

# LLM Evaluation

COMP6211J

Binhang Yuan



# Overview

- A principled benchmark is essential to understand the capacity of LLMs.
- We will discuss:
  - The quality evaluation format and metrics.
  - Concrete quality benchmark examples:
    - General-purpose benchmarks.
    - Coding benchmarks.
    - Math benchmarks.
  - LLM service system evaluation.



**RELAXED**  
SYSTEM LAB

# LLM Evaluation Metric for Quality



# Quality Evaluation Format and Metric

- **Multiple-Choice Question (MCQ):**

- Models answer fixed-option questions (A/B/C/D, etc.), selecting one correct choice.
- Evaluation metric: *accuracy*, i.e., the percentage of questions answered correctly.

- **Coding Problem:**

- Models generate code to solve programming problems. The prompt might be a function description or a problem statement, and the model outputs code.
- Execute the generated code against the tests.
- Evaluation metric: a common metric is *pass@k* – e.g., pass@1 is the percentage of problems solved on the first attempt, and pass@k indicates if at least one of k tries passes all tests.



# Evaluation Format and Metric

RELAXED  
SYSTEM LAB

- **Open-ended Question:**

- Evaluation method 1- exact match: a stricter form of *accuracy* commonly used in QA evaluation. EM requires the model's output to match the gold answer *exactly*, down to every character. If there's any deviation (missing a word, extra punctuation, etc.), it counts as incorrect.
- Evaluation method 2 – partial match: compute the precision, recall, and f1 score between the *tokens* in the gold answer and the generated answer:
  - **Precision**: the percentage of the answer that is actually correct;
  - **Recall** is the percentage of the gold answer that the model captured.
  - **F1 Score**: The harmonic mean of precision and recall,  $F1 = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$ .



# Partial Match Example

- Question: "What are the symptoms of influenza?"
  - Gold Answer: "Fever, cough, sore throat, muscle aches, fatigue."
  - LLM Response: "Fever, cough, sore throat, fatigue."
- Tokenization:
  - Reference Tokens: ["fever", "cough", "sore", "throat", "muscle", "aches", "fatigue"]
  - LLM Response Tokens: ["fever", "cough", "sore", "throat", "fatigue"]
  - Correctly Matched Tokens: ["fever", "cough", "sore", "throat", "fatigue"] (Total Matched: 5)
- Calculating Metrics:
  - Precision = Matched Tokens / Total Tokens in LLM Response =  $5 / 5 = 1.0$
  - Recall = Matched Tokens / Total Tokens in Reference Answer =  $5 / 7 \approx 0.714$
  - F1 Score =  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \approx 0.833$



# Evaluation Format and Metric

RELAXED  
SYSTEM LAB

- **Open-ended Question:**

- Evaluation method 3 - human evaluation: human judgment is the ultimate metric -- but it is usually very hard to scale. Format includes:
  - ***Likert scale ratings:*** e.g., rating an output's quality from 1–5;
  - ***Pairwise comparisons:*** show two model outputs and ask which is better.
- Evaluation method 4 – LLM-as-a-Judge: Have an AI model evaluate outputs to approximate human judgment at scale;
  - This approach should be validated to ensure it aligns with true human preferences.
  - Can generate a similar format to human subjective evaluation.



# Human Evaluation

- Human evaluation can capture more complicated qualities like helpfulness, coherence, creativity, and safety, e.g:
  - Factual accuracy in open-ended answers;
  - Non-deterministic correctness (like whether an explanation is convincing);
  - User preference (which response they found more useful).
  - Issues like a model being too verbose or too terse – things hard to quantify automatically.
- **Pairwise comparisons**: A powerful strategy is to show a human two model outputs for the same prompt and ask which is better (or if they are equal):
  - Easier for people (choosing A vs B);
  - More consistent.
- **Repeated pairwise battles can produce a ranking of models.**



# Human Evaluation — Chatbot Arena

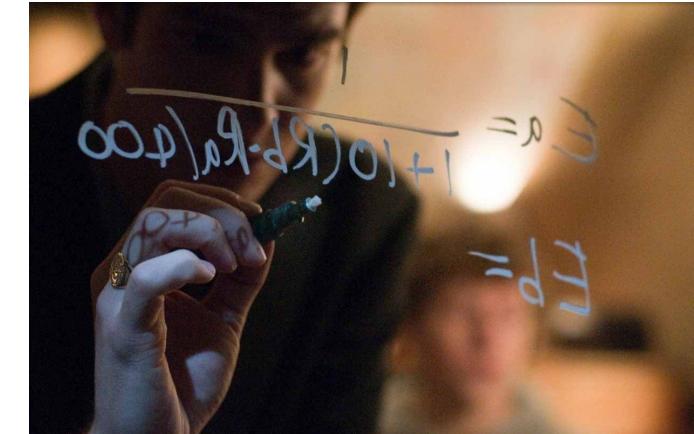
- Basic idea: ELO Rating System:
  - Performance is not measured absolutely; it is inferred from wins, losses, and draws against other players.
  - A player's rating, i.e., expected score  $R$ , is their probability of winning plus half their probability of drawing.
  - The rating goes up or down based on wins/losses/draws in these pairwise comparisons.

