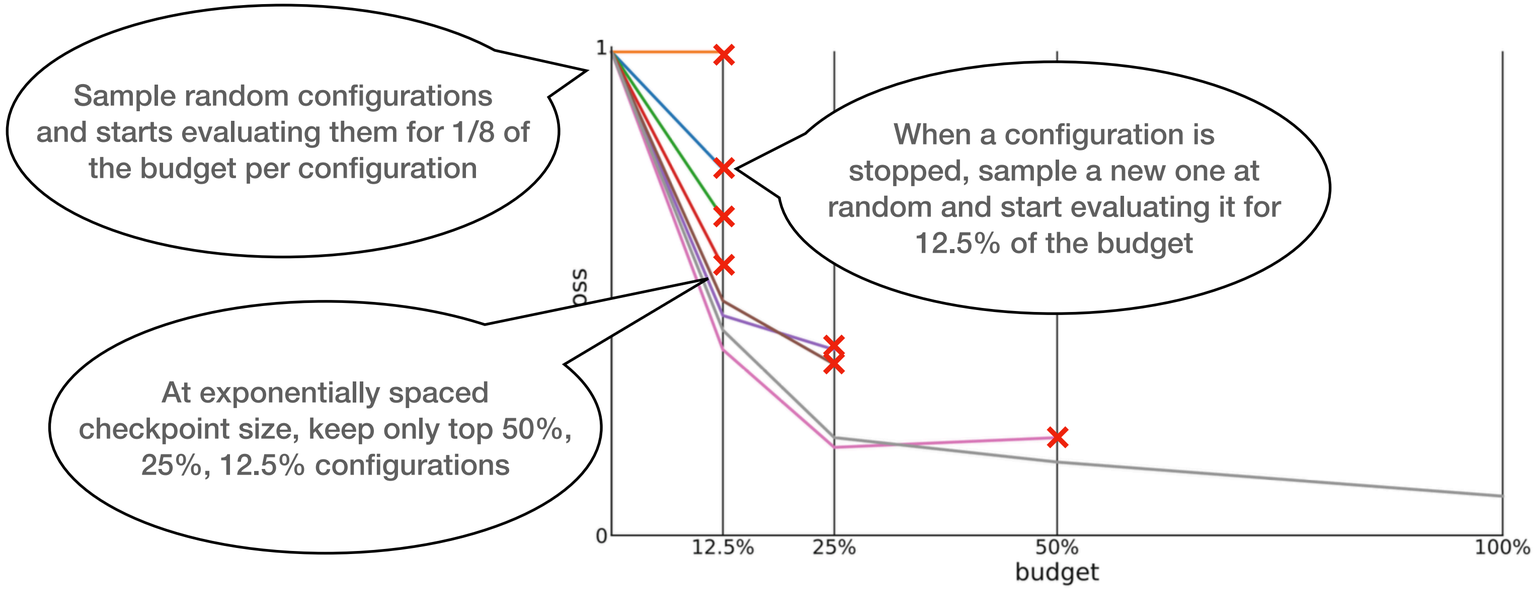


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

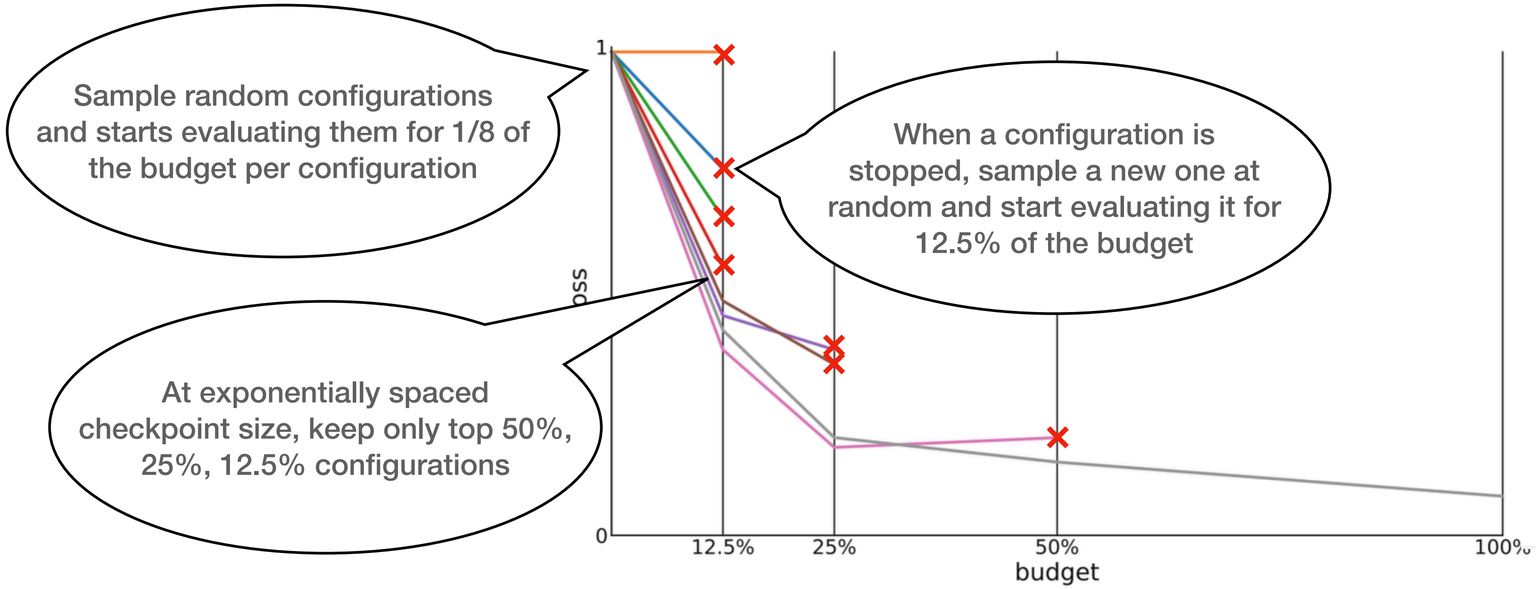
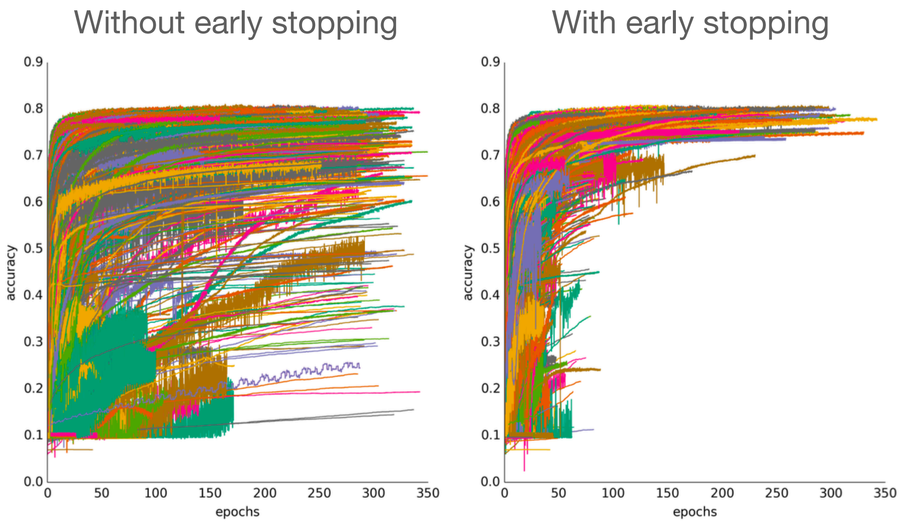


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

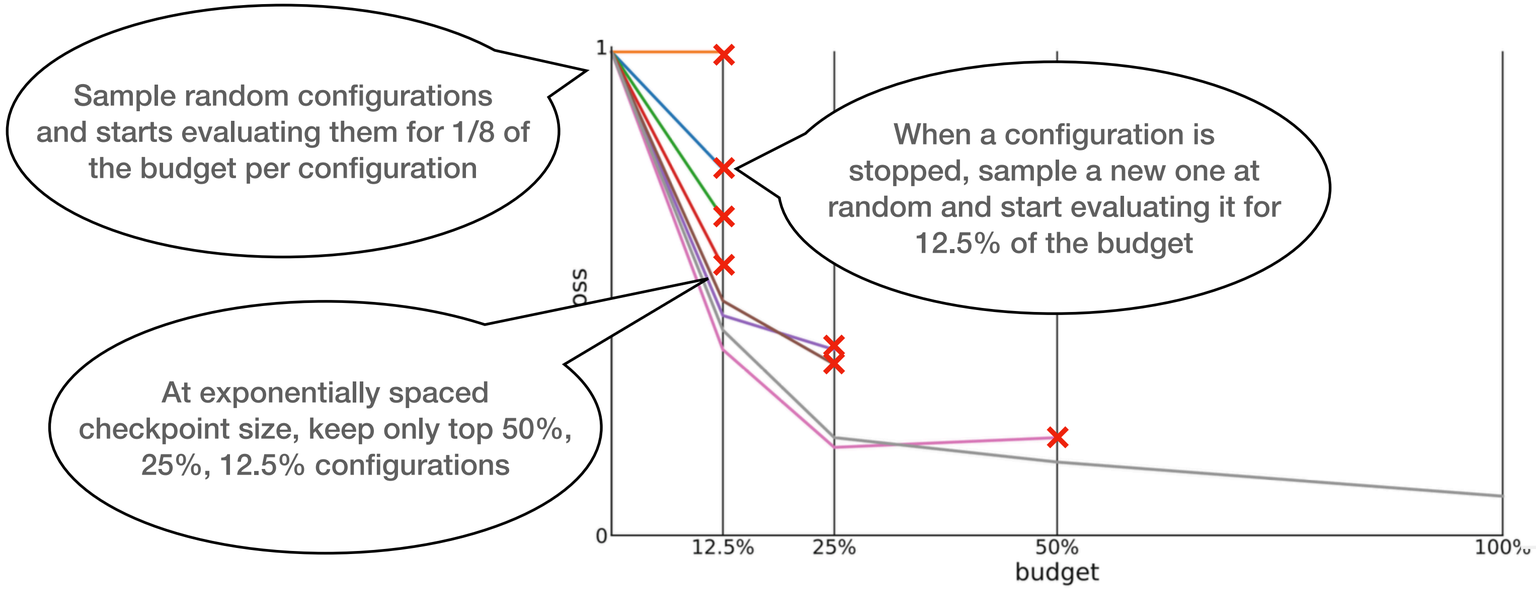
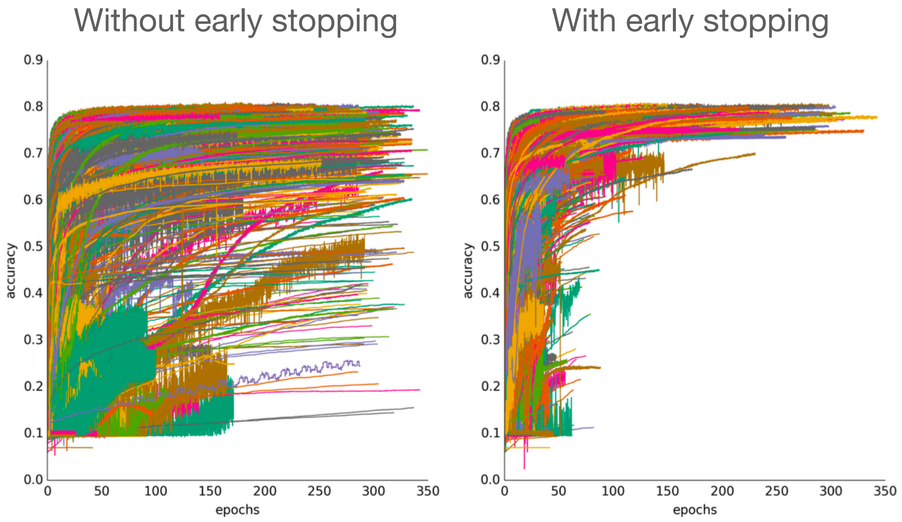


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

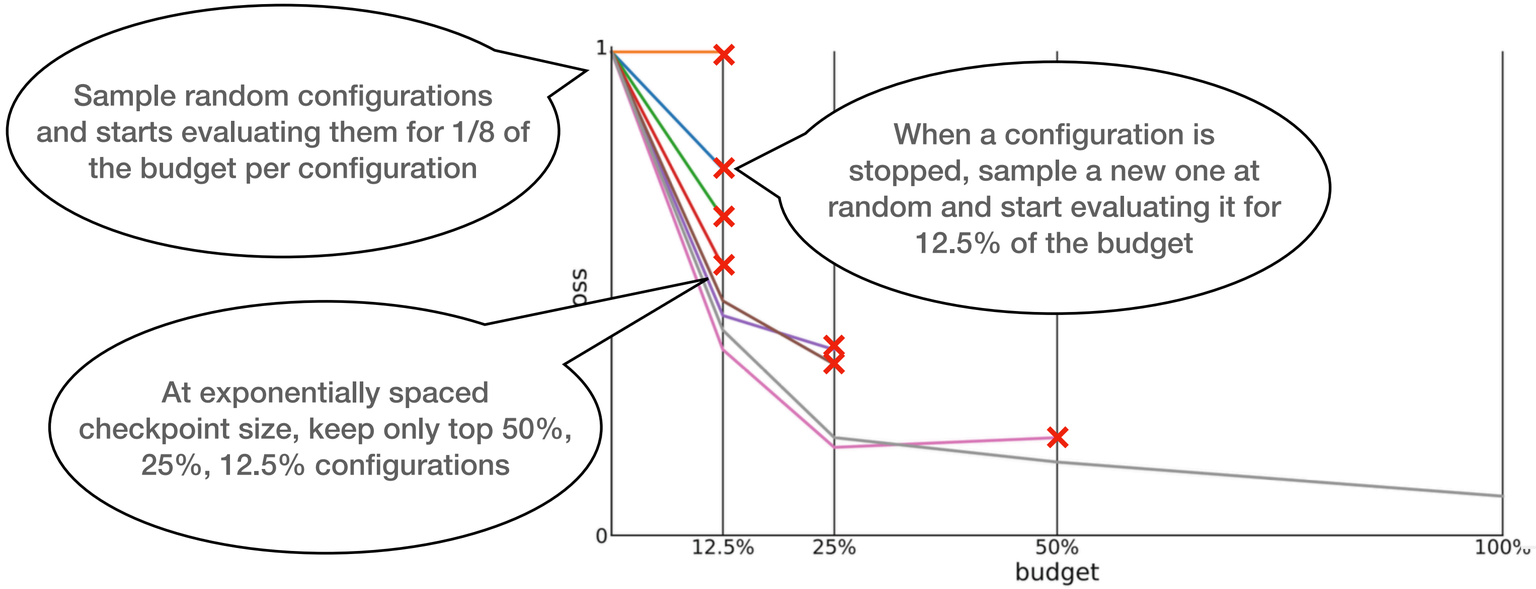
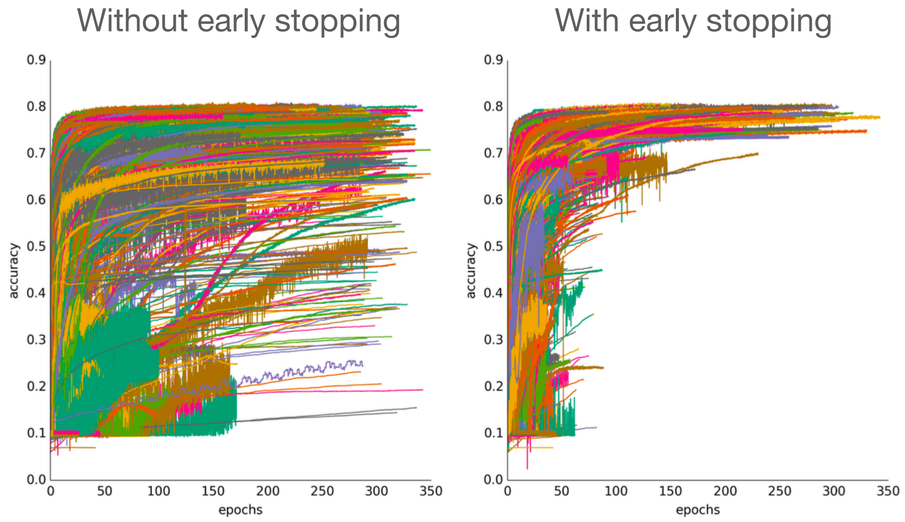


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

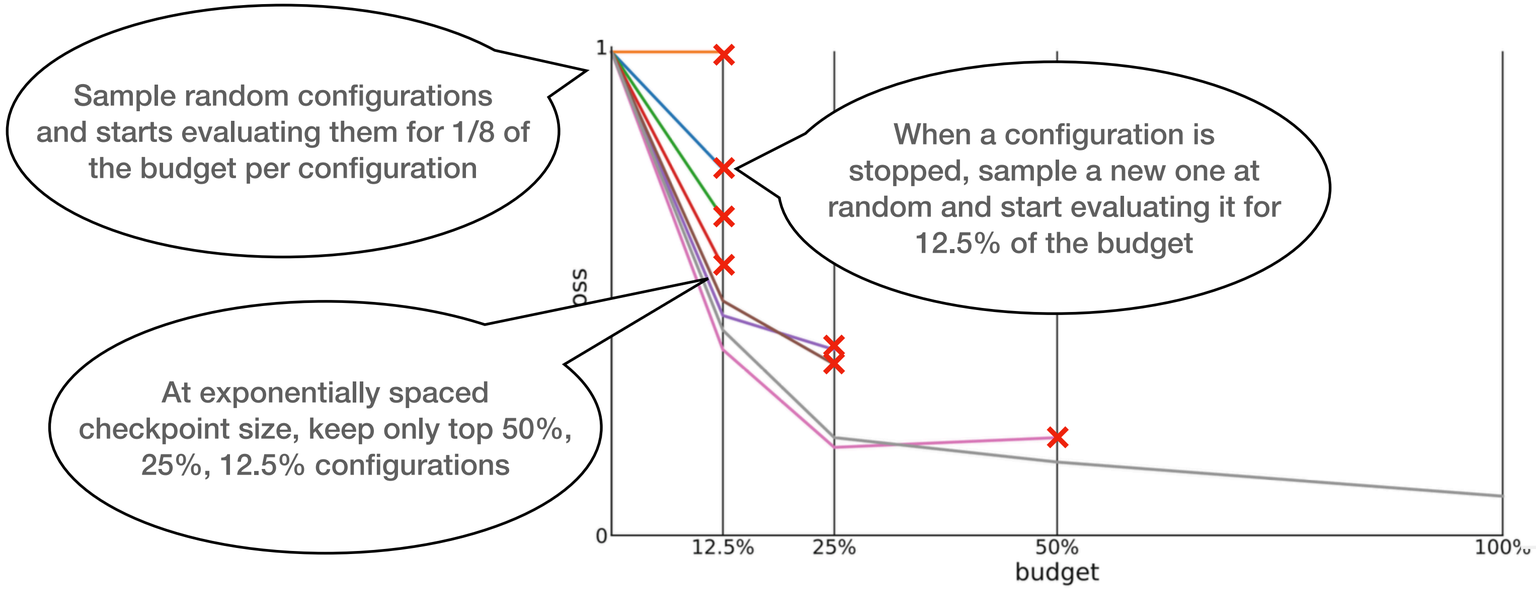
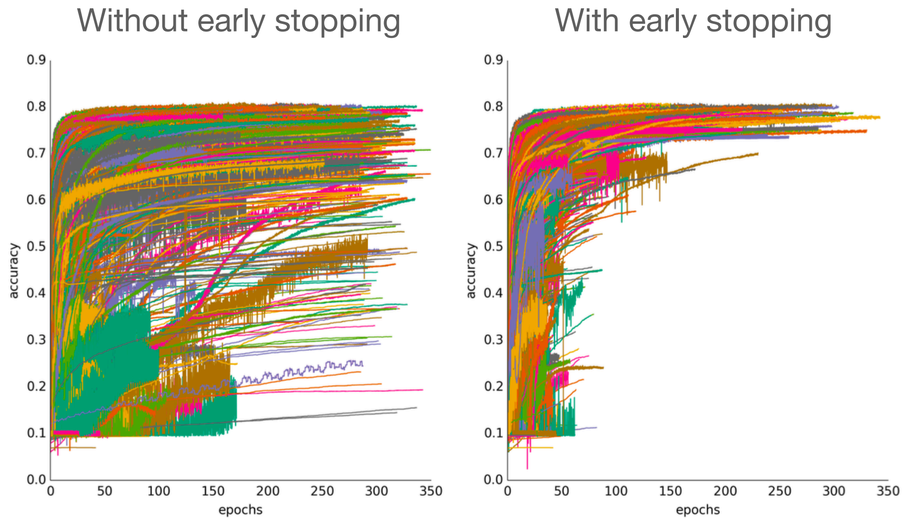


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?
- 🤔 We need to sort to discard the bottom half of configurations, how can we sort if we have multiple objectives?

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

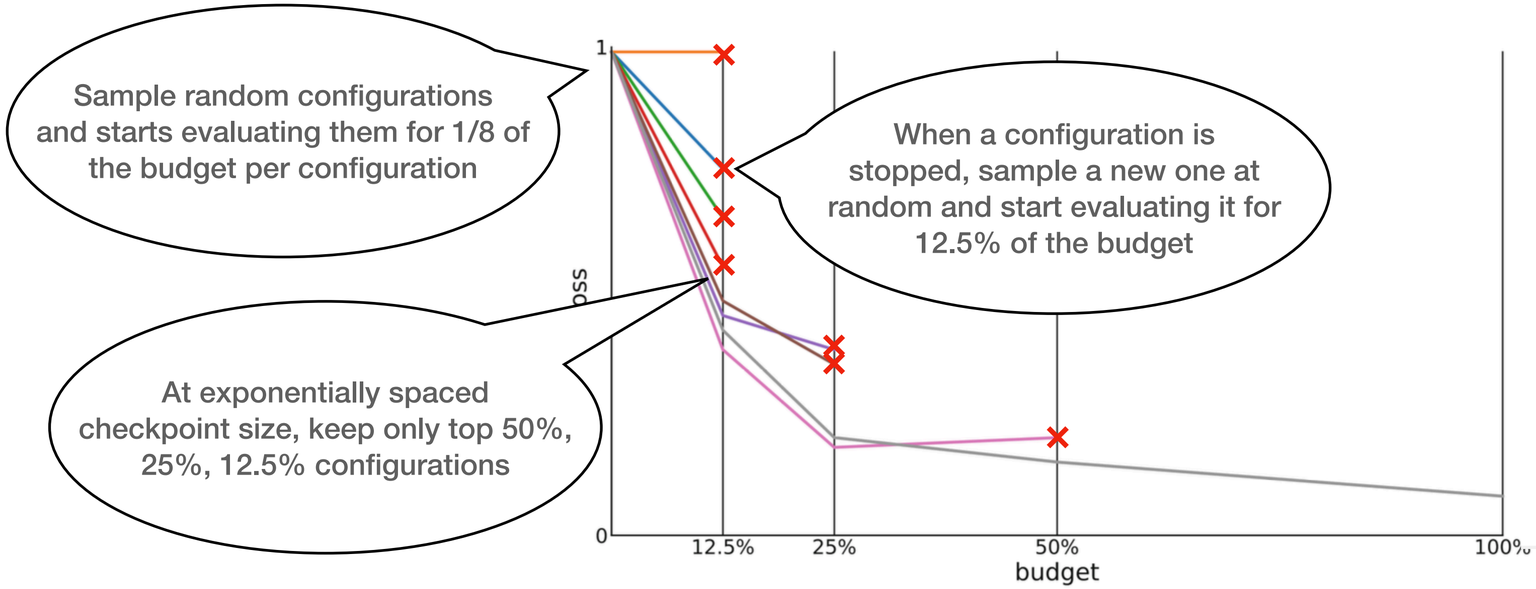
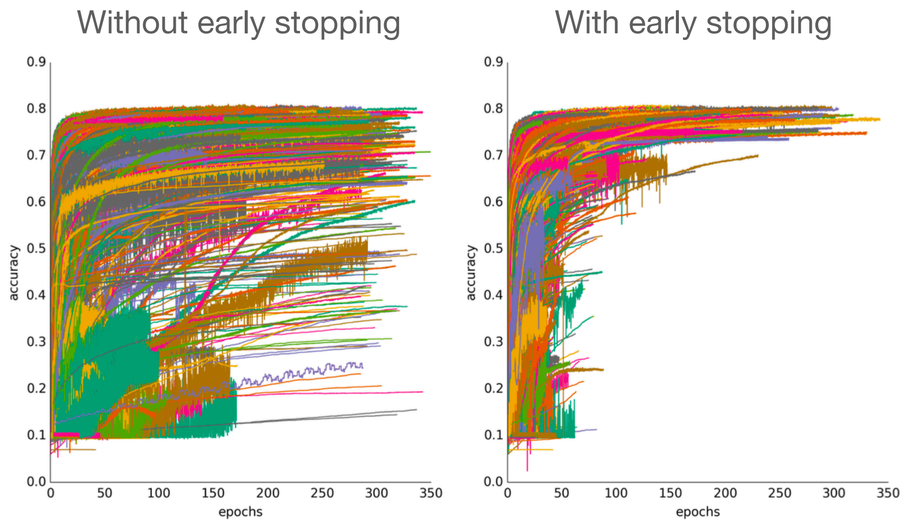


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?
- 🤔 We need to sort to discard the bottom half of configurations, how can we sort if we have multiple objectives?