- Expected score calculation:

- Suppose there are two players, A and B, with ratings  $R_A$  and  $R_B$ .
- The system predicts the expected outcome of a match between them, e.g., the expected score for Player A against Player B is calculated using the formula:  $E_A = \frac{1}{1+10^{(R_A-R_B)/400}}$

- Rating update after a match:

- After the match, Player A's rating is updated based on the actual outcome compared to the expected score:  $R'_A = R_A + K(S_A - E_A)$ .
  - $R'_A$  is the new rating for Player A;
  - $K$  is the development coefficient (commonly 10, 20, or 40), determining the sensitivity of rating changes;
  - $S_A$  is the actual score achieved by Player A (1 for a win, 0.5 for a draw, 0 for a loss)



*The Social Network*



# Human Evaluation — Chatbot Arena

- Chatbot Arena: allows users to chat with two anonymous models side-by-side and then vote which response is better (or declare a tie).
- This generates large-scale pairwise comparison data in a crowd-sourced manner.
- Users don't know which model is which (they are usually labeled Model A and Model B randomly).
- This prevents brand bias (e.g., “I'll pick GPT-4 because I know it's GPT-4”) and focuses on the content quality.

The screenshot shows the homepage of Chatbot Arena. At the top, there are navigation links: Arena (battle), Arena (side-by-side), Direct Chat, Leaderboard, Prompt-to-Leaderboard, Arena Explorer, and About Us. Below the navigation is a banner for "Chatbot Arena (formerly LMSYS): Free AI Chat to Compare & Test Best AI Chatbots". It includes links to Discord, Twitter, 小红书, Blog, GitHub, Paper, Dataset, and Kaggle Competition. A callout box says "New Arena UI at BETA.imarena.ai! Check it out & give feedback!"

**How It Works:**

- Blind Test: Ask any question to two anonymous AI chatbots (ChatGPT, Gemini, Claude, Llama, and more).
- Vote for the Best: Choose the best response. You can keep chatting until you find a winner.
- Play Fair: If AI identity reveals, your vote won't count.
- NEW features: Upload an image and chat. Use Search for online LLMs. Use Text-to-Image models like DALL-E 3, Flux, Ideogram to generate images! Use RepoChat tab to chat with Github repos.

**Chatbot Arena LLM Leaderboard:**

- Backed by over 1,000,000+ community votes, our platform ranks the best LLM and AI chatbots. Explore the top AI models on our LLM leaderboard!

**Chat now!**

Expand to see the descriptions of 102 models

Model A: Can you compute  $3+4-5$ ?  
Okay, let's calculate that:  
1.  $3 + 4 = 7$   
2.  $7 - 5 = 2$   
So,  $3 + 4 - 5 = 2$ .

Model B: Can you compute  $3+4-5$ ?  
 $3 + 4 - 5 = 7 - 5 = 2$   
So,  $3 + 4 - 5 = 2$ .

At the bottom, there are four buttons: A is better, B is better, Tie, and Both are bad.



# Chatbot Arena Leaderboard

Arena (battle) Arena (side-by-side) Direct Chat Leaderboard Prompt-to-Leaderboard Arena Explorer About Us

## 🏆 Chatbot Arena LLM Leaderboard: Community-driven Evaluation for Best LLM and AI chatbots

[Discord](#) | [Twitter](#) | [小红书](#) | [Blog](#) | [GitHub](#) | [Paper](#) | [Dataset](#) | [Kaggle Competition](#)

Chatbot Arena is an open platform for crowdsourced AI benchmarking, developed by researchers at UC Berkeley [SkyLab](#) and [LMArena](#). With over 1,000,000 user votes, the platform ranks best LLM and AI chatbots using the Bradley-Terry model to generate live leaderboards. For technical details, check out our [paper](#).

Chatbot Arena thrives on community engagement — cast your vote to help improve AI evaluation!

New Arena UI at [BETA.lmarena.ai!](#) Check it out & give feedback!

Language Overview Price Analysis WebDev Arena Vision Text-to-Image Copilot Arena Search Arena-Hard-Auto

Total #models: 229. Total #votes: 2,887,373. Last updated: 2025-04-22.