💡 Non-dominated sort allows to sort even when we have multiple objectives

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \setminus \mathscr{P}(y)$, break ties with an heuristic

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic

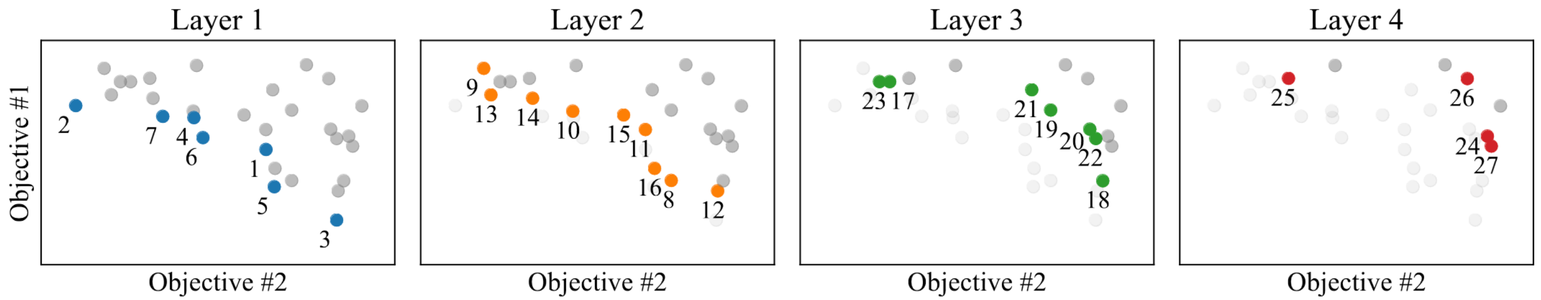


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic

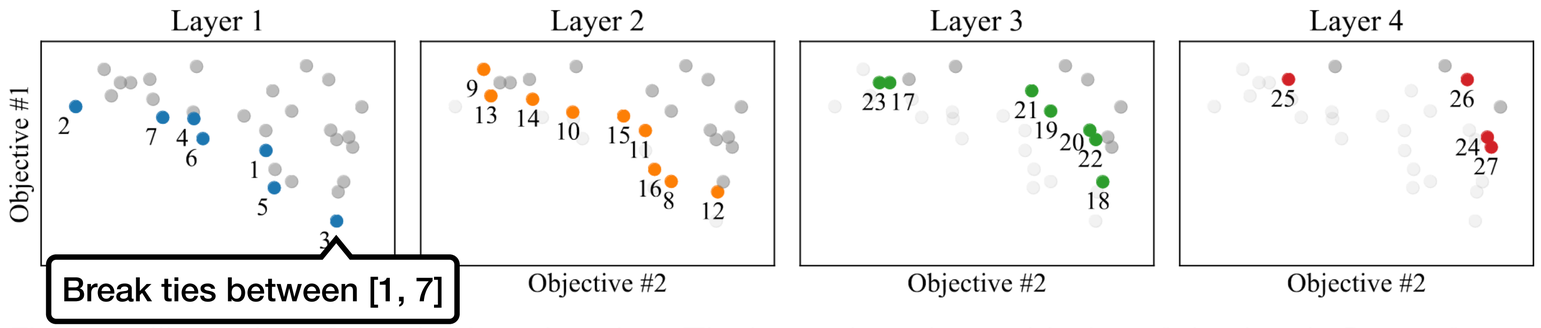


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:

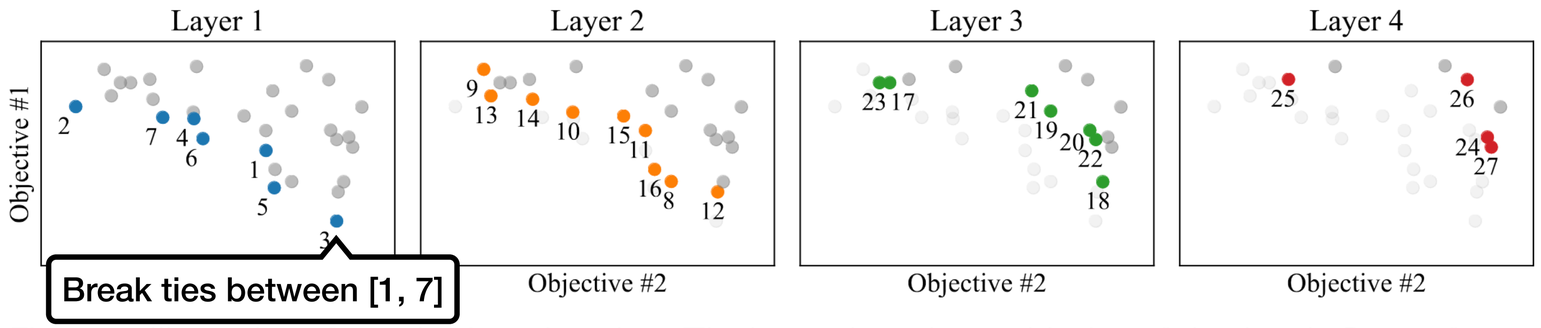


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:

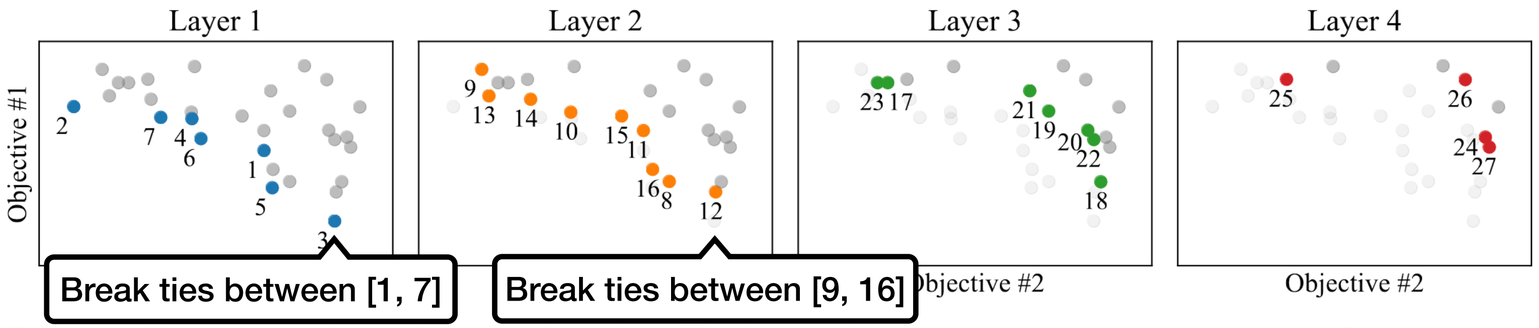


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance

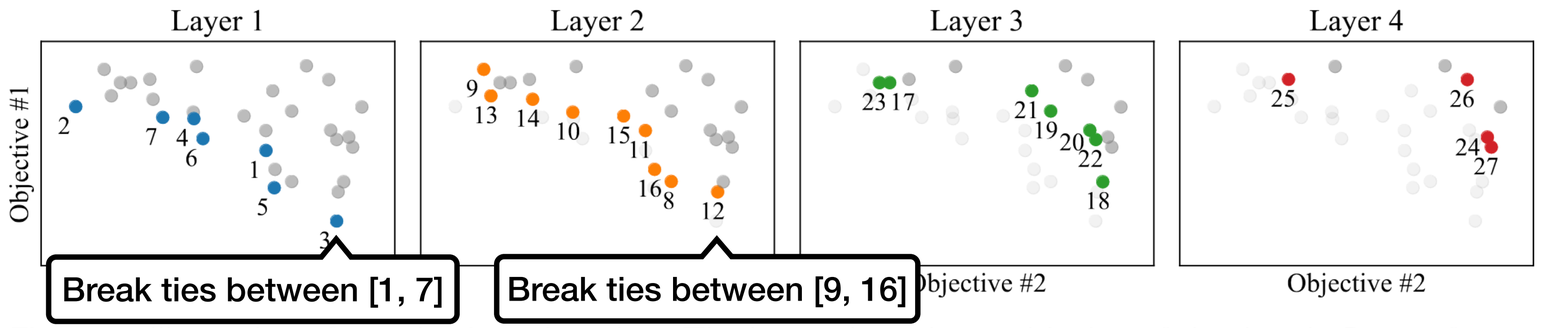


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance
    - Epsilon-net

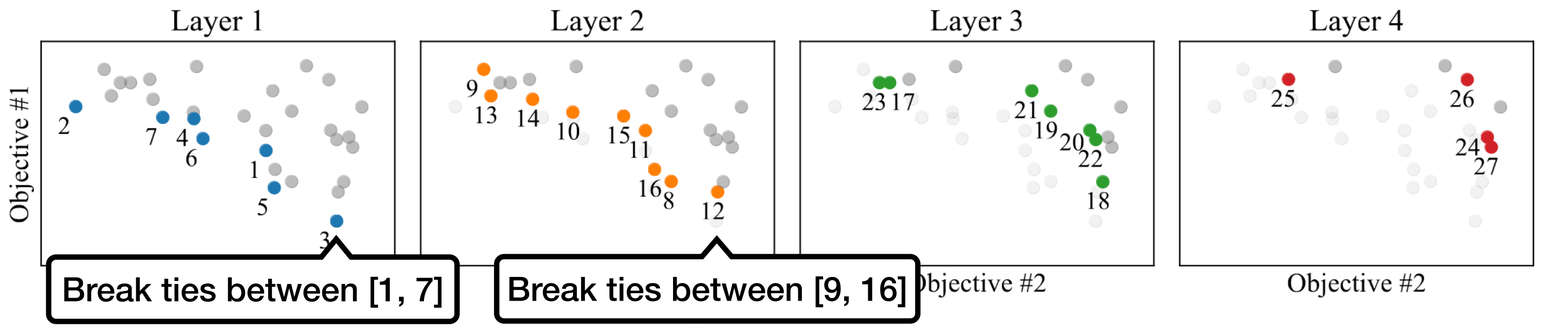


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Multiobjective optimization

Non dominated sort: extending sorting to multiple objectives

- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance
    - Epsilon-net
    - …

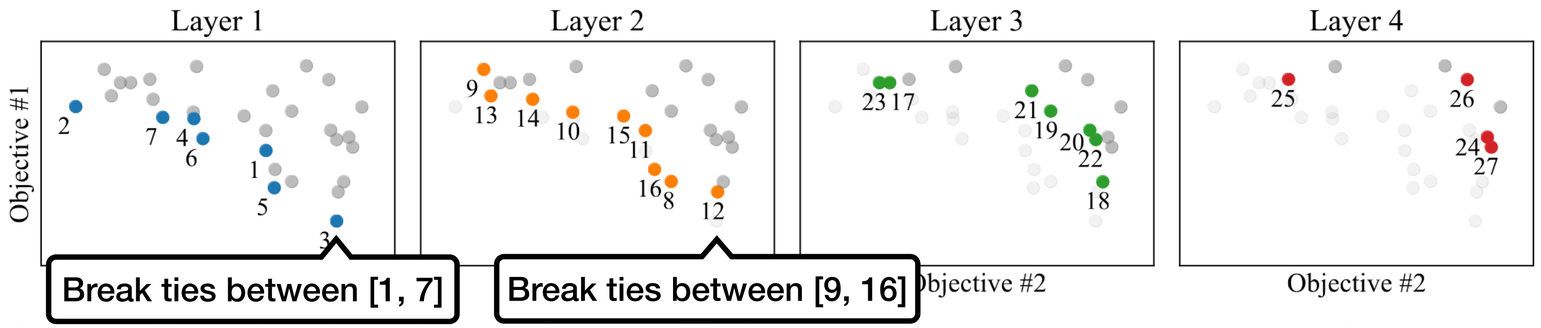
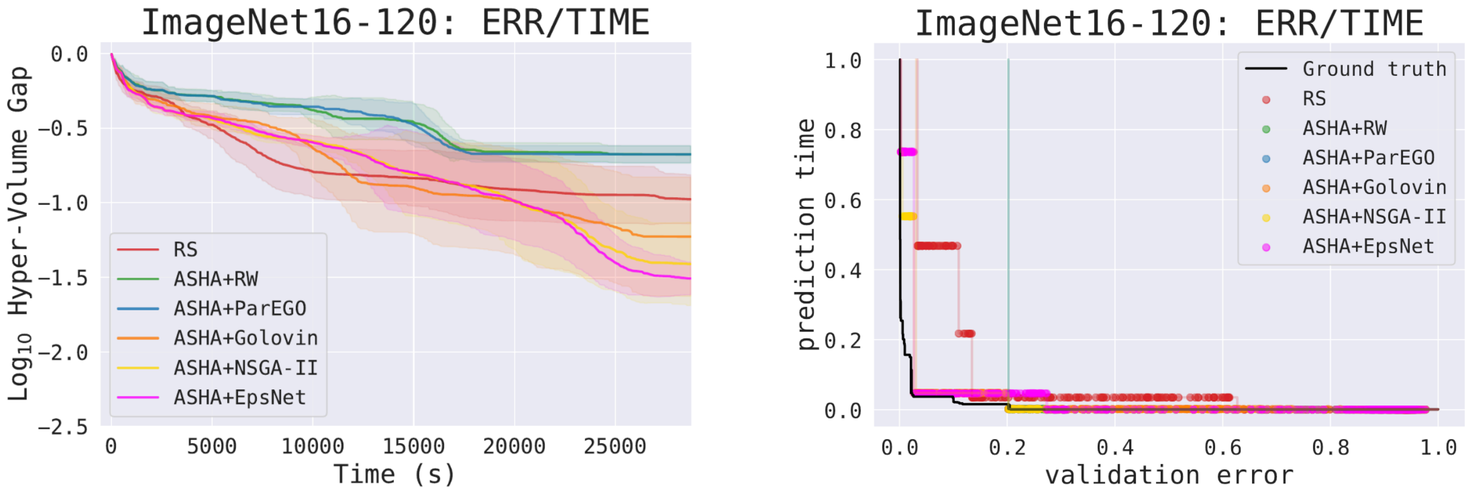


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

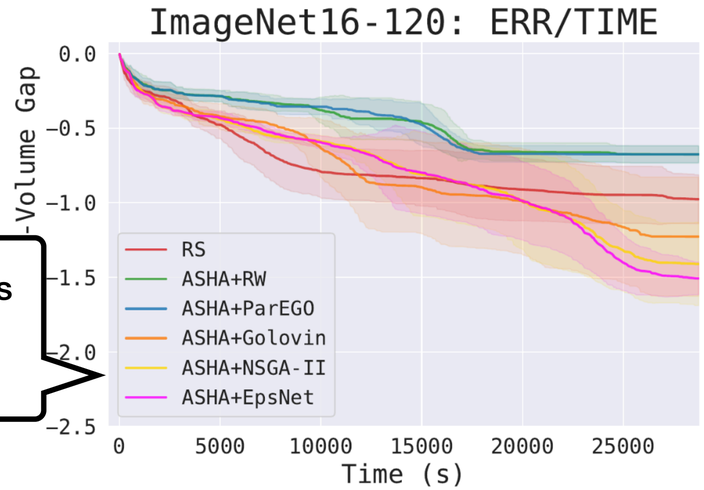
Extending Multifidelity to multi-objective

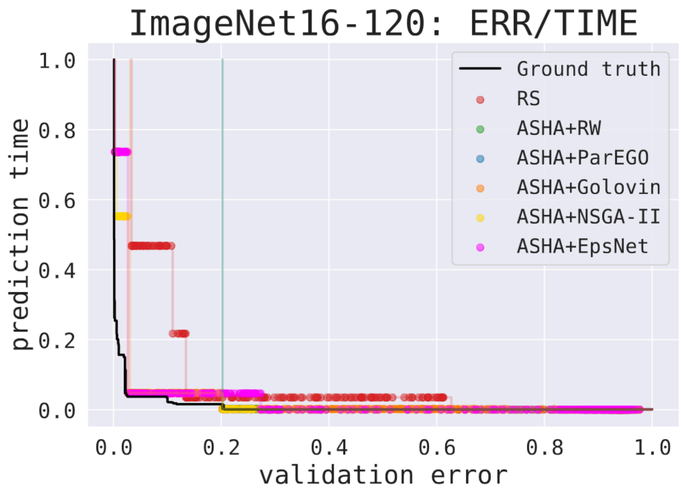


Multi-objective Asynchronous Successive Halving [Schmucker 2021]

Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!

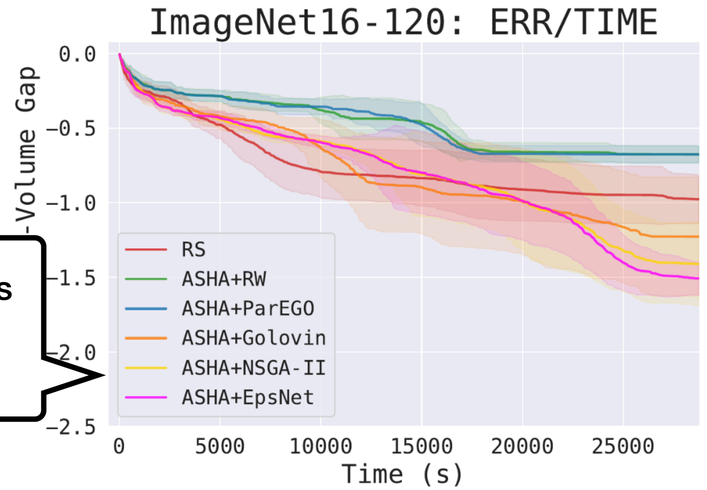


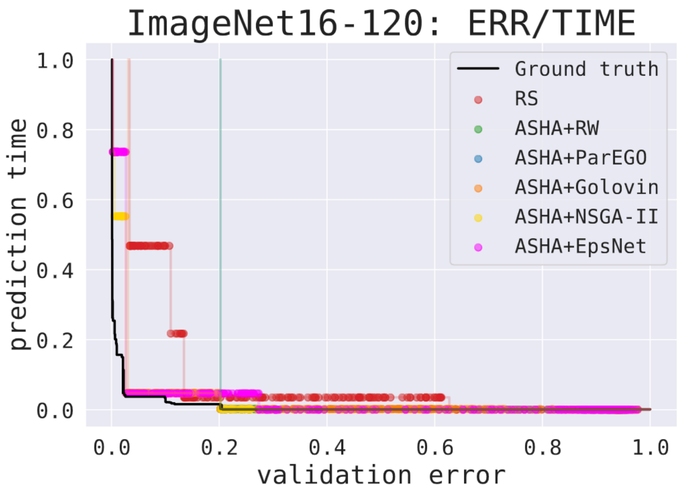


Multi-objective Asynchronous Successive Halving [Schmucker 2021]

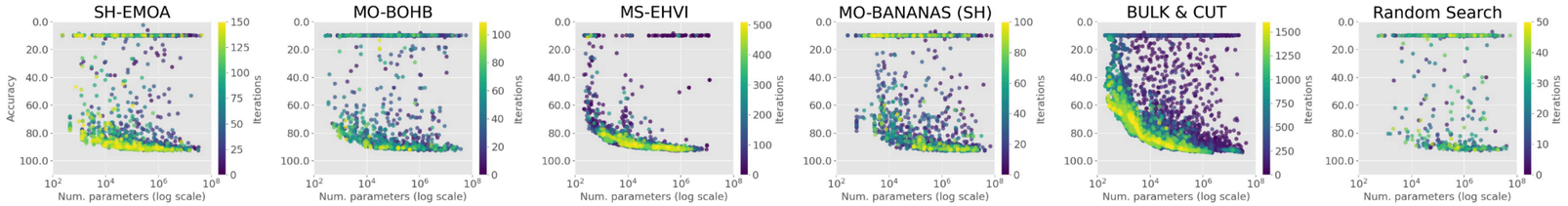
Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!