Code to recreate leaderboard tables and plots in this [notebook](#). You can contribute your vote at [lmarena.ai!](#)

Category	Overall	Apply filter	<input type="checkbox"/> Style Control	<input type="checkbox"/> Show Deprecated	Overall Questions	#models: 229 (100%) #votes: 2,887,373 (100%)	
Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	1	<a href="#">Gemini-2.5-Pro-Exp-03-25</a>	1439	+6/-5	10389	Google	Proprietary
2	1	<a href="#">o3-2025-04-16</a>	1418	+14/-9	2211	OpenAI	Proprietary
2	3	<a href="#">ChatGPT-4o-latest (2025-03-26)</a>	1408	+6/-5	9229	OpenAI	Proprietary
3	5	<a href="#">Grok-3-Preview-02-24</a>	1402	+4/-5	14840	xAI	Proprietary
3	5	<a href="#">Gemini-2.5-Flash-Preview-04-17</a>	1393	+10/-7	4073	Google	Proprietary
4	3	<a href="#">GPT-4.5-Preview</a>	1398	+4/-5	15285	OpenAI	Proprietary
7	12	<a href="#">Gemini-2.0-Flash-Thinking-Exp-01-21</a>	1380	+4/-4	26903	Google	Proprietary
7	5	<a href="#">DeepSeek-V3-0324</a>	1373	+6/-7	6792	DeepSeek	MIT
8	5	<a href="#">GPT-4.1-2025-04-14</a>	1363	+10/-9	2927	OpenAI	Proprietary
9	7	<a href="#">DeepSeek-R1</a>	1358	+5/-4	16857	DeepSeek	MIT

<https://lmarena.ai/?leaderboard>

(Checked in 2025-04-28)



# A Commercial Version Yupp AI

The screenshot shows the Yupp AI platform. At the top, there's a vertical sidebar with icons for user profile, message, crown, gear, and other settings. The main header reads "World's smartest AIs, side-by-side with you". Below the header is a search bar with placeholder text "Ask 800+ AIs anything" and buttons for "Choose models" and "Upload". A red "Ask" button with an upward arrow is prominently displayed. In the center, a question box contains the query "Who is famous for painting 'The Starry Night'?", with a small "X" icon to its right. At the bottom left, a notification box says "Claude Sonnet 4.5 available on Yupp! ⚡ X". At the very bottom, a footer note states "By using Yupp, you agree to the [Terms](#) and have read our [Privacy Policy](#)".

- AI startup Yupp has raised \$33 million in a seed funding round led by a16z crypto, founded by a team of veterans from Coinbase, Google, and X.
- Yupp is a free platform that allows users to compare answers from multiple AI models side by side, using crypto rails for rewards and payouts.

<https://yupp.ai/>



# A Commercial Version Yuppy AI

This model is great for this type of prompt

QuickTake  
Stream Multiprocessor manages multiple threads, executing instructions in parallel on GPUs for efficient processing.

This model is great for this type of prompt

Qwen3 VL 235B A22B Instruct

Sure! Here's a brief explanation of a Stream Multiprocessor (SM):

A Stream Multiprocessor (SM) is a key processing unit within NVIDIA's GPU architecture (such as in CUDA-enabled GPUs). It is responsible for executing groups of threads (called warps) in parallel. Each SM contains multiple CUDA cores, shared memory, registers, and other execution resources that allow it to handle many threads concurrently.

Think of an SM as a "mini-processor" inside the GPU — multiple SMs work together to deliver massive parallelism. The number of SMs in a GPU directly impacts its parallel processing power.

For example, in modern architectures like Ampere or Hopper:

- Each SM can schedule and execute multiple warps (typically 32 threads per warp).
- It manages thread blocks, shared memory access, and synchronization primitives.
- Performance scales roughly with the number of SMs and their efficiency.

In short: An SM is the fundamental parallel execution unit in NVIDIA GPUs, enabling high-throughput computing by running many threads simultaneously.

Let me know if you'd like a diagram or code example to illustrate how SMs are used in CUDA!

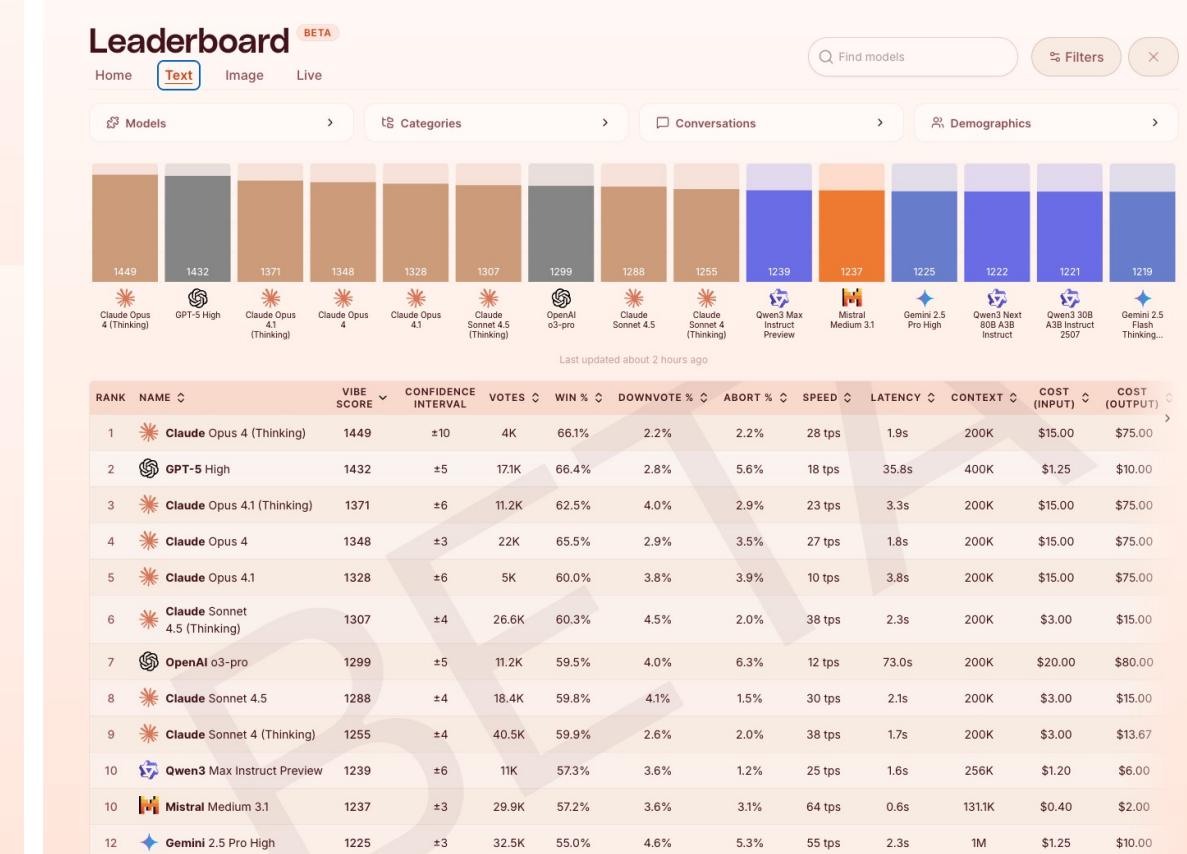
I prefer this

Why was DeepSeek V3.1 Terminus Chat better?  
On target! Fast Style Interesting Nice tone

Why was Qwen3 VL 235B A22B Instruct not as good?  
Wrong Hallucination Slow Too long Not detailed

Write more. Help the AIs get better!

Send feedback





# LLM-as-a-Judge

- Instead of humans, the LLM judge scores each response (for relevance, correctness, etc.), giving a quantitative evaluation.
- This “AI judging AI” approach can quickly evaluate open-ended outputs *at scale*, though care is needed to ensure the scores align with human quality judgments.
- **Pairwise comparison:** The judge LLM is presented with two responses to the same prompt and asked to determine which one is better.
  - Example Prompt: "Given the following two responses to the prompt, which one is more accurate and informative?"
- **Single output scoring (reference-free):** The judge LLM evaluates a single response based on predefined criteria without a reference answer.
  - Example Prompt: "Rate the relevance of the following response on a scale from 1 to 5."
- **Single output scoring (reference-based):** The judge LLM compares a generated response to a reference answer, evaluating similarity or correctness.
  - Example Prompt: "Compare the generated response to the reference answer and rate its accuracy on a scale from 1 to 5."



# LLM-as-a-Judge Example

**Question:** *"What are the health benefits of regular exercise?"*

## Model Responses:

- Model A:** *"Regular exercise improves cardiovascular health, strengthens muscles, enhances flexibility, and boosts mental well-being."*
- Model B:** *"Engaging in regular physical activity can lead to better heart health, improved mood, weight management, and increased energy levels."*

### Prompt of LLM-as-a-Judge:

You are an expert evaluator. Given a question and two responses from different models, determine which response is more helpful and accurate. Provide a brief explanation for your choice.

Question:

*"What are the health benefits of regular exercise?"*

Response from Model A:

*"Regular exercise improves cardiovascular health, strengthens muscles, enhances flexibility, and boosts mental well-being."*

Response from Model B:

*"Engaging in regular physical activity can lead to better heart health, improved mood, weight management, and increased energy levels."*

Which response is better? Please answer with 'Model A' or 'Model B' and provide a short justification.

### LLM-as-a-Judge Response:

Model A. Justification: Both responses are informative, but Model A provides a more comprehensive overview by mentioning flexibility and mental well-being, which are significant benefits of regular exercise.



**RELAXED**  
SYSTEM LAB

# General Purpose Benchmark



# MMLU

- Massive Multitask Language Understanding (MMLU) is a broad knowledge test that covers 57 subjects spanning STEM, humanities, social sciences, etc., from real examinations (college, high school, professional exams) to evaluate a model's breadth of knowledge.
- It was introduced to challenge models beyond simple language tasks.
- Question set: 15,908 multiple-choice questions (4 options each).
- Exact-match evaluation: Each question has one correct answer. Models are typically tested under zero-shot or few-shot settings and evaluated by accuracy (% of questions answered correctly). Only an exactly correct choice counts.