Multi-objective Asynchronous Successive Halving [Schmucker 2021]

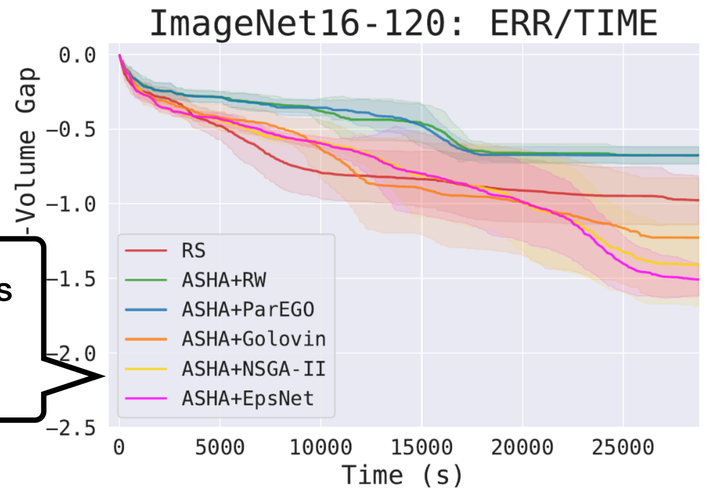


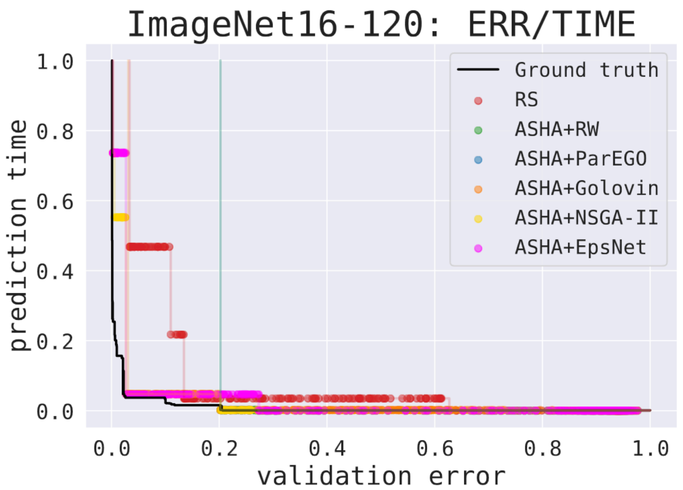
(b) Sampled configurations on Fashion-MNIST dataset.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

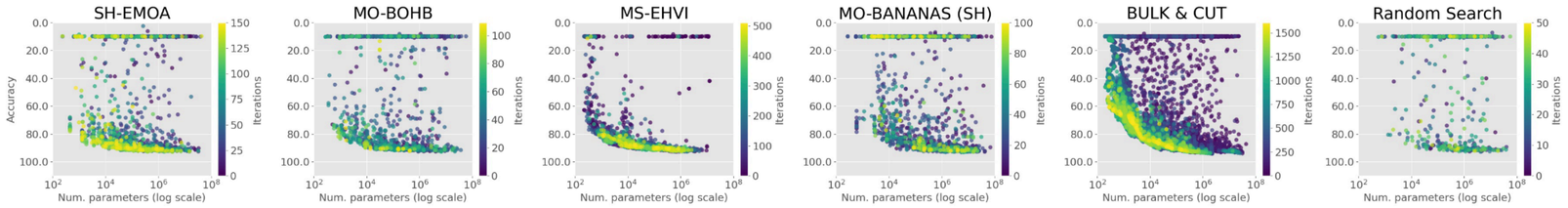
Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!

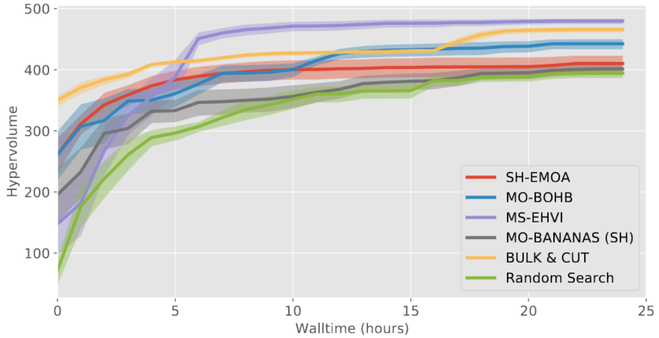




Multi-objective Asynchronous Successive Halving [Schmucker 2021]



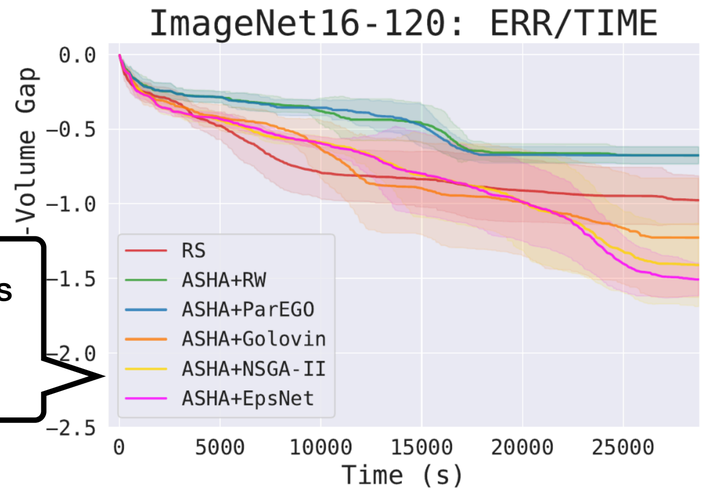
(b) Sampled configurations on Fashion-MNIST dataset.



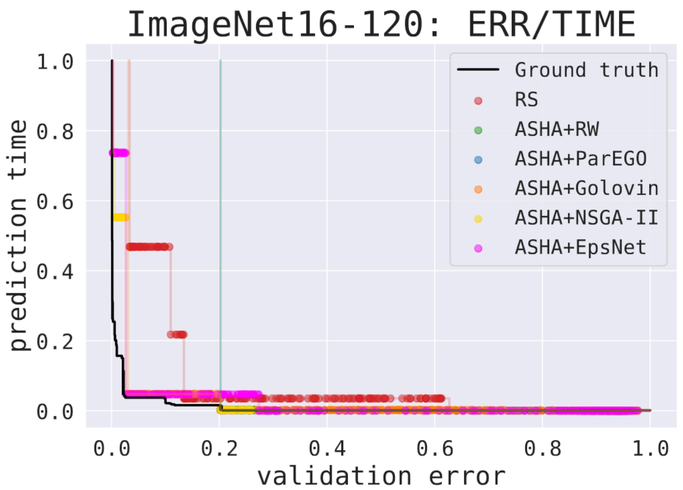
(d) Fashion-MNIST dataset on test split.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

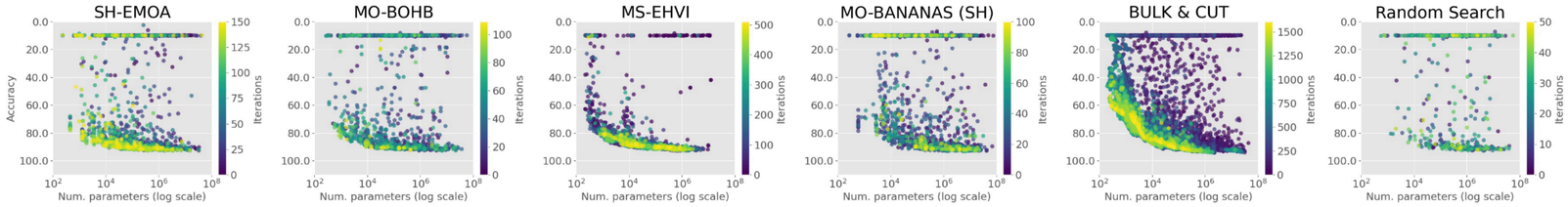
Extending Multifidelity to multi-objective



Non-dominated sort, works much better than scalarization!

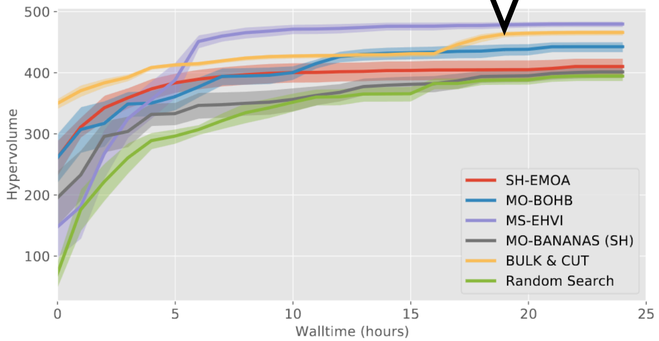


Multi-objective Asynchronous Successive Halving [Schmucker 2021]



(b) Sampled configurations on Fashion-MNIST dataset.

Non-dominated sort, works well in general…



(d) Fashion-MNIST dataset on test split.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

A case study: tuning LLM judges

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: “What is some cool music from the 1920s?”

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM



LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM → "Model 1"

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM → "Model 1"

- Pros: cheaper than human annotations & can evaluate open-ended text

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM → "Model 1"

- Pros: cheaper than human annotations & can evaluate open-ended text
- Cons: can be biased, very often rely on GPT-4 or closed models

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM → "Model 1"

- Pros: cheaper than human annotations & can evaluate open-ended text
- Cons: can be biased, very often rely on GPT-4 or closed models
- Can be used for model selection, leaderboard…

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM →

Rank	Model Name	LC Win Rate	Win Rate
1	GPT-4 Omni (05/13)	57.5%	51.3%
2	GPT-4 Turbo (04/09)	55.0%	46.1%
3	Yi-Large Preview	51.9%	57.5%
4	GPT-4 Preview (11/06)	50.0%	50.0%
5	Claude 3 Opus (02/29)	40.5%	29.1%
6	GPT-4	38.1%	23.6%
7	Qwen1.5 72B Chat	36.6%	26.5%
8	GPT-4 (03/14)	35.3%	22.1%
9	Claude 3 Sonnet (02/29)	34.9%	25.6%
10	Llama 3 70B Instruct	34.4%	33.2%

Alpaca Eval Leaderboard: winrate against GPT4-turbo

- Pros: cheaper than human annotations & can evaluate open-ended text
- Cons: can be biased, very often rely on GPT-4 or closed models
- Can be used for model selection, leaderboard…

LLM as a judge

Quick recap

- Idea 💡: ask LLM to tell which LLM output is better

Please say which model is better when answering the question: "What is some cool music from the 1920s?"

Model 1: The 1920s was a fantastic decade for music, marked by the rise of jazz, blues …

Model 2: The 1920s, often referred to as the "Roaring Twenties," was a period…

LLM →

Rank	Model Name	LC Win Rate	Win Rate
1	GPT-4 Omni (05/13)	57.5%	51.3%
2	GPT-4 Turbo (04/09)	55.0%	46.1%
3	Yi-Large Preview	51.9%	57.5%
4	GPT-4 Preview (11/06)	50.0%	50.0%
5	Claude 3 Opus (02/29)	40.5%	29.1%
6	GPT-4	38.1%	23.6%
7	Qwen1.5 72B Chat	36.6%	26.5%
8	GPT-4 (03/14)	35.3%	22.1%
9	Claude 3 Sonnet (02/29)	34.9%	25.6%
10	Llama 3 70B Instruct	34.4%	33.2%

Alpaca Eval Leaderboard: winrate against GPT4-turbo

- Pros: cheaper than human annotations & can evaluate open-ended text
- Cons: can be biased, very often rely on GPT-4 or closed models
- Can be used for model selection, leaderboard…
- **Our Goal**: Provide high-accuracy, low-cost LLM judges with **open** models

Tuning LLM judges

Search space

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures
- Boolean to average predictions with both orders

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

Tuning LLM judges

Search space

- Prompt:
  - Output format: 5 options
  - 3 booleans to ask LLM to provide
    - answer: its own answer to the instruction as proposed
    - example: an example of a judgement
    - explanation: its explanation on the given preference
  - Boolean to use json formatting
- Inference hyperparameters:
  - 7 open-weight options (Llama3, Qwen2.5, Gemma 2 at different size)
  - 4 temperatures
- Boolean to average predictions with both orders

Prompt Template

You are a highly efficient assistant, please evaluate and select the best large language model based on the quality of their responses to a given instruction.

User Prompt: Who is Geoffrey Hinton?
Assistant A: Geoffrey Hinton is a research scientist.
Assistant B: I do not know who Geoffrey Hinton is.

 # Your Output
Format Description
Your output should follow this format:

```
{
 "answer":  <your answer to the user prompt>,
 "explanation":  <your explanation on why you think A or B is better>,
 "score_A":  <between 0 and 10 to rate the quality of A>,
 "score_B":  <between 0 and 10 to rate the quality of B>
}
```

Your output, do not repeat the input above.

Figure 3. Illustration of the prompt templating approach. We parametrize the prompt with the following hyperparameters: Provide answer, Provide explanation, Provide example, use JSON, output preference format. Given each of the $2^4 \times 5 = 80$ prompt hyperparameters, we generate a prompt like this one.

In total we have $5 \times 2^4 = 80$ prompts and $7 \times 4 \times 80 \times 2 = 4480$ possible judges

😱 Evaluating with Spearman correlation in brute force would cost $2M (!)

LLM as a judge

Hyperparameter optimization

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…
  - We could tune the judge with multi-objective optimization!

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…
  - We could tune the judge with multi-objective optimization!
  - … but evaluating a single judge configuration costs ~370$ 🥶 (need to evaluate many models to compute Spearman correlation)

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…
  - We could tune the judge with multi-objective optimization!
  - … but evaluating a single judge configuration costs ~370$ 🥶 (need to evaluate many models to compute Spearman correlation)
- Our approach:

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…
  - We could tune the judge with multi-objective optimization!
  - … but evaluating a single judge configuration costs ~370$ 🥶 (need to evaluate many models to compute Spearman correlation)
- Our approach:
  - 1. Identify a cheaper and better metric than Spearman corr.

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

LLM as a judge

Hyperparameter optimization

- Multiobjective: accuracy & cost per evaluation
- Currently configurations are manually selected and evaluated…
  - We could tune the judge with multi-objective optimization!
  - … but evaluating a single judge configuration costs ~370$ 🥶 (need to evaluate many models to compute Spearman correlation)
- Our approach:
  - 1. Identify a cheaper and better metric than Spearman corr.
  - 2. Use a multifidelity-multiobjective method to tune configurations

Performance of some judge hyperparameters for Alpaca Eval

Chain of thought improves **correlation** but increases **cost**

	Human agreement	Price [$/1000 examples]	Time [seconds/1000 examples]	Spearman corr.	Pearson corr.
alpaca_eval_gpt4	69.2	13.6	1455	0.97	0.93
alpaca_eval_cot_gpt4_turbo_fn	68.6	6.3	1989	0.97	0.90
alpaca_eval_gpt4_turbo_fn	68.1	5.5	864	0.93	0.82
alpaca_eval_llama3_70b_fn	67.5	0.4	209	0.90	0.86

Llama3-70B instead of GPT4-turbo worsen **correlation** but improves **cost**

How to compare judges?

Identifying a better metric

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human

How to compare judges?

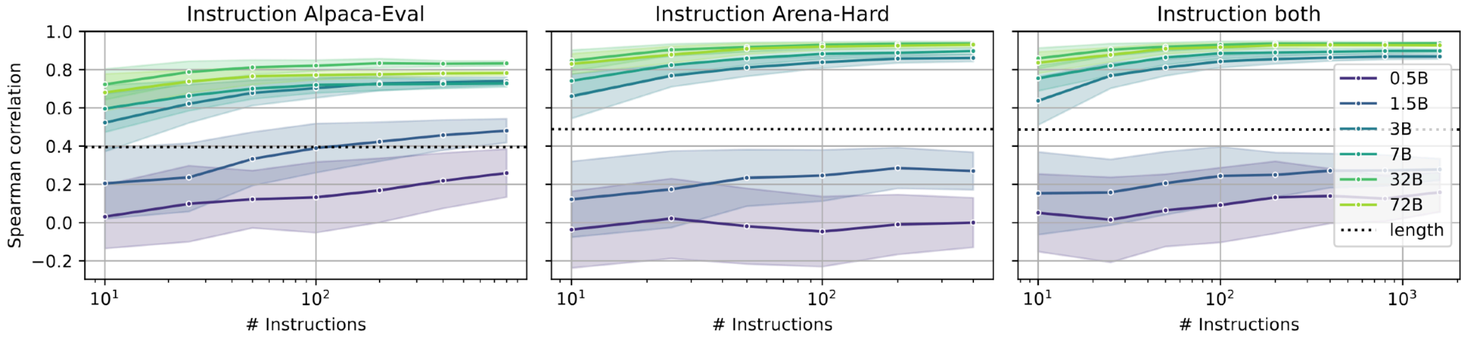
Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?

How to compare judges?

Identifying a better metric

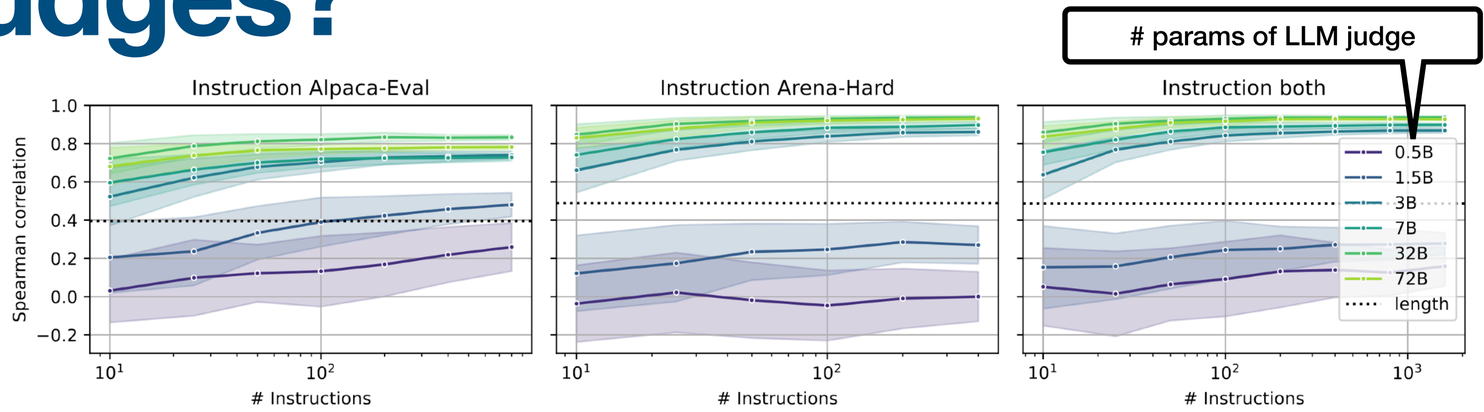
- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?



How to compare judges?

Identifying a better metric

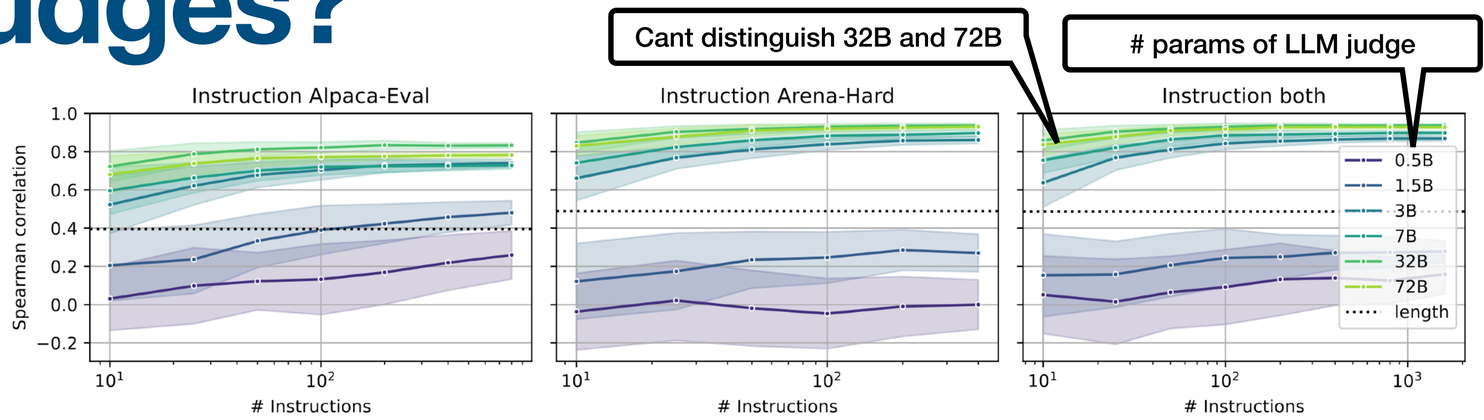
- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?



How to compare judges?

Identifying a better metric

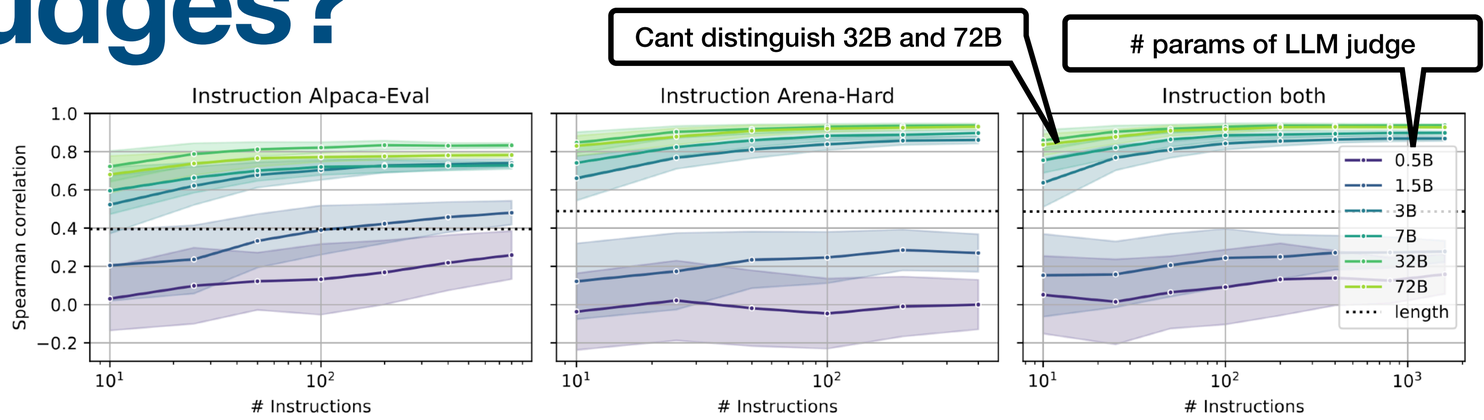
- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?

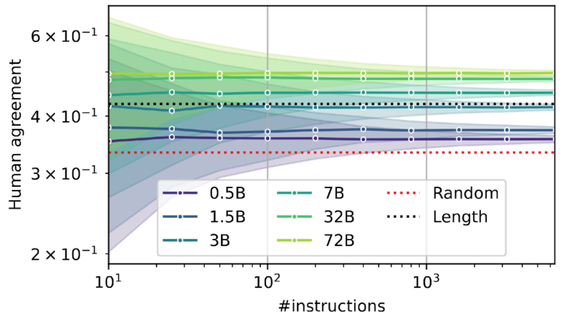


How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?

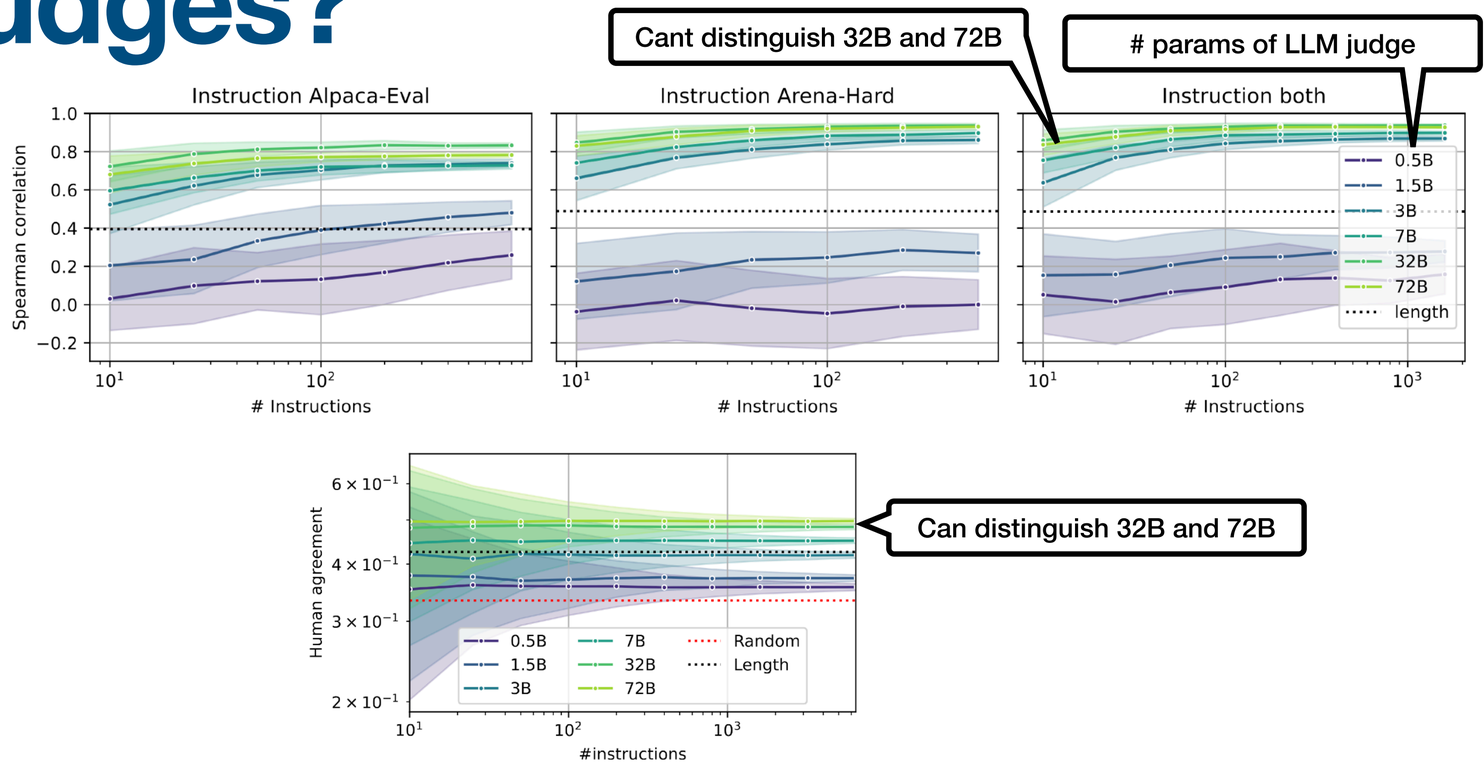




How to compare judges?

Identifying a better metric

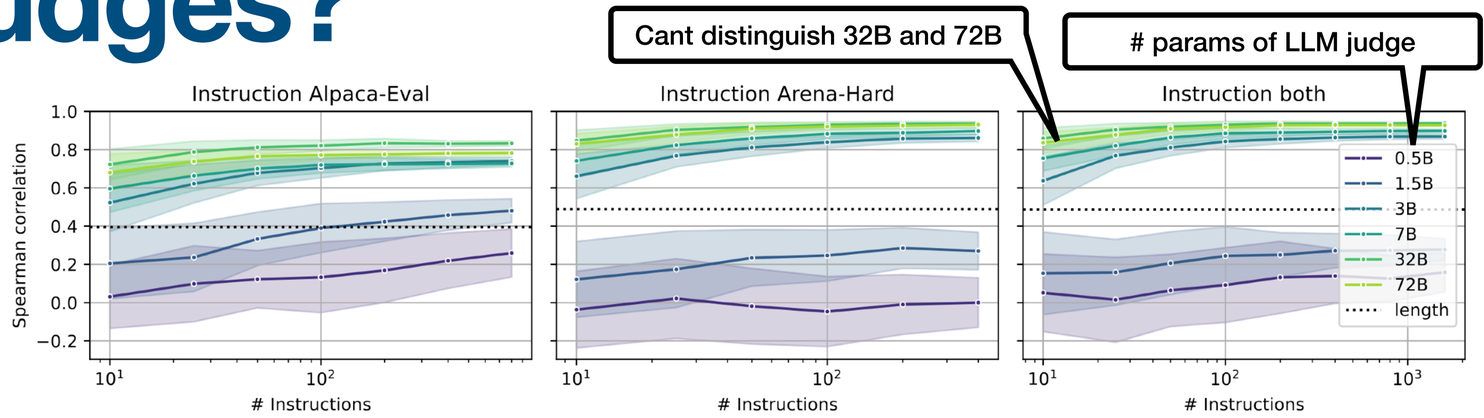
- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?

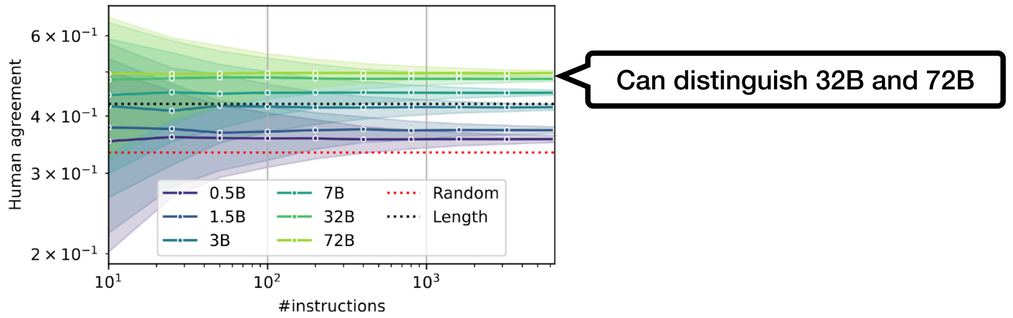


How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?





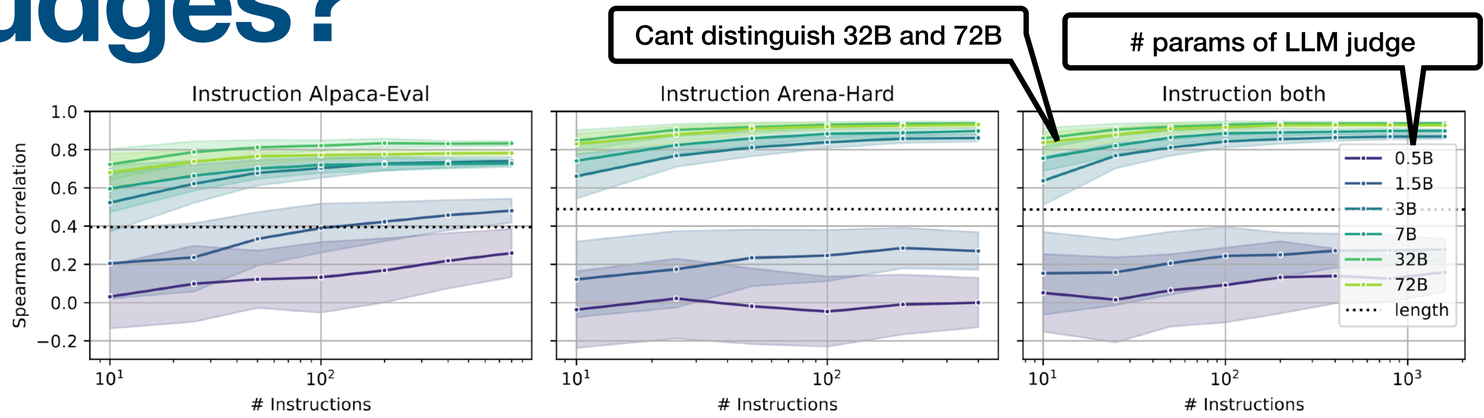
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

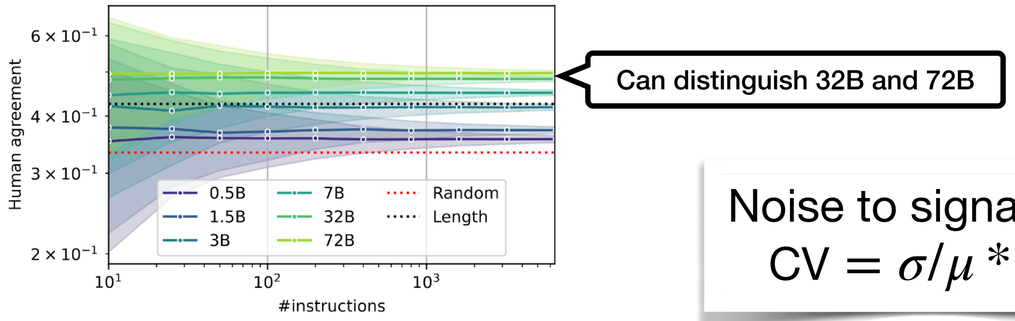
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?





Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

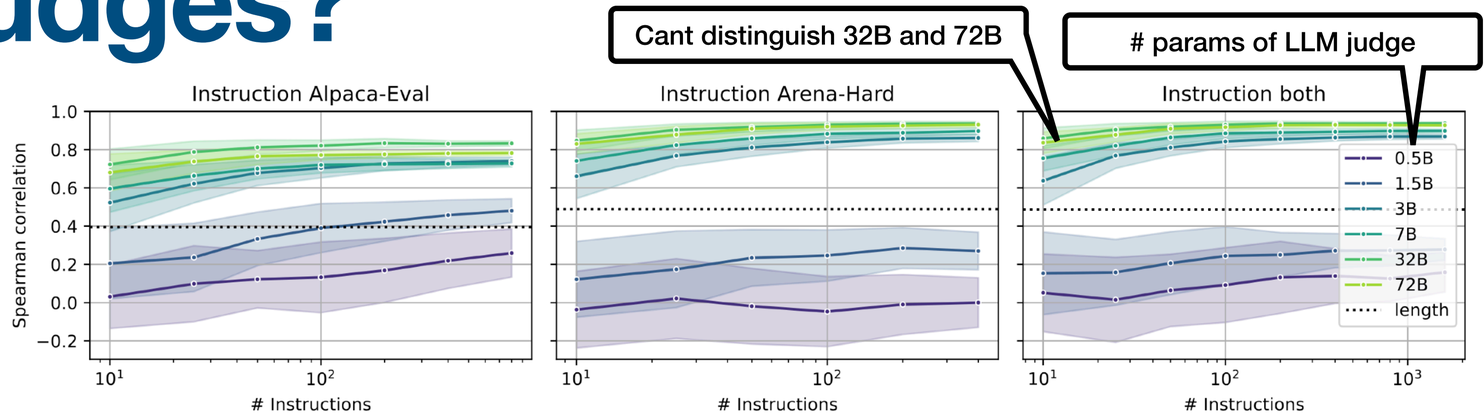
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

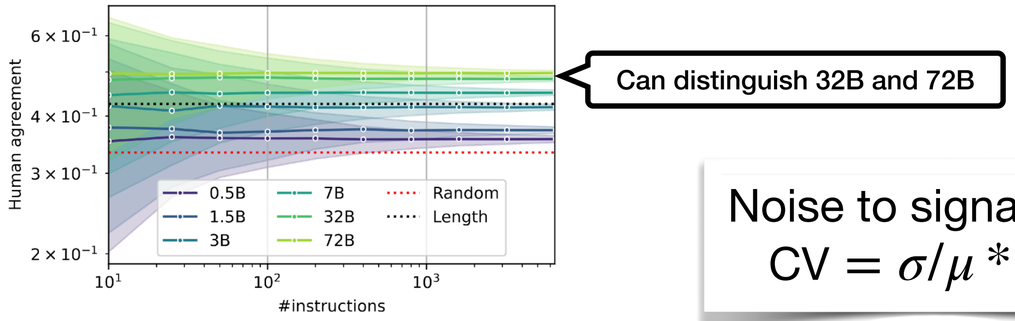
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?





Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

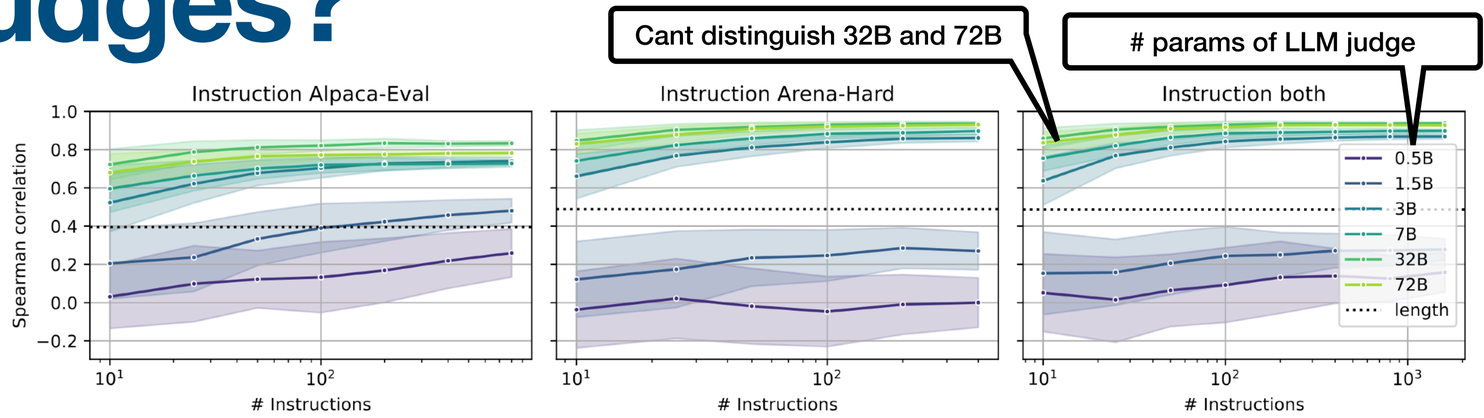
$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

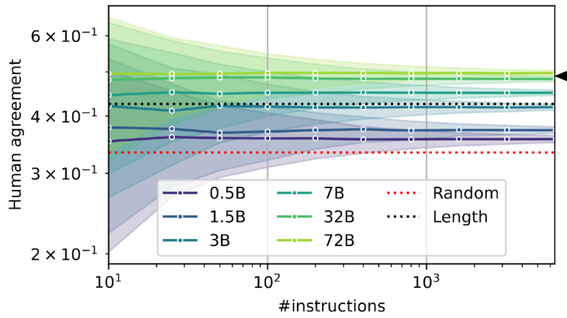
Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

How to compare judges?

Identifying a better metric

- Option 1: Spearman-correlation
  - 1. Evaluate a list of LLMs on a list of prompts
  - 2. Call LLM judge to compare with baseline (GPT-4) on all annotations
  - 3. Compute winrate
  - 4. Measure Spearman correlation with Chatbot Arena ranking
- Option 2: Human-agreement
  - Collect list of annotated preferences by human
  - Call LLM judge to annotate preference
  - Compare LLM judge preference with human
- How both options signal to noise ratio scale with the number of annotations?





Can distinguish 32B and 72B

Noise to signal ratio:
$\text{CV} = \sigma/\mu * 100$

$\#_{\text{params}}$	Sp. corr. (↑)	CV(↓)	Hum. agr. (↑)	CV(↓)
0.5B	0.09 ± 0.189	207.60	0.36 ± 0.006	1.70
1.5B	0.33 ± 0.137	41.26	0.37 ± 0.006	1.61
3B	0.82 ± 0.066	8.11	0.42 ± 0.006	1.45
7B	0.83 ± 0.082	9.84	0.45 ± 0.006	1.33
32B	0.90 ± 0.052	5.75	0.48 ± 0.006	1.29
72B	0.88 ± 0.074	8.43	0.50 ± 0.006	1.27

Table 1. Comparison of the variability of Spearman-correlation and Human-agreement metrics when using 6500 random annotations and the same default prompt for all model sizes.

Human agreement has **much** better signal to noise ratio!

Tuning LLM judges

Multifidelity & multiobjective optimization

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions

Tuning LLM judges

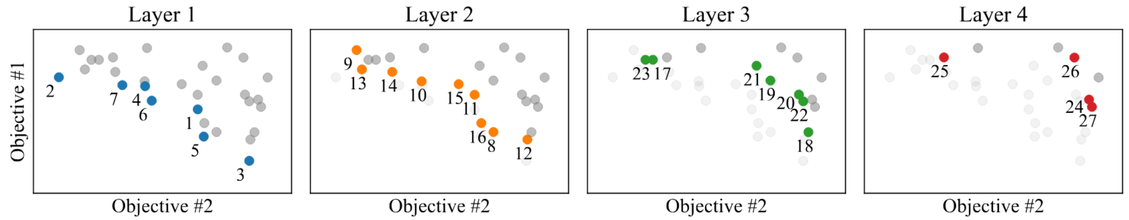
Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations

Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations



Tuning LLM judges

Multifidelity & multiobjective optimization

- Two objectives: human-agreement & cost per annotation
- Multi-fidelity multi-objective to the rescue!
  - Evaluate all 4480 configurations on 400 instructions
  - Pick top 1200 and evaluate on 1200 instructions
  - Pick again top 400 and evaluate on 3548 Instructions
- Use Non dominated sort to determine top configurations

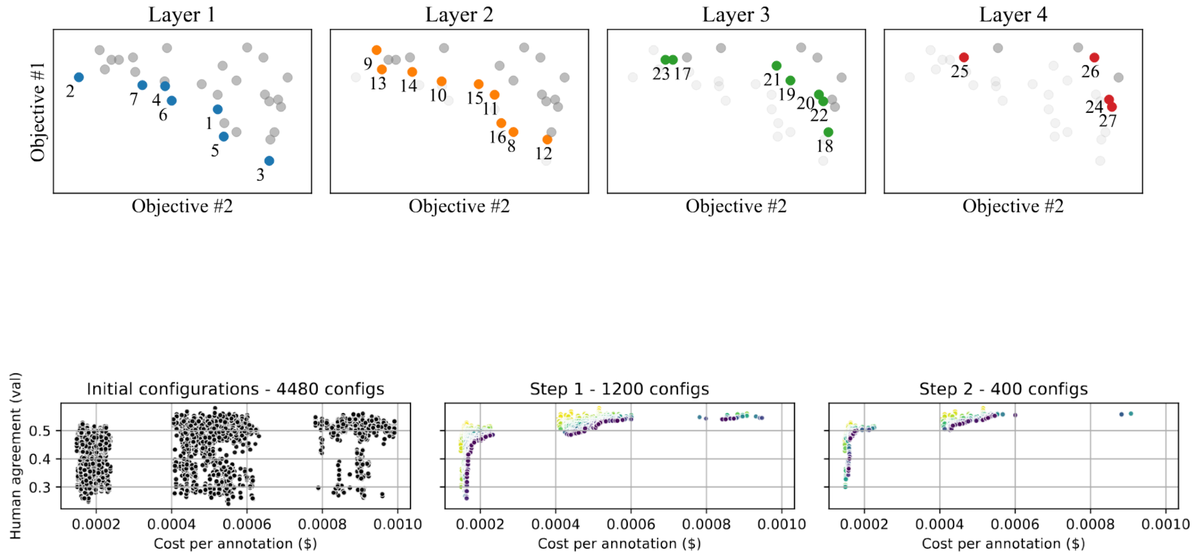


Figure 4. Illustration of the selection process. All 4 480 configurations are first evaluated on 400 instructions (left), the top 1 200 configurations are then evaluated on 1 200 instructions (center) and finally the top 400 configurations are evaluated on 3 548 instructions (right). The color denotes the ranking assigned by the non-dominated sort procedure.

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

LMsys test set

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

Judge	Human agr. (↑)
Random	0.33
Length	0.60
GPT-3.5	0.63
GPT-4	0.67
PandaLM-7B	0.60
PandaLM-70B	0.67
Ours-small	0.67
Ours-medium	0.78
Ours-large	0.76

Table 3. Comparison with PandaLM on PandaLM test set. Note that our method is not fine-tuned as opposed to PandaLM.

LMsys test set

PandaLM test set

Results of top judges

How are tuned judges from open-weights compared to previous approaches?

- How are tuned judges from open-weights compared to previous approaches?
- With tuning, we can outperform close-weight and fine-tune judges while having a much lower cost

Judge	Human agr. (↑)	Cost per 1K ann. (↓)
Random	0.33 +/- 0.01	-
Length	0.42 +/- 0.01	-
PandaLM-7B	0.38 +/- 0.01	6.0
JudgeLM-7B	0.42 +/- 0.01	8.6
Arena-Hard	0.50 +/- 0.01	1.2
Ours-small	0.45 +/- 0.01	0.21
Ours-medium	0.47 +/- 0.01	0.48
Ours-large	0.49 +/- 0.01	0.48

Table 2. Comparison of judges on LMSys test instructions. For each judge, we report the bootstrap mean and std for human-agreement on 3K test instructions with 100 seeds.

LMsys test set

Judge	Human agr. (↑)
Random	0.33
Length	0.60
GPT-3.5	0.63
GPT-4	0.67
PandaLM-7B	0.60
PandaLM-70B	0.67
Ours-small	0.67
Ours-medium	0.78
Ours-large	0.76

Table 3. Comparison with PandaLM on PandaLM test set. Note that our method is not fine-tuned as opposed to PandaLM.

PandaLM test set

Judge	Sp. corr. (↑)	Cost per 1K ann. (↓)
Length	0.50 +/- 0.21	-
Arena-hard + Claude	0.82 +/- 0.12	75.0
Arena-hard + GPT4	0.90 +/- 0.06	50.0
Ours-small	0.81 +/- 0.10	0.21
Ours-medium	0.93 +/- 0.05	0.48
Ours-large	0.86 +/- 0.09	0.48

Table 4. For each judge, we compute the Spearman correlation between win-rates using the protocol of Arena-Hard and ELO-ratings computed from human annotations from Chatbot Arena. We report the mean and std over 100 boostraps of the set of models.

ArenaHard

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

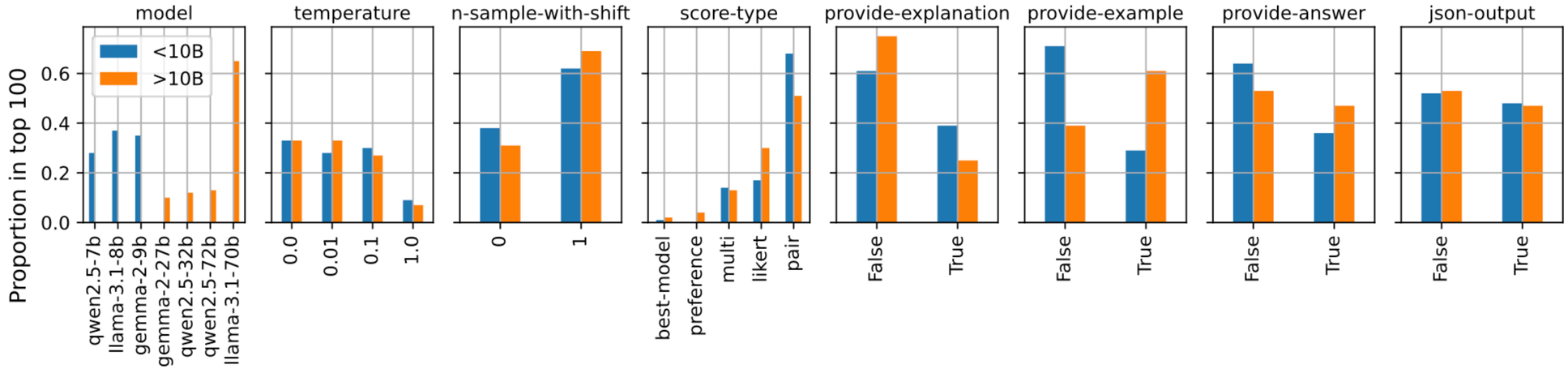


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

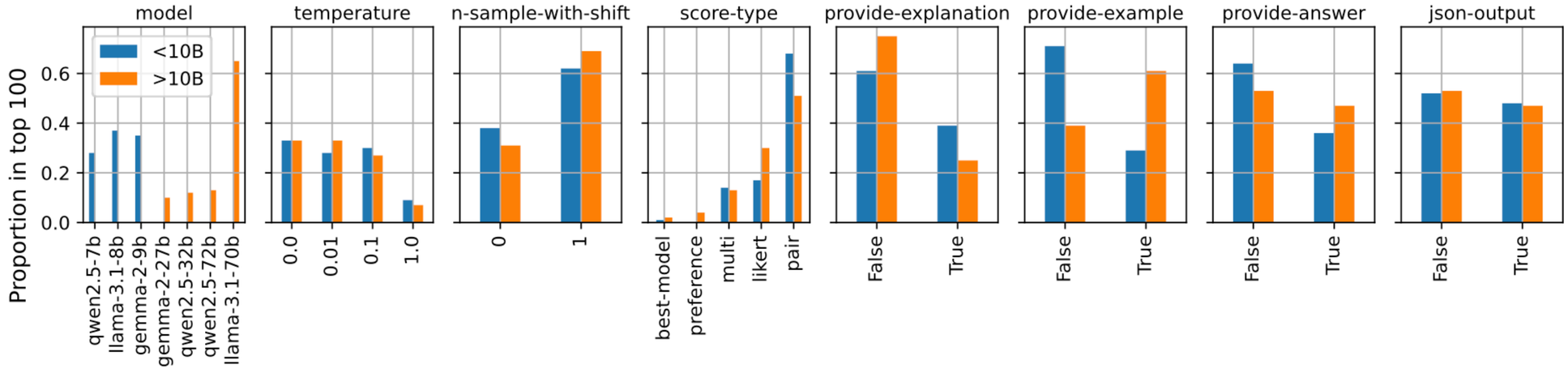


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

Format used for output is very important.