Published as a conference paper at ICLR 2021

**MEASURING MASSIVE MULTITASK LANGUAGE UNDERSTANDING**

Dan Hendrycks  
UC Berkeley      Collin Burns  
Columbia University      Steven Basart  
UChicago      Andy Zou  
UC Berkeley

Mantas Mazeika  
UIUC      Dawn Song  
UC Berkeley      Jacob Steinhardt  
UC Berkeley

**ABSTRACT**

We propose a new test to measure a text model's multitask accuracy. The test covers 57 tasks including elementary mathematics, US history, computer science, law, and more. To attain high accuracy on this test, models must possess extensive world knowledge and problem solving ability. We find that while most recent models have near random-chance accuracy, the very largest GPT-3 model improves over random chance by almost 20 percentage points on average. However, on every one of the 57 tasks, the best models still need substantial improvements before they can reach expert-level accuracy. Models also have lopsided performance and frequently do not know when they are wrong. Worse, they still have near-random accuracy on some socially important subjects such as morality and law. By comprehensively evaluating the breadth and depth of a model's academic and professional understanding, our test can be used to analyze models across many tasks and to identify important shortcomings.

**1 INTRODUCTION**

Natural Language Processing (NLP) models have achieved superhuman performance on a number of recently proposed benchmarks. However, these models are still well below human level performance for language understanding as a whole, suggesting a disconnect between our benchmarks and the actual capabilities of these models. The General Language Understanding Evaluation benchmark (GLUE) (Wang et al., 2018) was introduced in 2018 to evaluate performance on a wide range of NLP tasks, and top models achieved superhuman performance within a year. To address the shortcomings of GLUE, researchers designed the SuperGLUE benchmark with more difficult tasks (Wang et al., 2019). About a year since the release of SuperGLUE, performance is again essentially human-level (Raffel et al., 2019). While these benchmarks evaluate linguistic skills more than overall language understanding, an array of commonsense benchmarks have been proposed to measure basic reasoning and everyday knowledge (Zellers et al., 2019; Huang et al., 2019; Bisk et al., 2019). However, these recent benchmarks have similarly seen rapid progress (Khashabi et al., 2020). Overall, the near human-level performance on these benchmarks suggests that they are not capturing important facets of language understanding.

Transformer models have driven this recent progress by pretraining on massive text corpora, including all of Wikipedia, thousands of books, and numerous websites. These models consequently see extensive information about specialized topics, most of which is not assessed by existing NLP benchmarks. It consequently remains an open question just how capable current language models are at learning and applying knowledge from many domains.

To bridge the gap between the wide-ranging knowledge that models see during pretraining and the existing measures of success, we introduce a new benchmark for assessing models across a diverse set of subjects that humans learn. We design the benchmark to measure knowledge acquired during pretraining by evaluating models exclusively in zero-shot and few-shot settings. This makes the benchmark more challenging and more similar to how we evaluate humans. The benchmark covers 57 subjects across STEM, the humanities, the social sciences, and more. It ranges in difficulty from an elementary level to an advanced professional level, and it tests both world knowledge and problem solving ability. Subjects range from traditional areas, such as mathematics and history, to more



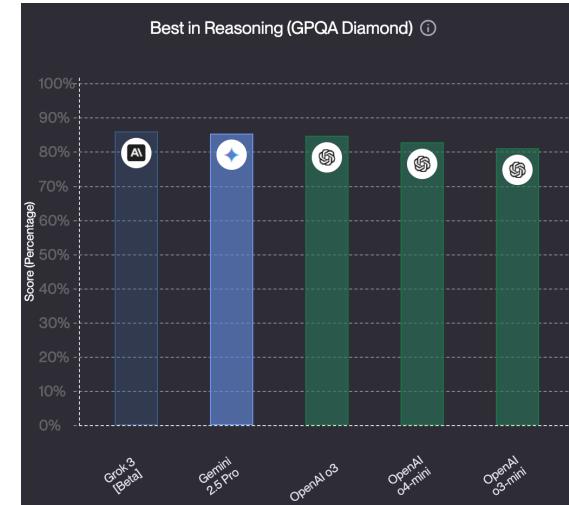
# MMLU Improved Versions

- **MMLU-Pro**: an enhanced benchmark introduced in 2024 to be more challenging and robust than the original MMLU:
  - Add more challenging, reasoning-focused questions;
  - Each question in MMLU-Pro has 10 answer options instead of 4.
- **MMMLU**: Multilingual MMLU, an extension of the MMLU benchmark translated into various languages.
  - MMMLU test set has been professionally translated into 14 languages (Chinese, Arabic, Spanish, etc.)
  - checks if a model's broad knowledge holds up in languages other than English: The translated questions maintain the same content difficulty, just expressed in the target language.
  - Same multiple-choice format (4 options) for each question.



# GPQA

- GPQA “Google-proof” question answer: a benchmark of challenging multiple-choice questions in biology, physics, and chemistry, written at graduate level.
  - Aims to assess advanced reasoning in scientific domains.
  - The questions are designed such that you can’t easily find the exact answer via a quick web search.
  - They often require combining knowledge or understanding an explanation —simulating what a grad student should know, not just copying and pasting from Wikipedia.
- Question set: GPQA consists of 448 multiple-choice questions (with 4 options each) curated by domain experts (e.g., PhD students or professors)
- **GPQA Diamond** is a high-quality subset of the GPQA benchmark, consisting of the most challenging 198 questions.