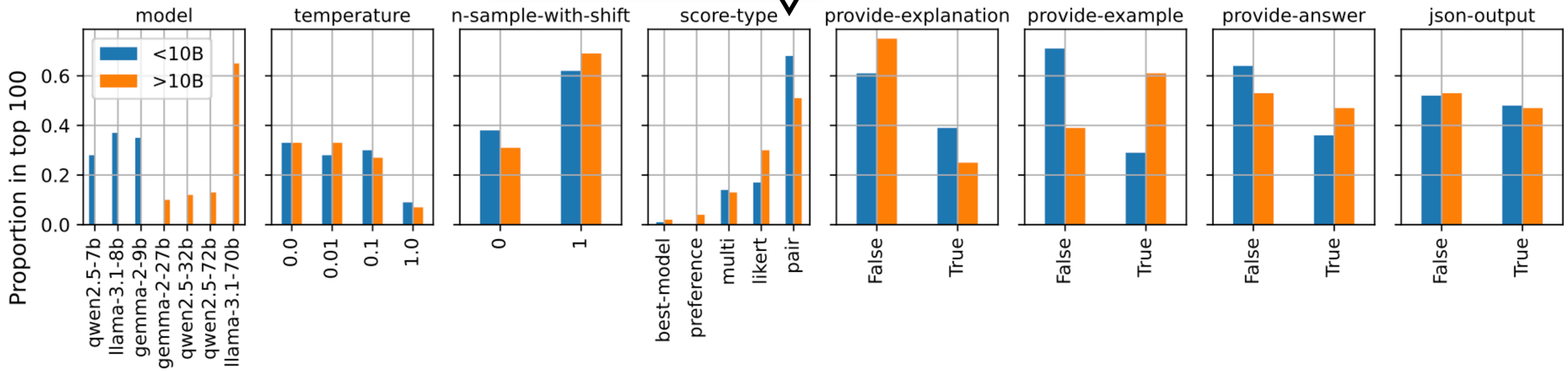


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

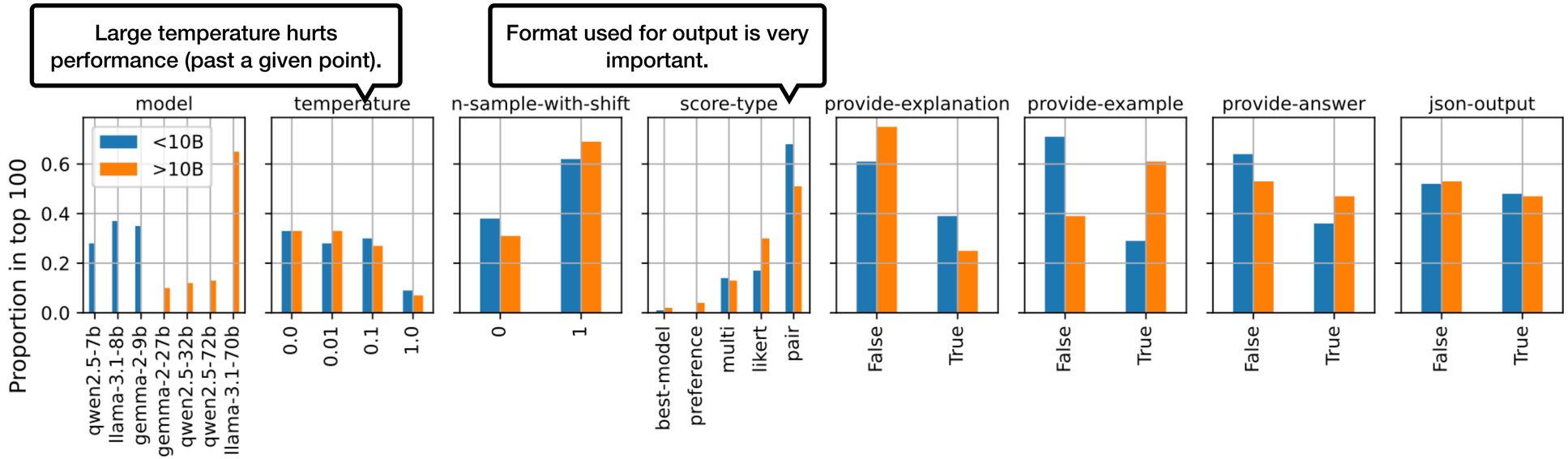


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

How many hyperparameters in the top 100 from the 4480 judges?

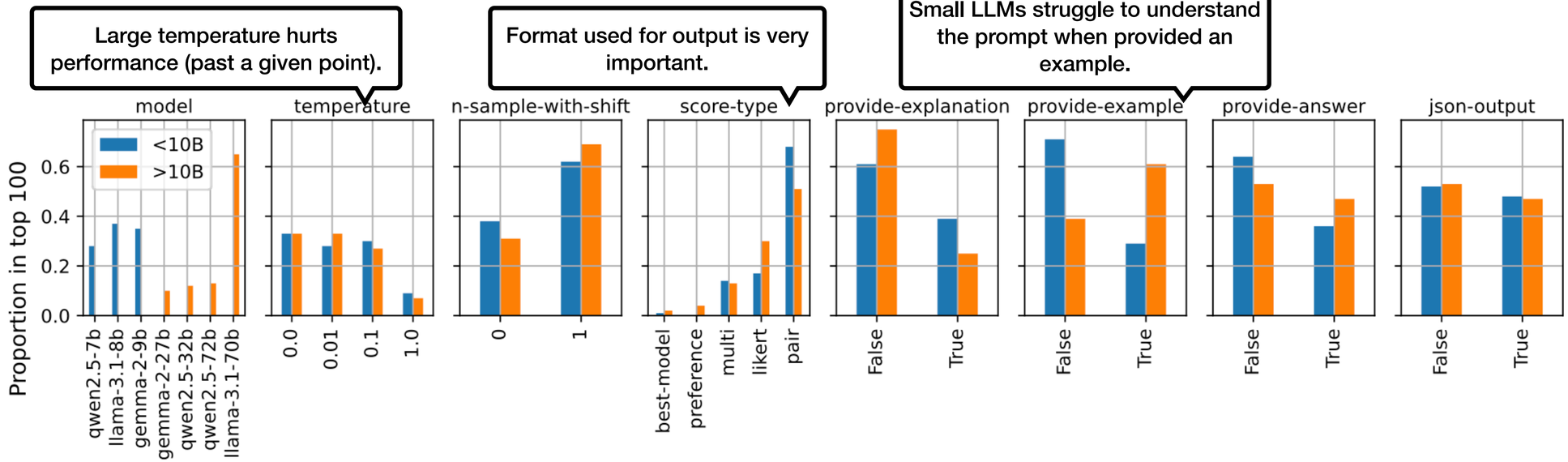


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Analysing judge performance

What hyperparameter/prompt works best?

- What is working best for LLM judges?

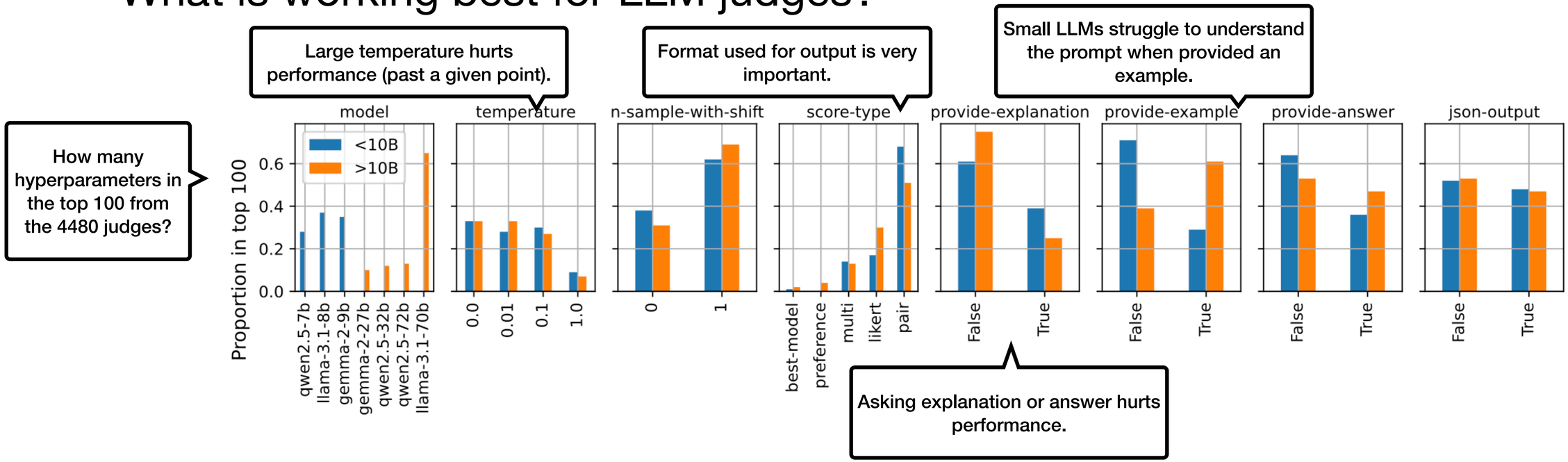


Figure 6. Fraction of time each hyperparameter appears in the top 100 configurations for small (<10B) and large models (>10B).

Prompt generalization

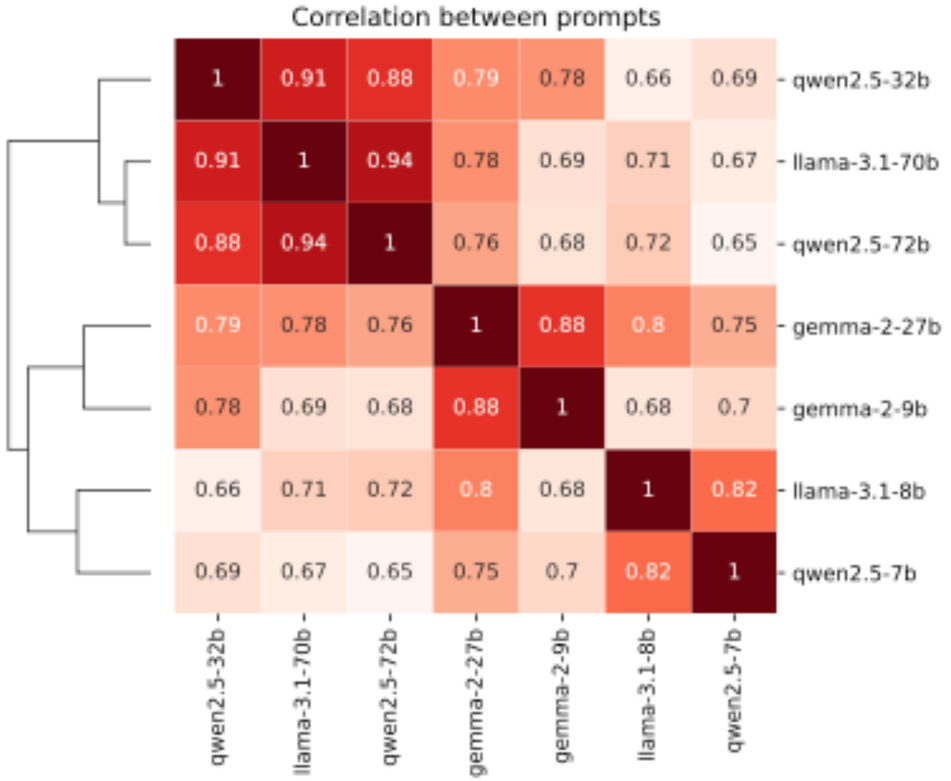
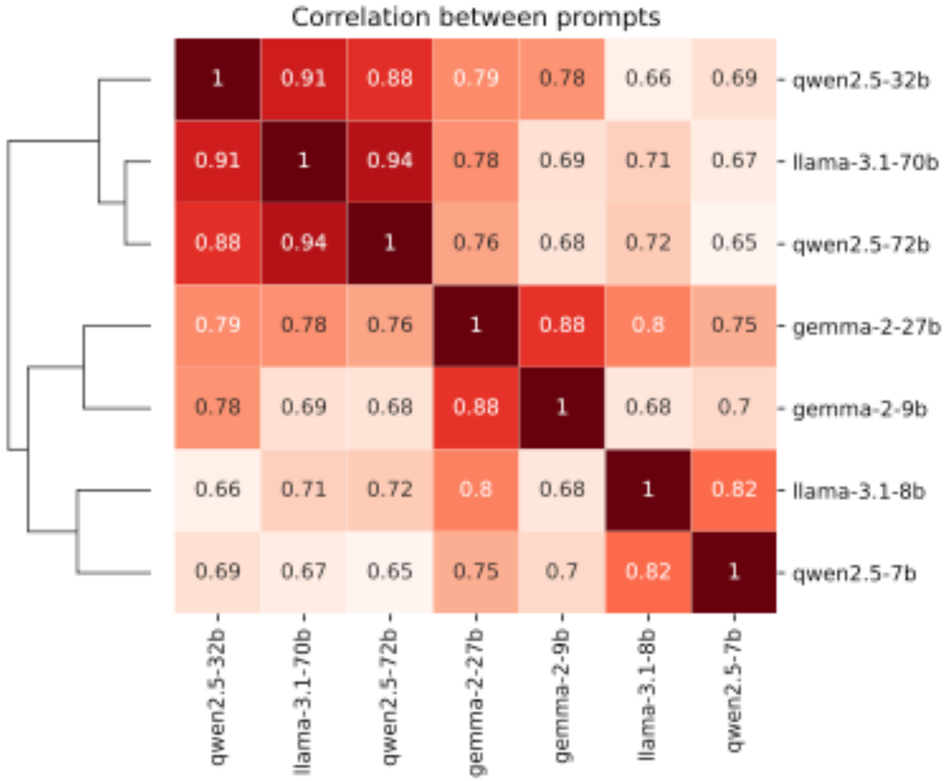


Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

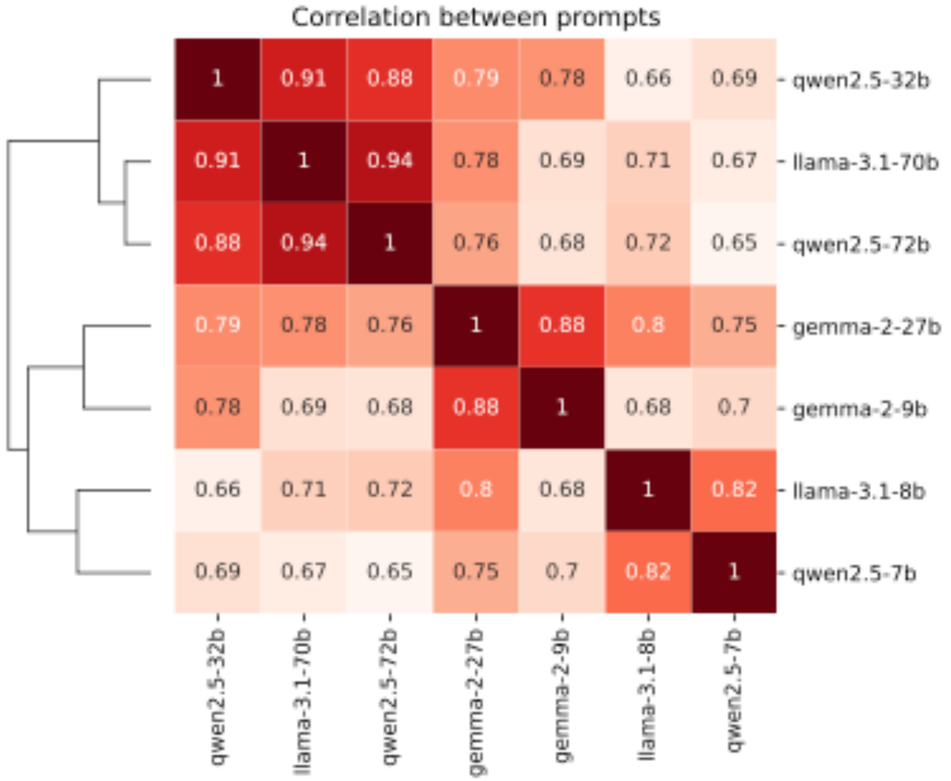
Prompt generalization



Correlation between rank of prompt configurations across models

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

Prompt generalization

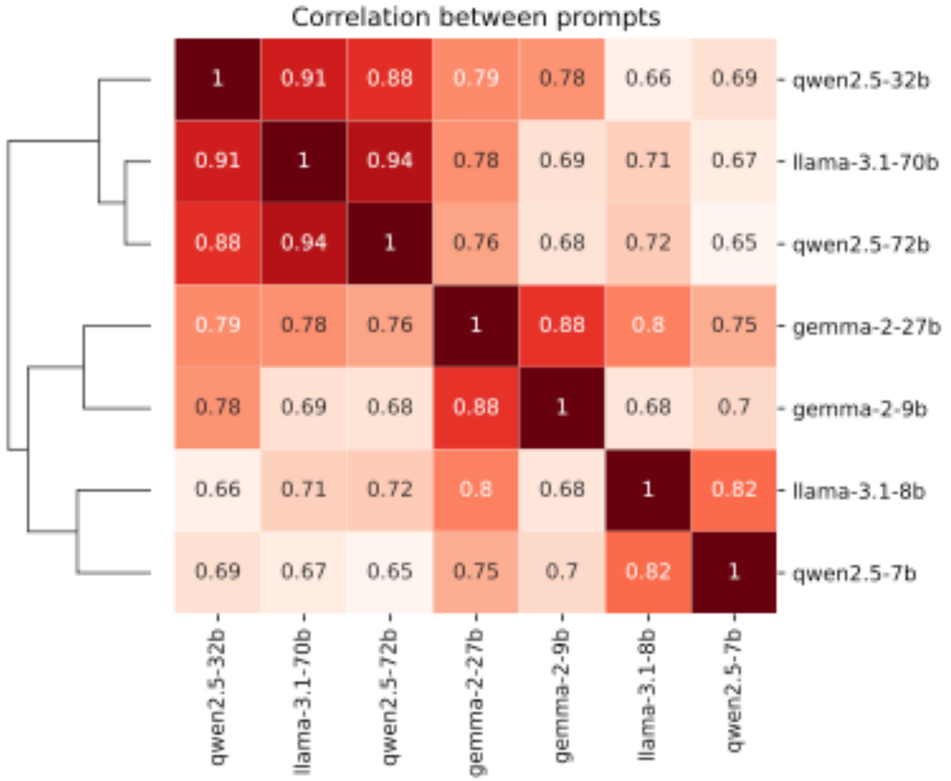


Correlation between rank of prompt configurations across models

High correlation => best prompt relatively stable across models

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

Prompt generalization



Correlation between rank of prompt configurations across models

High correlation => best prompt relatively stable across models

Correlation higher when model sizes and family are close

Figure 7. Prompt performance stability across different models. We show the correlation matrix between models when looking at their performance on all of the 80 different prompts.

```
You are a highly efficient assistant, who evaluates and selects the best large language
↪  model based on the quality of their responses to a given instruction.
You will be shown one instruction and the output of Assistant A and Assistant B and will
↪  have to decide which one was best.
Make sure to not over-confidently prefer one assistant or the other and also make sure to
↪  not bias your preference based on the ordering or on the length of the answers.

<|User Prompt|>
Who is Barack Obama?

<|The Start of Assistant A's Answer|>
Barack Obama is a former US president.
<|The End of Assistant A's Answer|>

<|The Start of Assistant B's Answer|>
I do not know who Barack Obama is.
<|The End of Assistant B's Answer|>

# Your output

## Format description
Your output should follow this format:
```
answer: <your answer to the user prompt>
score: <one of A>>B, A>B, A=B, A<B, A<<B, see instruction bellow>
```
The "score" value should indicate your preference for the assistant. You must output only
↪  one of the following choices as your final verdict with a label:

A>>B: Assistant A is significantly better
A>B: Assistant A is slightly better
A=B: Tie, relatively the same
B>A: Assistant B is significantly better
B>>A: Assistant B is significantly better

## Your output, do not repeat the input above
```
```

*Figure 10.* Example of a prompt for the user prompt "Who is Barack Obama?". In this case, the judge is asked to provide its answer. It is asked to use the Likert format and provide its answer in raw text.
```

Conclusion

Conclusion

- LLM judges can be tuned at a reasonable cost

Conclusion

- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches

Conclusion

- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly

Conclusion

- LLM judges can be tuned at a reasonable cost
- Tuning hyperparameters allows to match or outperform previous approaches
  - while diminishing the cost significantly
  - … and using only open-weight models

Next steps / future work

Next steps / future work

Use open-weight models to evaluate OpenEuroLLM instruction tuned models



Next steps / future work

Use open-weight models to evaluate OpenEuroLLM instruction tuned models



Optimize instruction tuning hyperparameters (DPO, learning-rate) that maximizes judge ratings

Pretrained model → Instruction-tuned model

Next steps / future work

Use open-weight models to evaluate OpenEuroLLM instruction tuned models



Optimize instruction tuning hyperparameters (DPO, learning-rate) that maximizes judge ratings

Pretrained model → Instruction-tuned model

Better AutoML for judges

Next steps / future work

Use open-weight models to evaluate OpenEuroLLM instruction tuned models



Optimize instruction tuning hyperparameters (DPO, learning-rate) that maximizes judge ratings

Pretrained model → Instruction-tuned model

Better AutoML for judges

- AutoML: tabular benchmark released https://github.com/geoalgo/judgetuning (together with code to reproduce results)
- Ensemble? Portfolio? Model-based optimizers?