<https://www.vellum.ai/open-llm-leaderboard>



**RELAXED**  
SYSTEM LAB

# Coding Benchmark



# CodeContests

- CodeContests is a benchmark dataset developed to evaluate the capabilities of large language models (LLMs) in solving competitive programming problems.
- The dataset aggregates problems from multiple competitive programming platforms, where each test includes:
  - Problem Statement: A textual description outlining the programming challenge.
  - Input and Output Specifications: Details on the expected input format and the desired output.
  - Public Test Cases: Sample inputs and outputs to illustrate the problem.
- The primary metric for evaluating model performance on CodeContests is  $\text{pass}@k$ , which measures the percentage of problems solved by at least one of the top-k generated solutions.



# SWE-Bench

- SWE-Bench: a benchmark designed to evaluate the ability of large language models (LLMs) to autonomously resolve real-world software engineering issues.
- Each task provides:
  - Issue description: A textual description of a real-world software issue sourced from GitHub.
  - Codebase access: The complete codebase of the repository where the issue exists.
- Evaluation:
  - Ask the model to return the generated patch, using unix's patch program to the codebase. (similar to your GitHub commit)
  - Execute the unit and system tests associated with the task instance.
  - If the patch applies successfully and all of these tests pass we consider the proposed solution to have successfully resolved the issue.



<https://www.vellum.ai/open-llm-leaderboard>



**RELAXED**  
SYSTEM LAB

# Math Benchmark



# Math-500

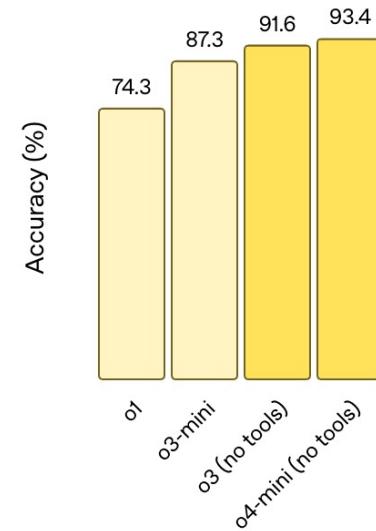
- Mixed Topic Math (MATH-500): Mixed-difficulty math set, a set of 500 math problems compiled to test a range of mathematical reasoning skills.
- These 500 problems probably cover algebra, geometry, number theory, combinatorics, calculus, etc. They serve as a broad, balanced math benchmark.
- Evaluation format:
  - **Input**: Each problem is given in text form. They range from short questions (“Solve for  $x$ : ...”) to paragraph-long word problems. There’s no multiple choice; the model must work it out.
  - **Output**: A simplified exact answer (integer, fraction, expression). The evaluation expects that exact string.
- Evaluation: The metric is Exact Match (EM) — the model’s final answer must exactly match the correct answer (usually a simplified expression or number). This again requires full solution correctness.
  - DeepSeek-V3 achieved **90.2% EM** on MATH-500.



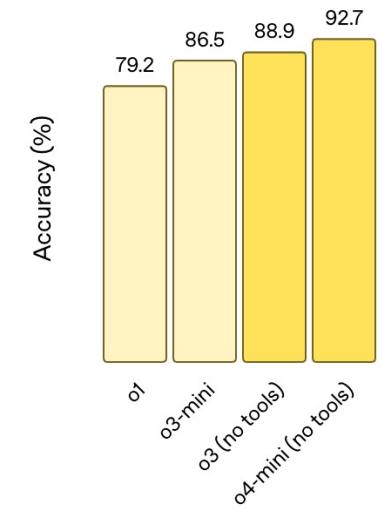
# AIME

- Advanced math challenge: e.g., AIME 2024 refers to problems from the American Invitational Mathematics Examination 2024.
- AIME is a prestigious high school math contest with very challenging problems – harder than typical school exams, requiring creative problem solving.
  - These problems are designed to require reasoning and multiple steps.
  - They often combine clever insights, algebraic manipulation, number theory, or geometry.
- Format:
  - **Input:** Each problem is given as a word problem (often a paragraph with some math setup).
  - **Output:** The model should output a number (0-999).
- Exact match: AIME problems have an integer answer 0-999. The model must carry through the entire solution correctly to get that exact answer.
  - Latest results from OpenAI reasoning models ...