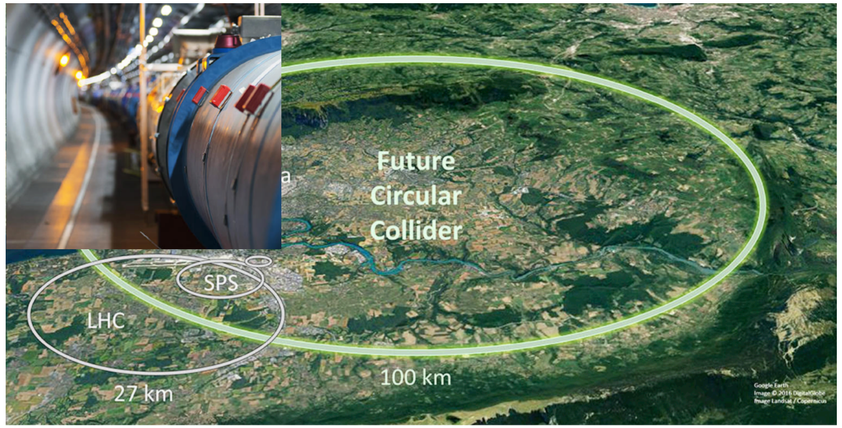
A case of openness

A case of openness

Some of humanity largest projects

A case of openness

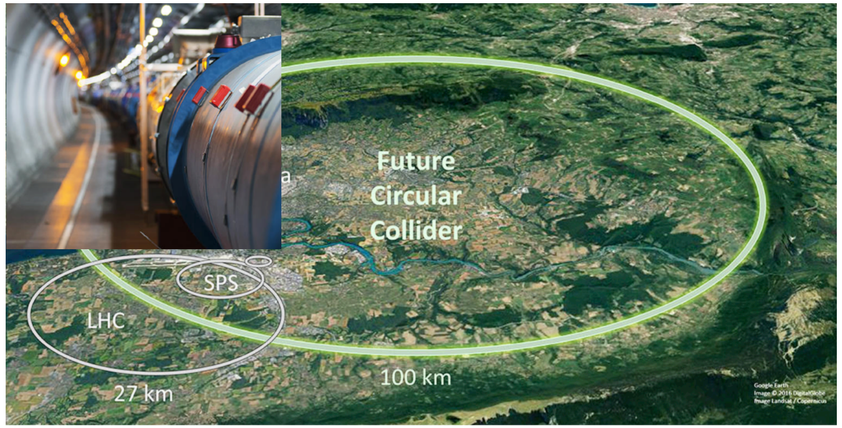
Some of humanity largest projects

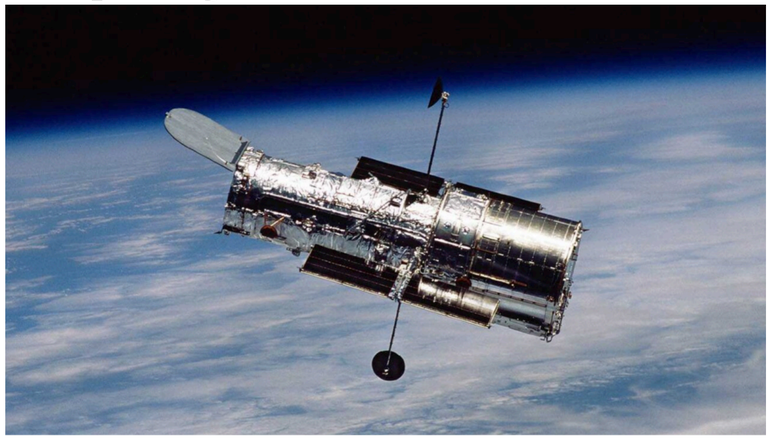


LHC: $5 Billion, 23 countries

A case of openness

Some of humanity largest projects

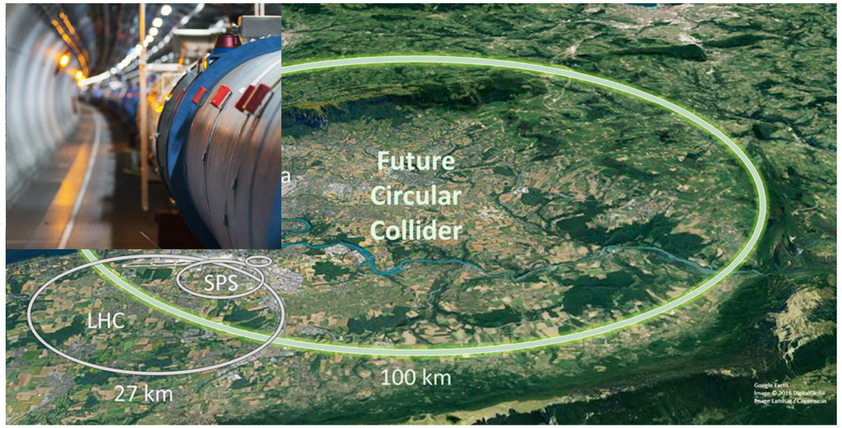




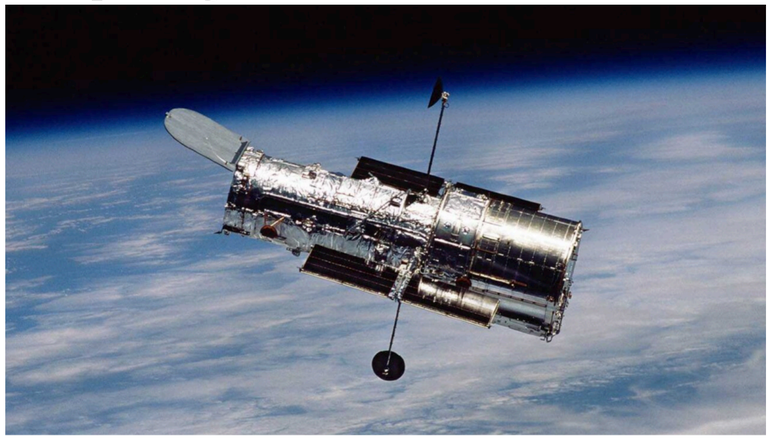
LHC: $5 Billion, 23 countries

A case of openness

Some of humanity largest projects



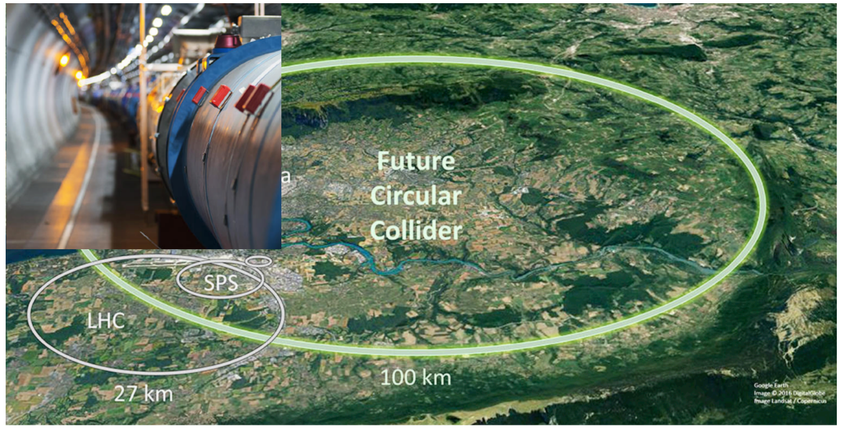
LHC: $5 Billion, 23 countries



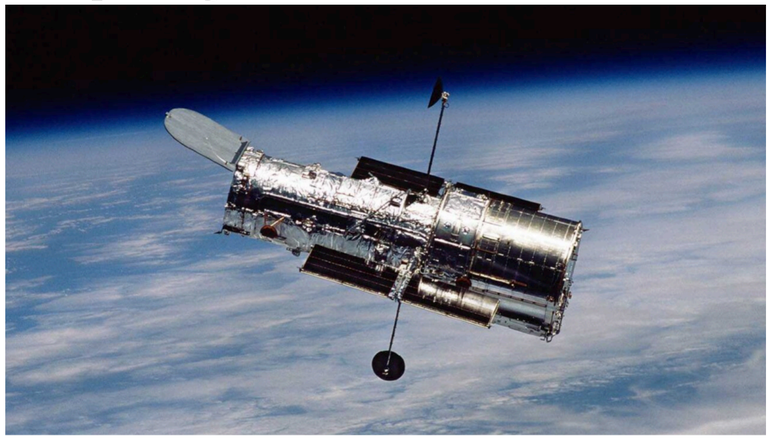
Hubble $16 billion, 11 countries

A case of openness

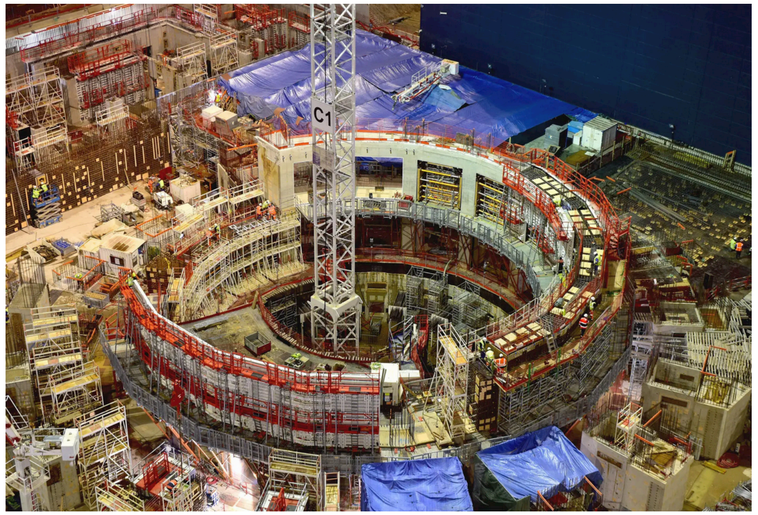
Some of humanity largest projects



LHC: $5 Billion, 23 countries

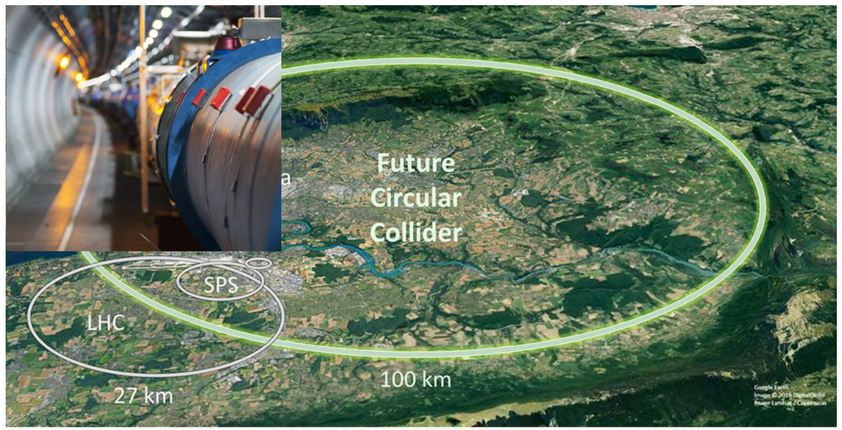


Hubble $16 billion, 11 countries

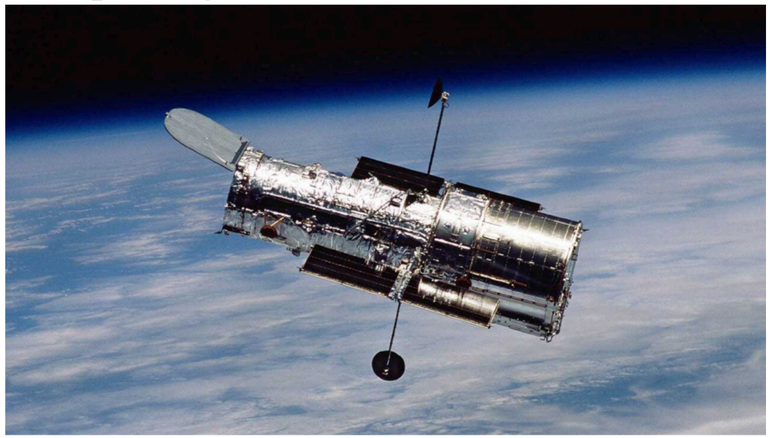


A case of openness

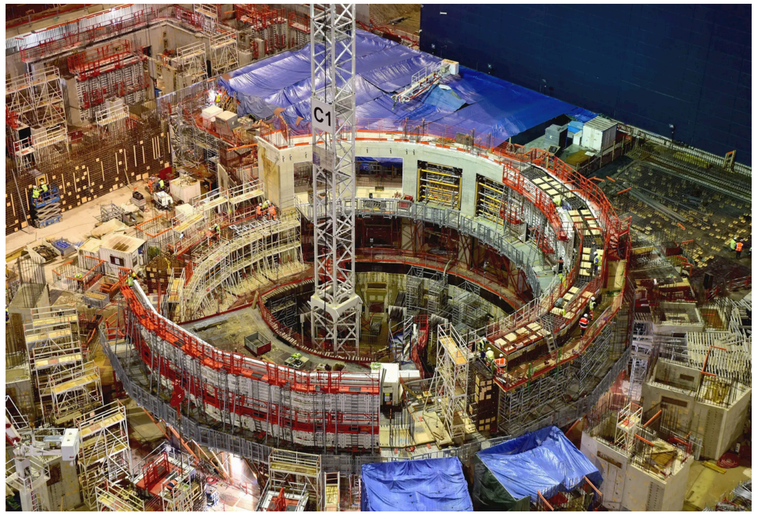
Some of humanity largest projects



LHC: $5 Billion, 23 countries



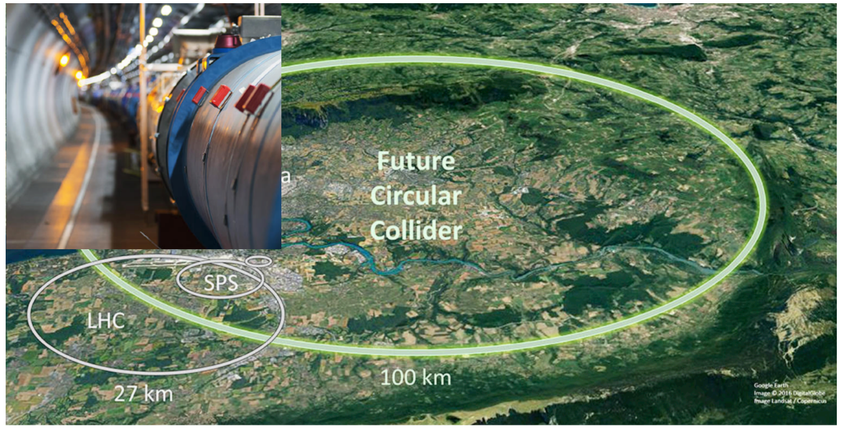
Hubble $16 billion, 11 countries



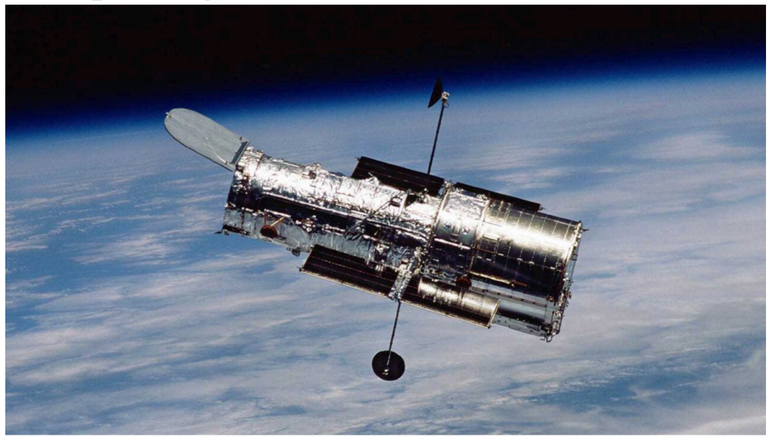
ITER: $45 Billion, 35 countries

A case of openness

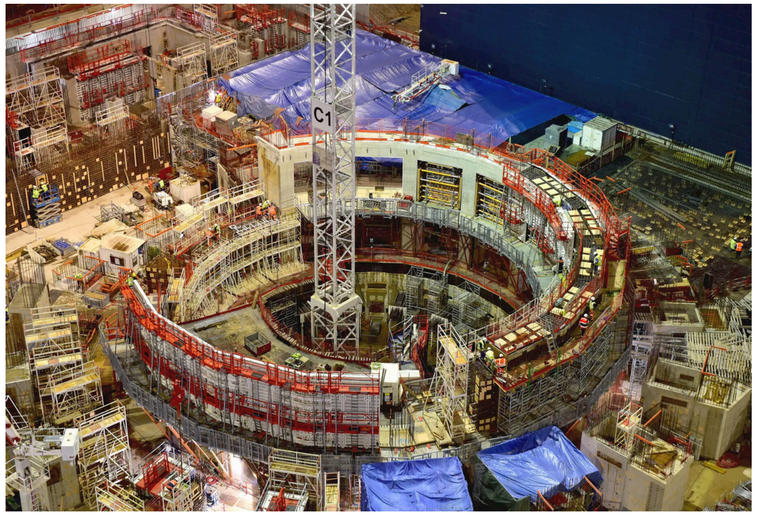
Some of humanity largest projects



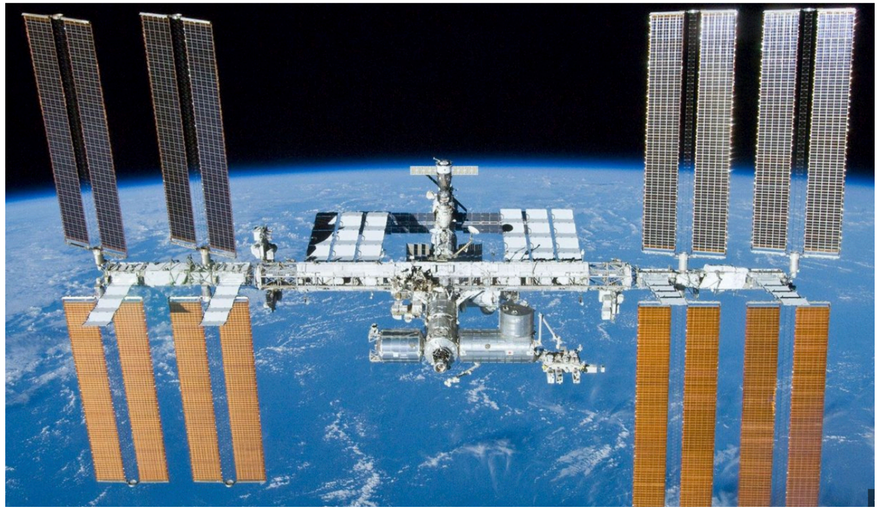
LHC: $5 Billion, 23 countries



Hubble $16 billion, 11 countries

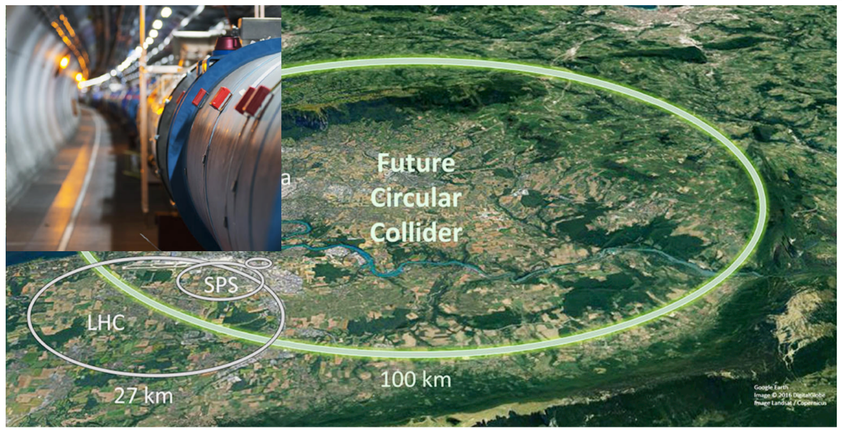


ITER: $45 Billion, 35 countries

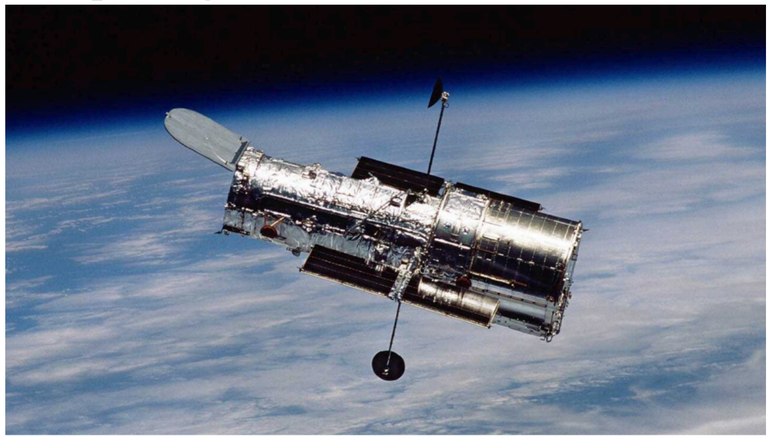


A case of openness

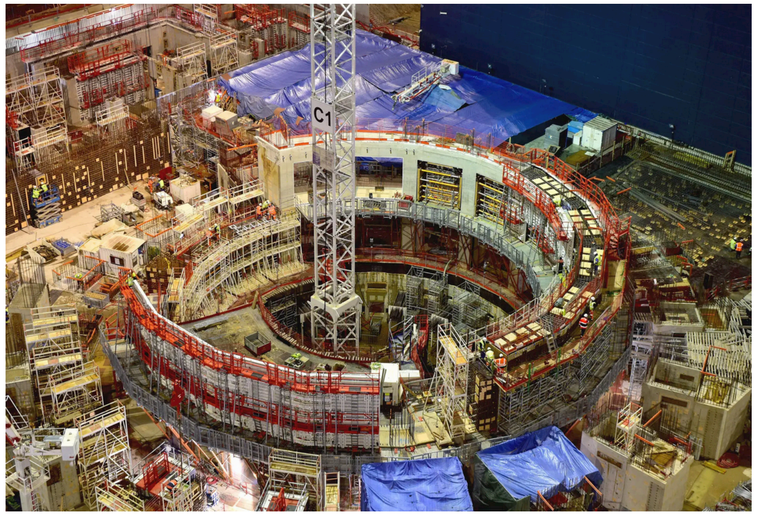
Some of humanity largest projects



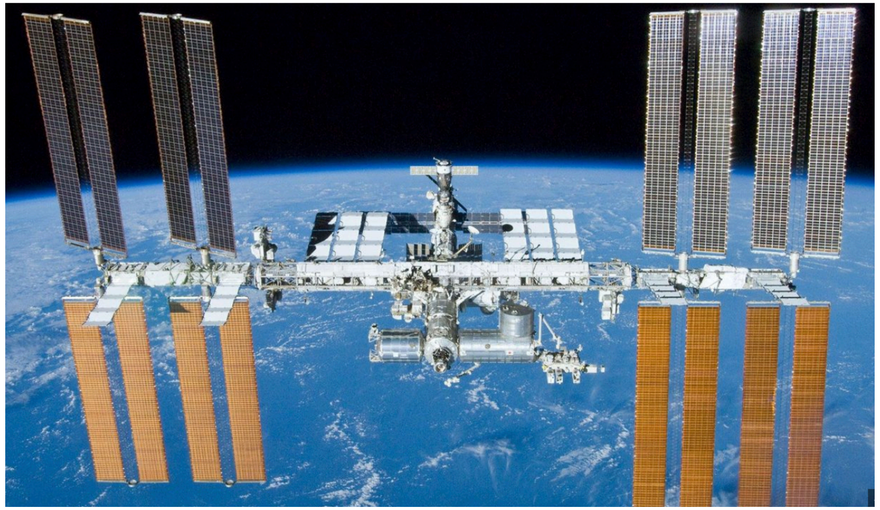
LHC: $5 Billion, 23 countries



Hubble $16 billion, 11 countries



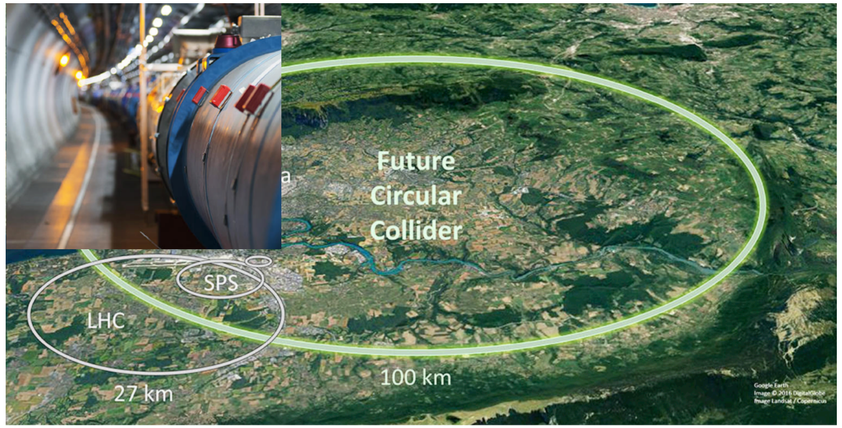
ITER: $45 Billion, 35 countries



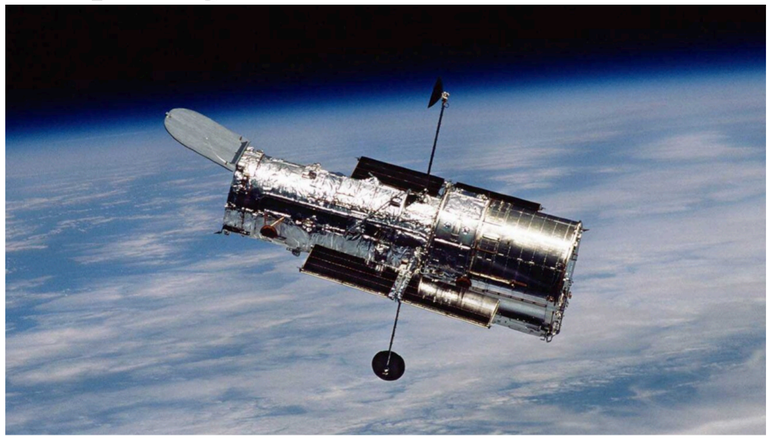
ISS: $100 Billion, 16 countries

A case of openness

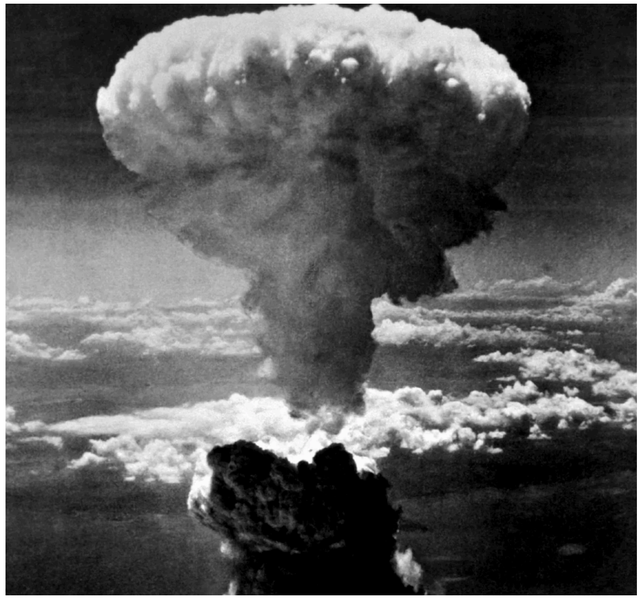
Some of humanity largest projects

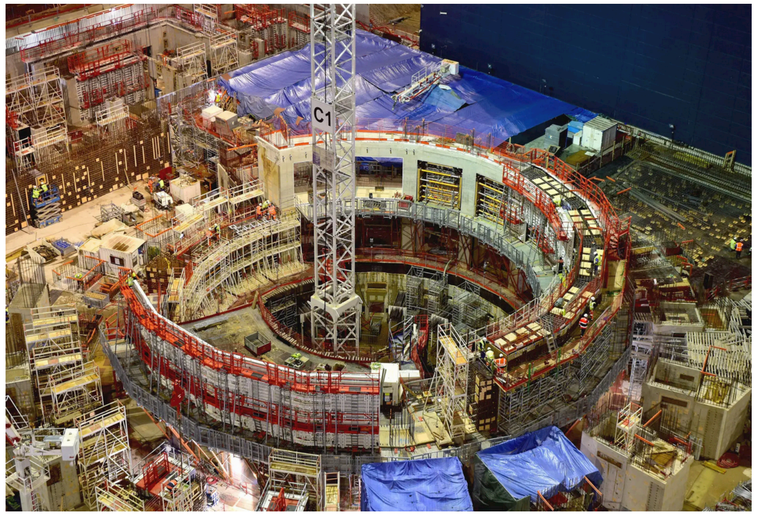


LHC: $5 Billion, 23 countries

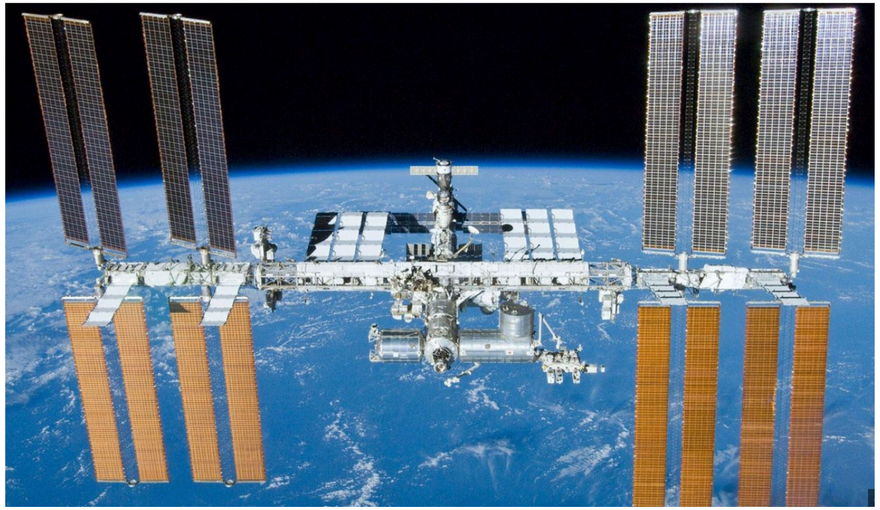


Hubble $16 billion, 11 countries





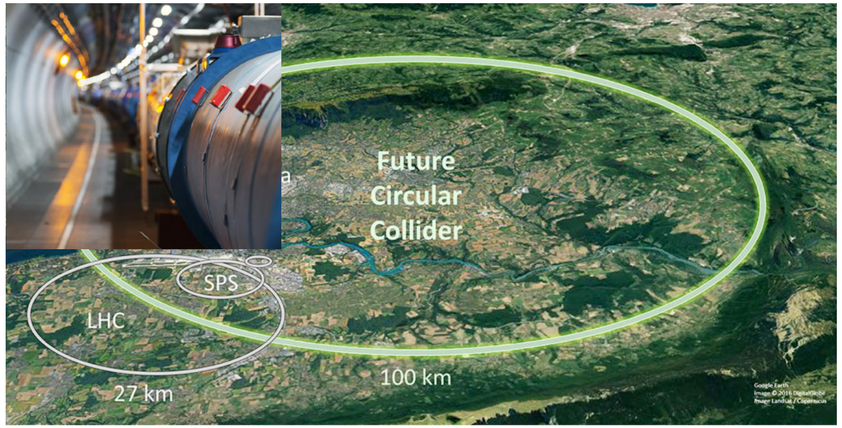
ITER: $45 Billion, 35 countries



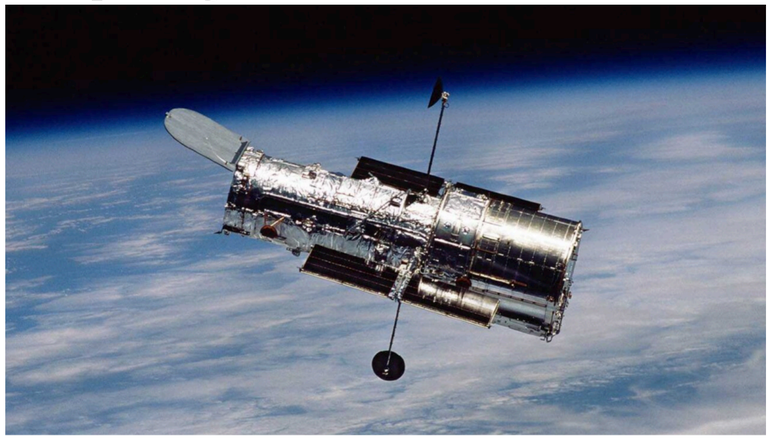
ISS: $100 Billion, 16 countries

A case of openness

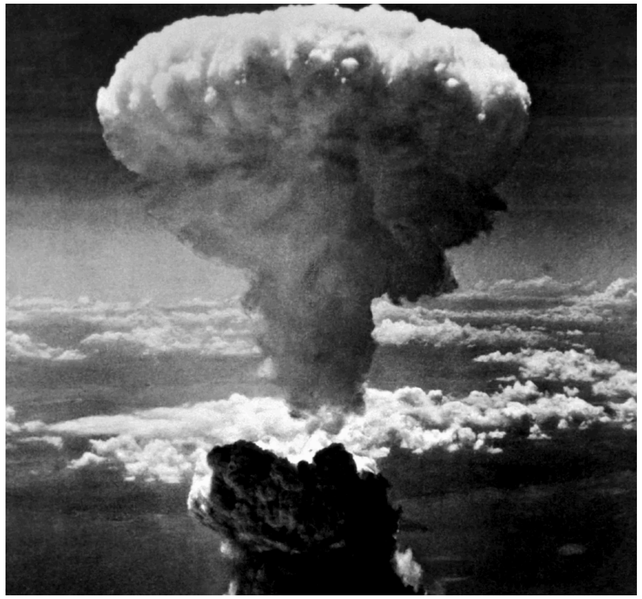
Some of humanity largest projects



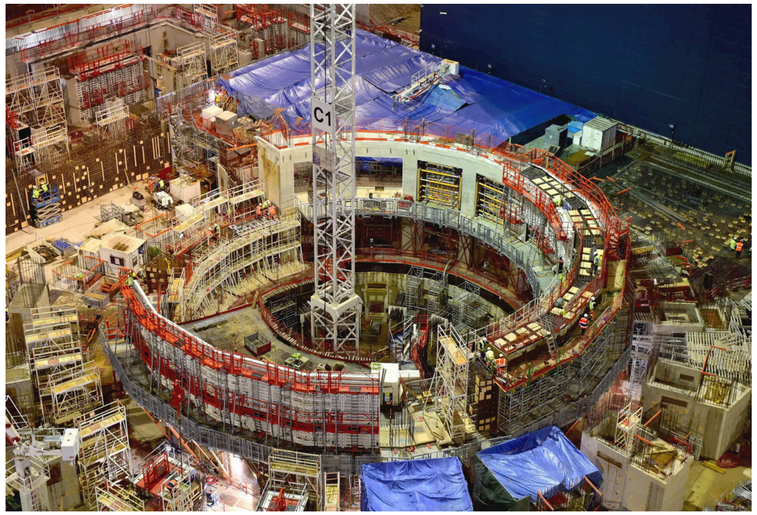
LHC: $5 Billion, 23 countries



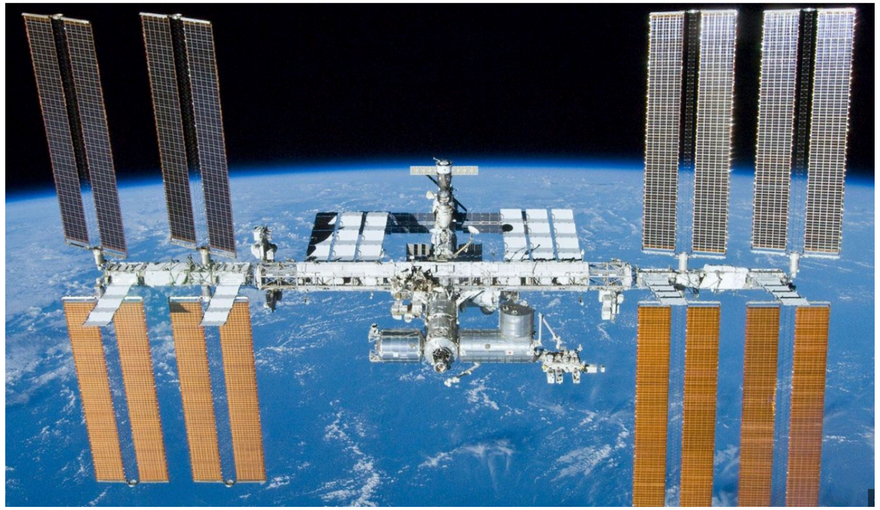
Hubble $16 billion, 11 countries



Manhattan project $30 billion, 3 countries



ITER: $45 Billion, 35 countries



ISS: $100 Billion, 16 countries

The case of LLMs

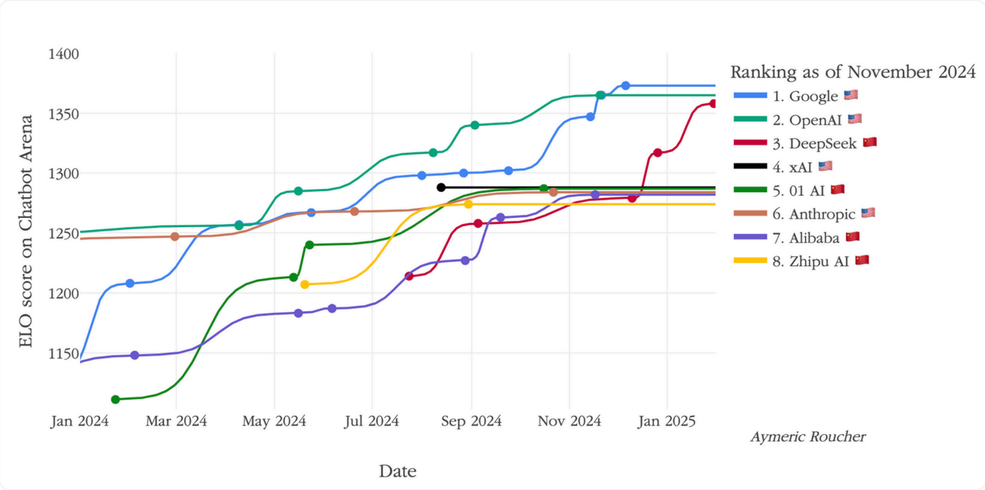
The case of LLMs

- Currently an arm race

The case of LLMs

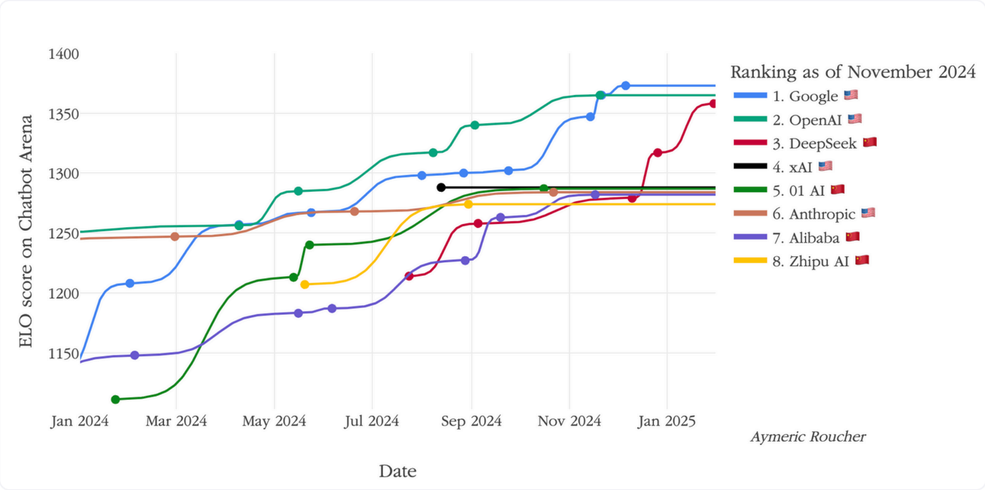
- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time

The case of LLMs



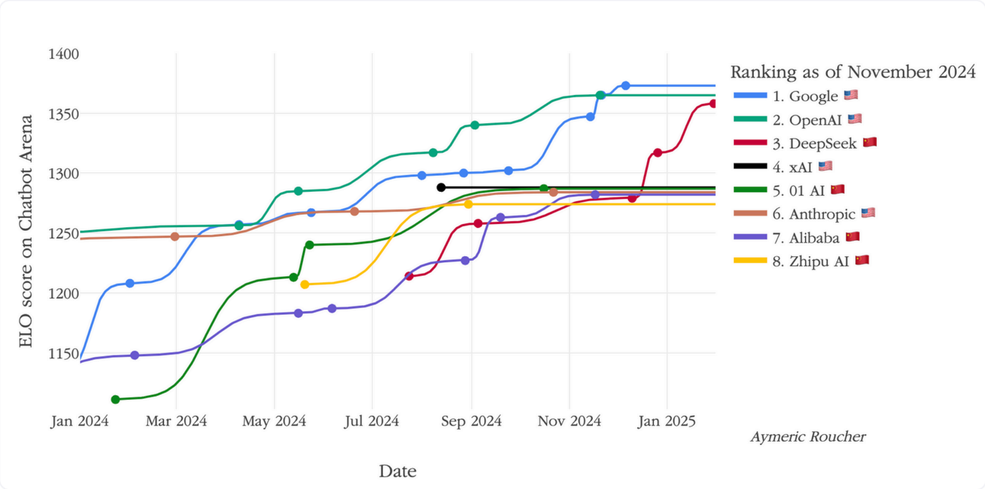
- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time

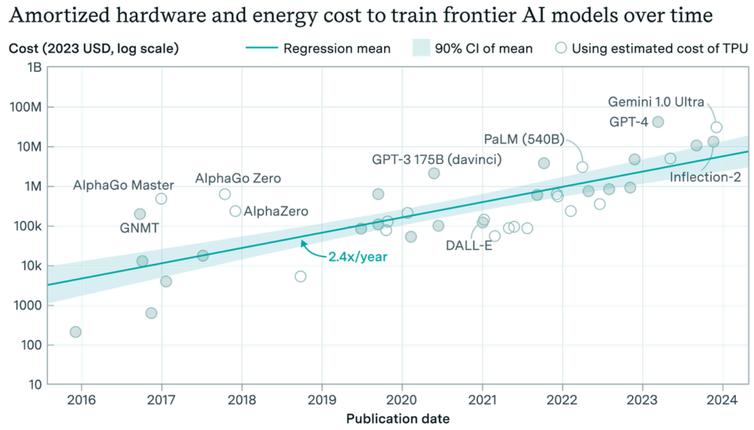
The case of LLMs



- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)

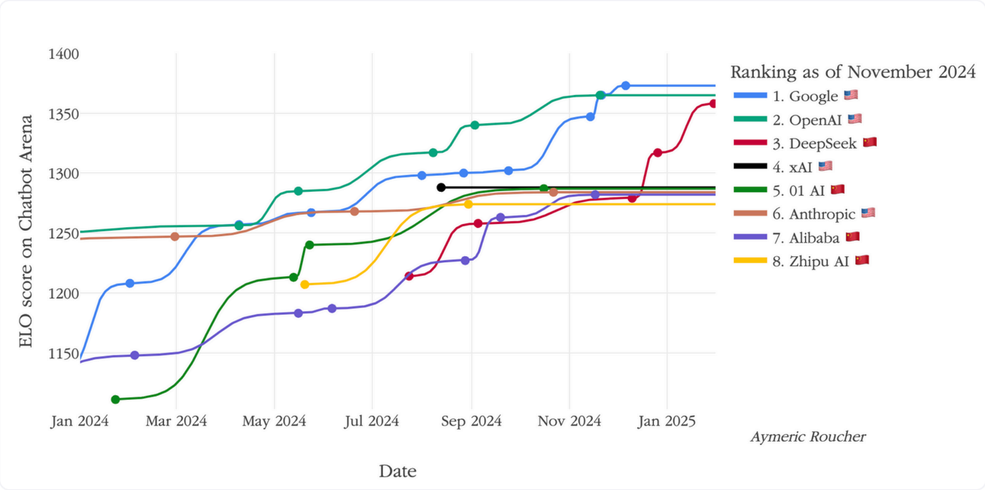
The case of LLMs

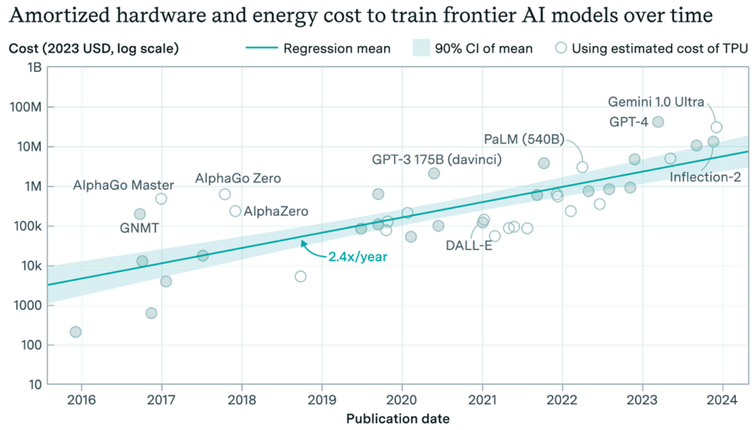




- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)