AIME 2024  
Competition Math



AIME 2025  
Competition Math





**RELAXED**  
SYSTEM LAB

# System Benchmark



# LLM API

- LLM API is an important component for you to build your LLM applications.
- **Simplified Integration:** LLM APIs offer a straightforward way to incorporate language understanding into applications. Developers can send text inputs to the API and receive generated responses
- **Scalability:** Using APIs allows applications to scale their language processing capabilities based on demand.

```
python
import openai

# Replace 'your-api-key' with your actual OpenAI API key
openai.api_key = 'your-api-key'

response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Can you explain how the OpenAI API works?"}
    ]
)

print(response['choices'][0]['message']['content'])
```

OpenAI API Call Python Example



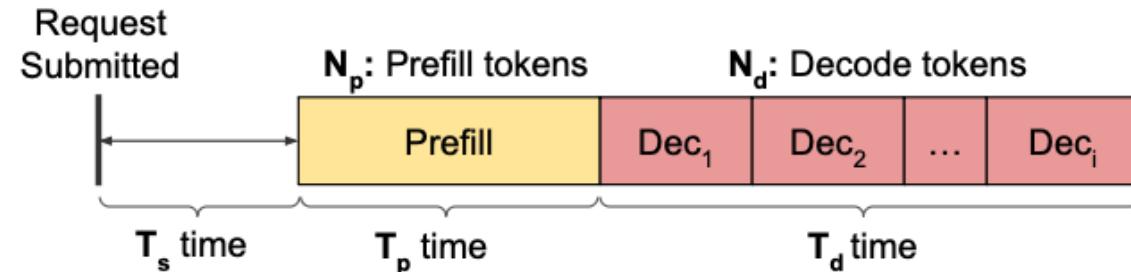
# Evaluating LLM API Service

- When selecting a Large Language Model (LLM) API, besides output quality, it's crucial to consider various system-level factors:
- **Output speed** (tokens per second): This measures how quickly the model generates tokens. Higher output speeds are beneficial for applications requiring rapid responses.
- **Latency** (time to first token): Lower latency ensures faster initial responses, which is critical for real-time applications.
- **End-to-end response time**: This encompasses the total time from sending a request to receiving the complete response. It's a comprehensive metric for assessing user experience.
- **Pricing**: LLM APIs often charge based on the number of input and output tokens processed. It's essential to compare input and output token costs across providers.



# Serving System Evaluation

- $T_s$ : scheduling time;
- $T_p$ : prefill processing time;
- $T_d$ : decode generation time;
- **Time To First Token (TTFT)**: Latency between request arrival and first output token, including scheduling delay and prompt processing time ( $T_s + T_p$ ) . Critical for interactive applications requiring responsiveness.
- **Time to Last Token (TTLT)**: End-to-end latency ( $T_s + T_p + T_d$ ) for complete request processing. Essential for applications like code completion where partial outputs have limited utility.
- **Time Between Tokens (TBT)**: Latency of each subsequent token generation ( $T_d^i$ ) during decoding. Directly impacts perceived model speed, with 6 tokens/second matching typical reading speed.
- **Time Per Output Token (TPOT)**: Average decode time per token ( $T_d/N_d$ ).





# API Leaderboard

API PROVIDER ↑↓	MODEL ↑↓	CONTEXT WINDOW ↑↓	FEATURES ↑↓	MODEL INTELLIGENCE ↑↓	PRICE ↑↓	OUTPUT TOKENS/S ↑↓	LATENCY ↑↓	END-TO-END RESPONSE TIME ↑↓	FURTHER ANALYSIS
			ARTIFICIAL ANALYSIS INTELLIGENCE INDEX ↑↓	BLENDED USD/1M Tokens ↑↓	MEDIAN Tokens/s ↑↓	MEDIAN First Chunk (s) ↑↓	TOTAL Response (s) ↑↓	REASONING Time (s) ↑↓	
OpenAI	o4-mini (high)	200k	70	\$1.93	126.6	36.00	39.94	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Microsoft Azure	o4-mini (high)	200k	70	\$1.93	79.2	65.21	71.52	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Google	G Gemini 2.5 Pro	1m	68	\$3.44	214.6	29.55	31.88	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Microsoft Azure	o3	128k	67	\$17.50	81.0	38.07	44.25	N/A	<a href="#">Model</a> <a href="#">Providers</a>
x1	x1 Grok 3 mini Reasoning (high)	131k	67	\$0.35	202.0	0.38	12.76	9.90	<a href="#">Model</a> <a href="#">Providers</a>
x1	x1 Grok 3 mini Reasoning (high) Fast	131k	67	\$1.45	214.9	0.46	12.10	9.31	<a href="#">Model</a> <a href="#">Providers</a>
OpenAI	o3-mini (high)	200k	66	\$1.93	156.9	46.97	50.16	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Microsoft Azure	o3-mini (high)	200k	66	\$1.93	205.5	40.61	43.05	N/A	<a href="#">Model</a> <a href="#">Providers</a>
OpenAI	o3-mini	200k	63	\$1.93	156.3	14.34	17.54	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Microsoft Azure	o3-mini	200k	63	\$1.93	201.4	12.64	15.13	N/A	<a href="#">Model</a> <a href="#">Providers</a>
OpenAI	o1	200k	62	\$26.25	68.9	50.87	58.13	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Microsoft Azure	o1	200k	62	\$26.25	114.9	26.30	30.65	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Google	G Gemini 2.5 Flash (Reasoning) (AI_Studio)	1m	60	\$0.99	347.5	8.35	9.79	N/A	<a href="#">Model</a> <a href="#">Providers</a>
Lambda	DeepSeek R1	164k	60	\$0.95	38.2	0.55	75.05	61.41	<a href="#">Model</a> <a href="#">Providers</a>
deepseek	DeepSeek R1	64k	60	\$0.96	23.5	3.86	125.12	99.96	<a href="#">Model</a> <a href="#">Providers</a>
Hyperbolic	DeepSeek R1	128k	60	\$2.00	71.5	2.11	41.93	32.83	<a href="#">Model</a> <a href="#">Providers</a>
AWS	DeepSeek R1	128k	60	\$2.36	65.2	0.44	44.08	35.97	<a href="#">Model</a> <a href="#">Providers</a>