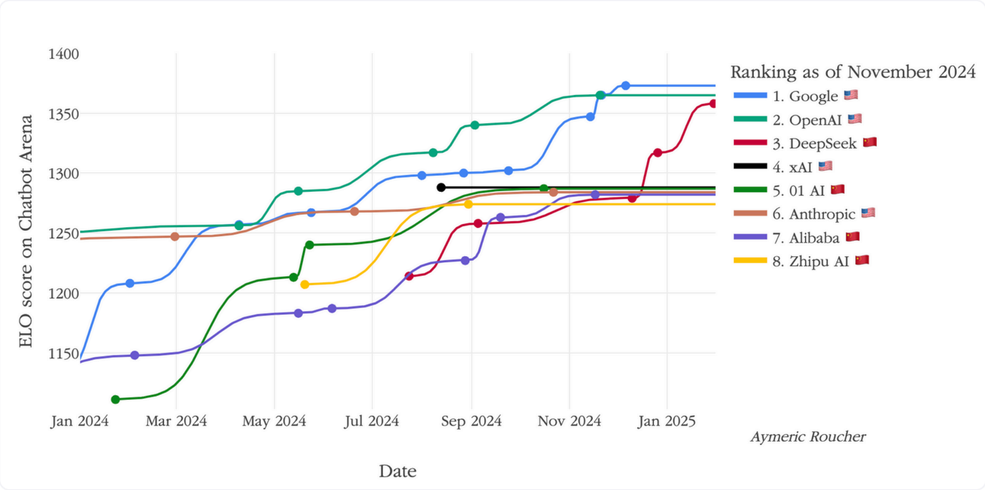
The case of LLMs

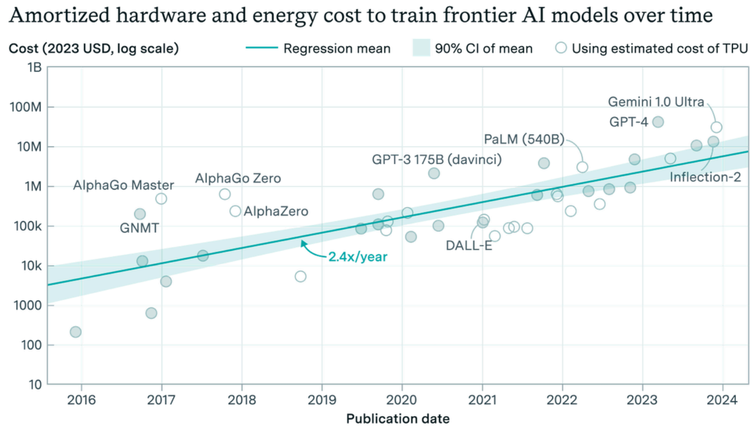




- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)
  - Algorithmic progress: ~4x/year? https://www.darioamodei.com/post/on-deepseek-and-export-controls

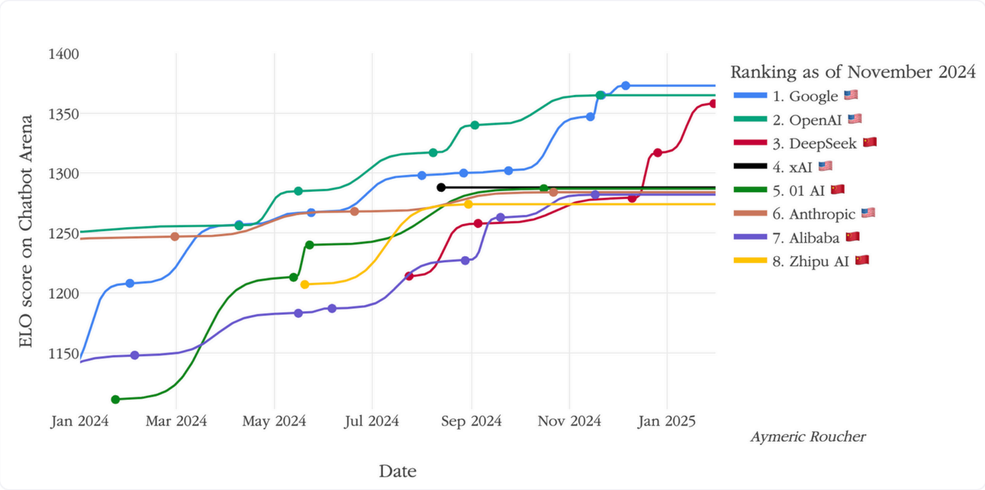
The case of LLMs

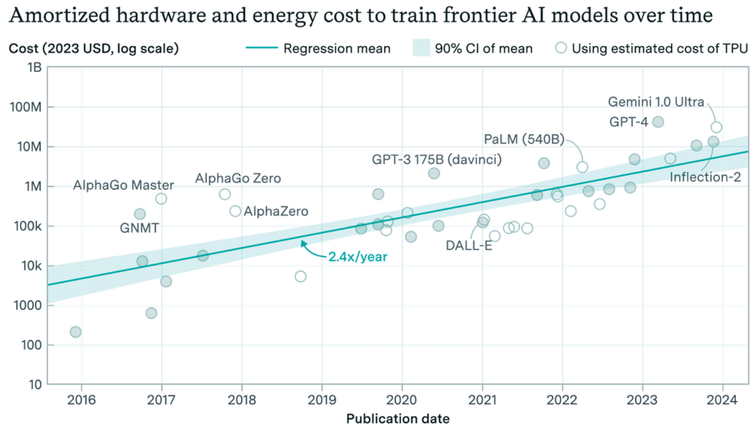




- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)
  - Algorithmic progress: ~4x/year? https://www.darioamodei.com/post/on-deepseek-and-export-controls
- Alternate model: companies & universities sharing open-weight models and sometimes fully open models

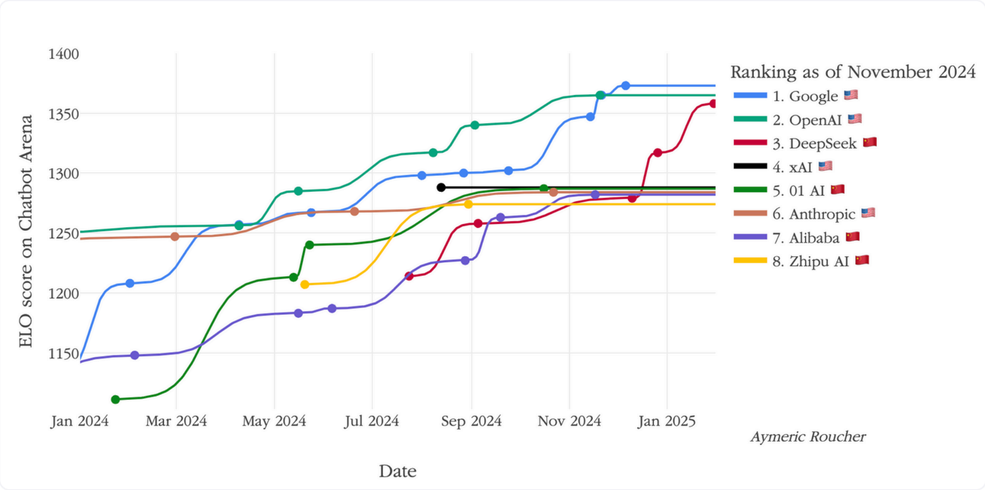
The case of LLMs

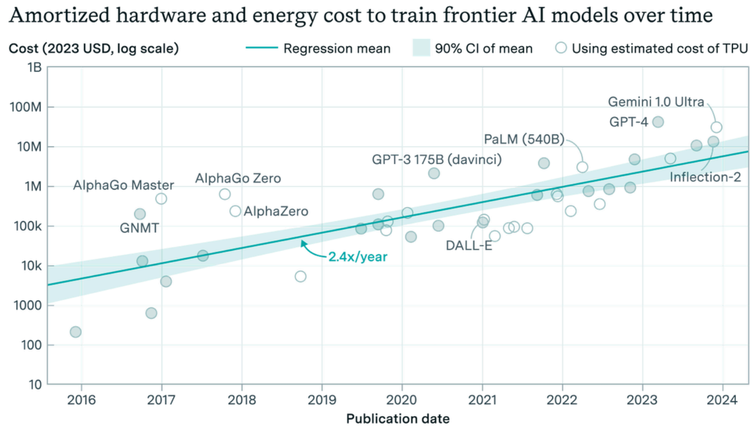




- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)
  - Algorithmic progress: ~4x/year? https://www.darioamodei.com/post/on-deepseek-and-export-controls
- Alternate model: companies & universities sharing open-weight models and sometimes fully open models
  - Open-weights: Meta, Google, Mistral, …

The case of LLMs





- Currently an arm race
  - One world with N actors developing N models and sharing less and less over time
  - Scaling compute efficiency (the bitter lesson from Sutter)
  - Algorithmic progress: ~4x/year? https://www.darioamodei.com/post/on-deepseek-and-export-controls
- Alternate model: companies & universities sharing open-weight models and sometimes fully open models
  - Open-weights: Meta, Google, Mistral, …
  - Fully open models: Stanford, AllenAI institute, Apple …

Collaboration for profit is possible!

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI



An EUV machine, $380 million

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI



An EUV machine, $380 million

It prints features of just a few nanometers; it is the key technology required to build GPU chips

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI
- EUV development costed ~$14-21 billion and involved **many** companies and countries over decades of research



An EUV machine, $380 million

It prints features of just a few nanometers; it is the key technology required to build GPU chips

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI
- EUV development costed ~$14-21 billion and involved **many** companies and countries over decades of research
- Currently built only by ASML, an EU company 🇪🇺🇳🇱🌷



An EUV machine, $380 million

It prints features of just a few nanometers; it is the key technology required to build GPU chips

Collaboration for profit is possible!

- AI rely on high-performance GPUs which relies on one of the biggest industrial collaboration in history!
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core** technology powering AI
- EUV development costed ~$14-21 billion and involved **many** companies and countries over decades of research
- Currently built only by ASML, an EU company 🇪🇺🇳🇱🌷
- Key geopolitical stake



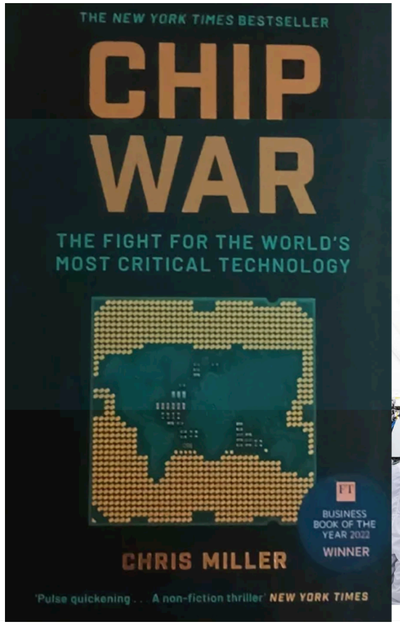
An EUV machine, $380 million

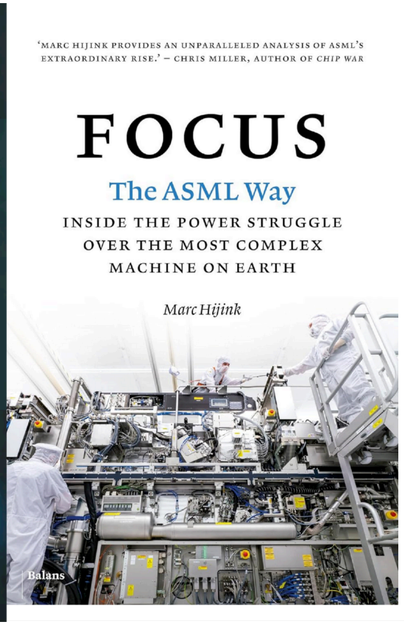
It prints features of just a few nanometers; it is the key technology required to build GPU chips

Collaboration for profit is possible!

- AI rely on high-performance GPUs wh
  biggest industrial collaboration in histo
- What is this technology? 🤔
- Extreme Ultraviolet (EUV) is **the core t**
- EUV development costed ~$14-21 bill
  companies and countries over decade
- Currently built only by ASML, an EU c
- Key geopolitical stake

Recommended reading 📚







An EUV machine, $380 million

It prints features of just a few nanometers; it is the key technology required to build GPU chips

OpenEuroLLM

Universities and Research Organizations

Charles University
AI Sweden
alt-edic
EDIA
Eberhard Karls Universität Tübingen
ellis Institute Tübingen
Fraunhofer IAIS
Barcelona Supercomputing Center
Centro Nacional de Supercomputación
Jülich Forschungszentrum
TU/e Eindhoven University of Technology
University of Helsinki
University of Oslo
University of Turku

Companies

Aleph Alpha
AMD Silo AI
ellamind
Lighton
prompsit

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC

Universities and Research Organizations

Charles University
AI Sweden
alt-edic et DEa
Eberhard Karls Universität Tübingen
ellis Institute Tübingen
Fraunhofer IAIS
Barcelona Supercomputing Center Centro Nacional de Supercomputación
Jülich Forschungszentrum
TU/e Eindhoven University of Technology
University of Helsinki
University of Oslo
University of Turku

Companies

Aleph Alpha
AMD Silo AI
ellamind
Lighton
prompsit

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models

Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models



Reference analysis training 1.7B models from scratch for different datasets

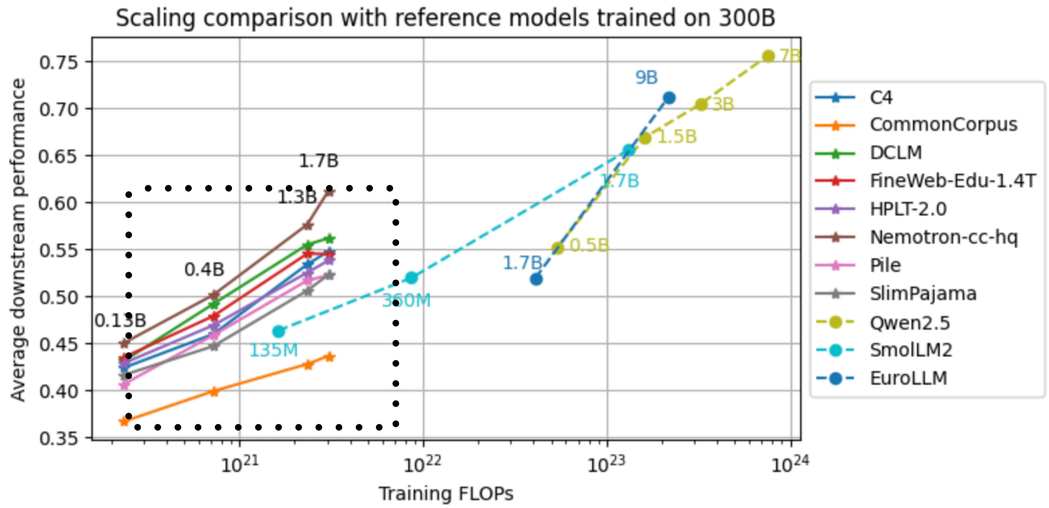
Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models



Reference analysis training 1.7B models from scratch for different datasets

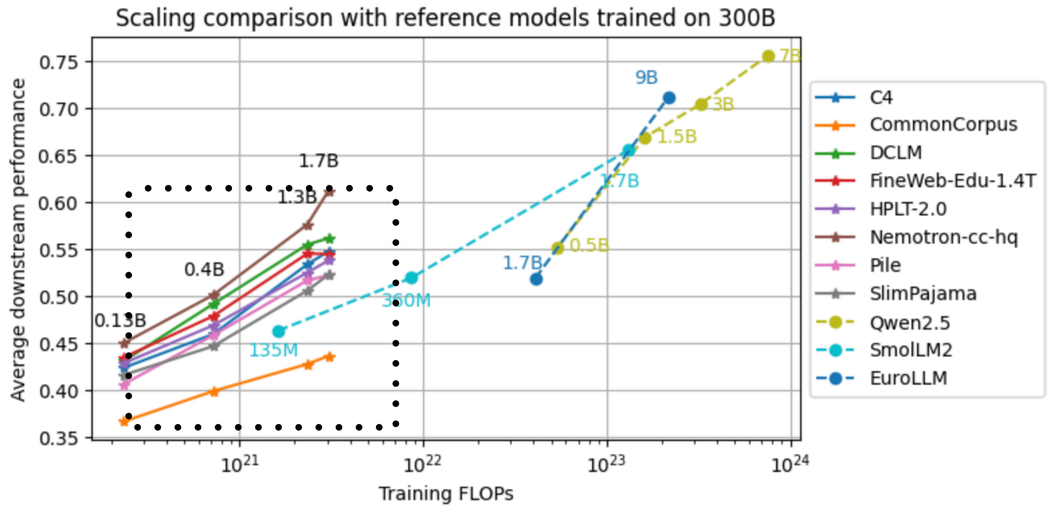
Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models
- Currently hiring 9 ML researchers / engineers at ELLIS! Also internships 👋



Reference analysis training 1.7B models from scratch for different datasets

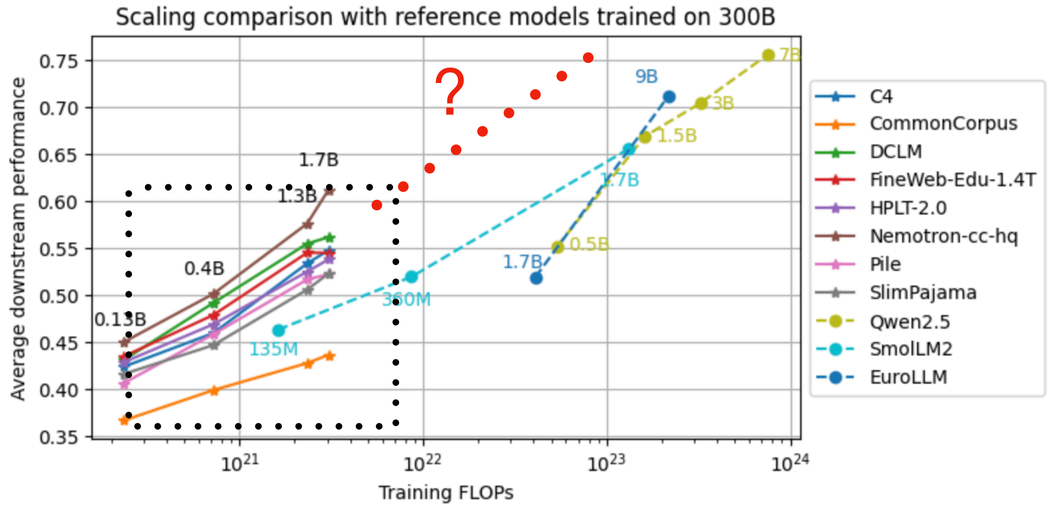
Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models
- Currently hiring 9 ML researchers / engineers at ELLIS! Also internships 👋



Reference analysis training 1.7B models from scratch for different datasets

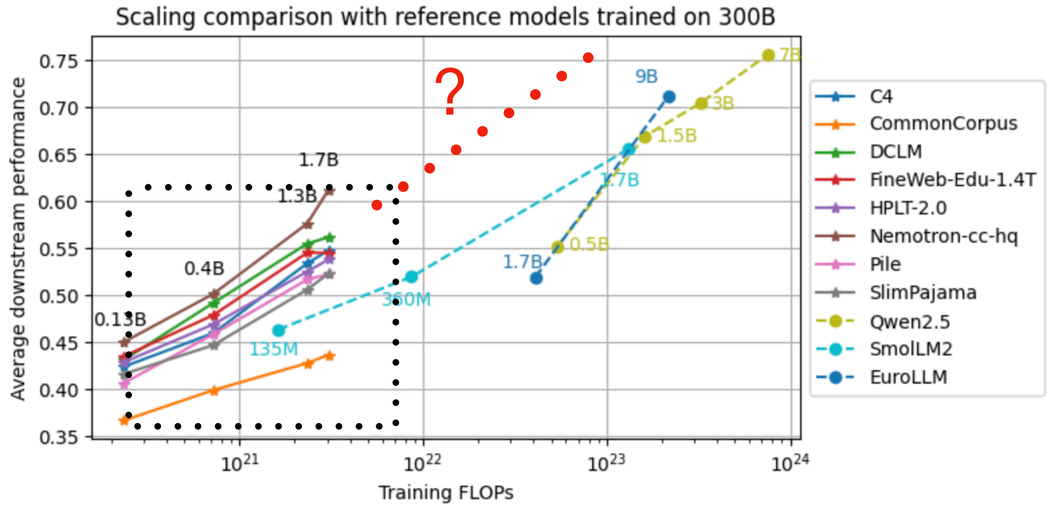
Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models
- Currently hiring 9 ML researchers / engineers at ELLIS! Also internships 👋
- Ping me and Aaron if interested 🤗



Reference analysis training 1.7B models from scratch for different datasets

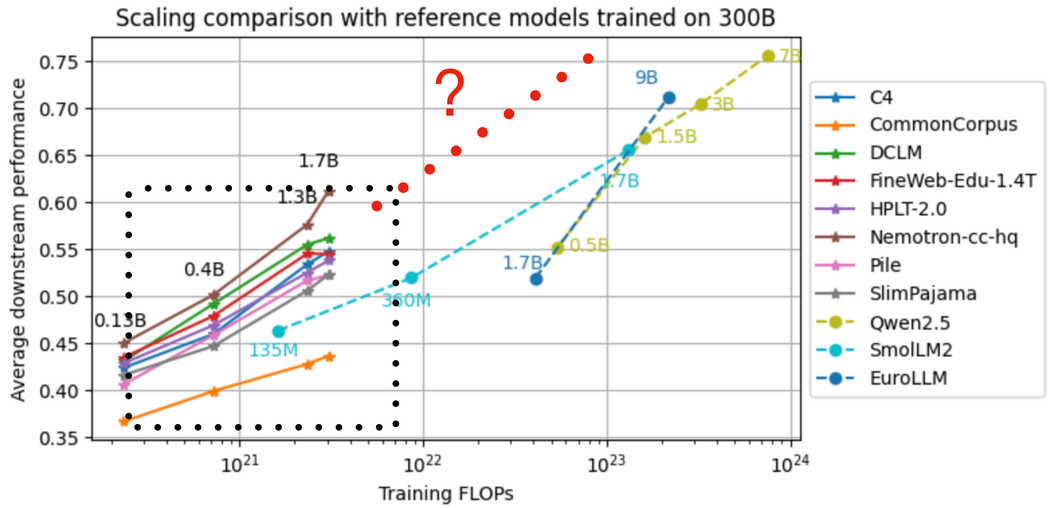
Universities and Research Organizations

Companies

Co-funded by the European Union

OpenEuroLLM

- An effort to build multilingual LLMs from scratch by 2028
  - Started in February 2025
  - Fully open: weights & code & data
  - €37.4 million funding. In addition many millions of GPU hours in EuroHPC
- Will release soon reference 1.7B models with SOTA performance among fully open models
- Currently hiring 9 ML researchers / engineers at ELLIS! Also internships 👋
- Ping me and Aaron if interested 🤗
- Lots of areas for AutoML in pre-training, post-training, evaluation 🎉



Reference analysis training 1.7B models from scratch for different datasets

Universities and Research Organizations

Companies

Co-funded by the European Union

Conclusion

Conclusion

- LLM evaluation is an open problem
  - Fundamental difficulty to evaluate open-ended answers
  - Need to handle many dimensions (objectives, languages, cost)

Conclusion

- LLM evaluation is an open problem
  - Fundamental difficulty to evaluate open-ended answers
  - Need to handle many dimensions (objectives, languages, cost)
- AutoML has a lot to say
  - Many objectives => multiobjective optimization
  - Costly =>
    - Multifidelity optimization
    - Transfer/meta-learning, portfolio, …

Conclusion

- LLM evaluation is an open problem
  - Fundamental difficulty to evaluate open-ended answers
  - Need to handle many dimensions (objectives, languages, cost)
- AutoML has a lot to say
  - Many objectives => multiobjective optimization
  - Costly =>
    - Multifidelity optimization
    - Transfer/meta-learning, portfolio, …
- AutoML all the way for evaluations (LLM-judge), instruction tuning and maybe pretraining?

Questions