<https://artificialanalysis.ai/leaderboards/providers>



# Reference

- <https://docs.mistral.ai/guides/evaluation/>
- <https://medium.com/@saveriomazza/arena-elo-rating-system-5655e16fead5>
- [https://colab.research.google.com/drive/1KdwokPjirkTmpO\\_P1WByFNFiqxWQquwH](https://colab.research.google.com/drive/1KdwokPjirkTmpO_P1WByFNFiqxWQquwH)
- <https://lmarena.ai/?leaderboard>
- <https://www.theblock.co/post/358094/a16zcrypto-33-million-usd-seed-round-yupp-ai-model-evaluation>
- <https://www.evidentlyai.com/llm-guide/llm-as-a-judge>
- <https://en.wikipedia.org/wiki/MMLU>
- <https://arxiv.org/pdf/2406.01574>
- <https://huggingface.co/datasets/openai/MMMLU>
- <https://arxiv.org/pdf/2311.12022>
- <https://arxiv.org/abs/2310.06770>
- <https://huggingface.co/datasets/HuggingFaceH4/MATH-500>
- <https://www.vals.ai/benchmarks/aime-2025-03-11>
- <https://arxiv.org/abs/2507.09019>



**RELAXED**  
SYSTEM LAB

# Course Presentation



# Presentation

- In-class 20-minute presentation (30%), including one target paper, probably by a group of two students:
  - Rubric 1: Clearly organize the material and present the problem definition, motivation, methodology, and evaluation results appropriately. (20%, 18 minutes for presentation)
  - Rubric 2: Can answer the questions from the lecturers and other students appropriately. (5%, 2 minutes for QA)
  - Rubric 3: Submit short feedback for all the other presentation sessions under the same category. (5%)
- Weighted average of the lecture's feedback (30%) and the students' feedback (70%).
- The presentation score (Rubric 1 & 2) would be the same for the two group members. For Rubric 3, you should submit independent feedback for other presentations by yourself.



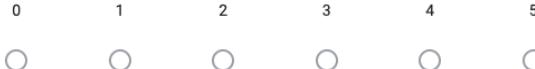
# Feedback Form

- The feedback form will be sent out in the morning before the lecture. To reduce the workload, you only need to submit feedback for presentations **under the same category**; the TA will make sure you receive the invitation to submit the feedback.
- Treat others how you would want to be treated.***

You HKUST email, ended with @connect.ust.hk \*

Short answer text

The talk presents the paper with a clear organization. \*



The talk is well-motivated. – "I found this presentation interesting" \*



The talk is self-explained. – "I learned something interesting from this presentation." \*



The presenter answered the question appropriately. \*





# Deadlines for Presentations

- Presentation slides upload:
  - Send the slides to me and cc the TA.
  - By 9:00 AM on your presentation day.
- Feedback for other groups:
  - Submit the Google form for presentation in that lecture.
  - By 23:59 on that presentation day.





# Swapping the Presentation Slot

- It is totally understandable that you would be traveling during the semester (e.g., participating in a conference), and you might need to swap slots.
- You should contact another group to seek agreement to swap ***by yourself***.
- If you get the agreement, you should send *me and the TAs* an email explaining the reason, cc *your teammates, and the group members of the other group* ***24 hours*** before the presentation session.
  - I will endorse reasonable requests.
- You can first ask the TA about the contact information of the other group members if you need:
  - PENG, You: [ypengbt@connect.ust.hk](mailto:ypengbt@connect.ust.hk)
  - SUN, Yijun: [ysuneb@connect.ust.hk](mailto:ysuneb@connect.ust.hk)



# Category 1. Architecture and Network for LLM

## Topic 1. AI Chip Design.

- Group 1-1
  - Member: Hongyi Wang and Xiangfeng Sun:
  - Paper: *Meta's Second Generation AI Chip: Model-Chip Co-Design and Productionization Experiences*
  - Slot: Session 1-1 (2025/10/21)
- Group 1-2
  - Member: Pengbo Li and Linkai Song
  - Paper: *The Sparsity-Aware LazyGPU Architecture*
  - Slot: Session 1-2 (2025/10/21)



# Category 1. Architecture and Network for LLM

## Topic 2. Heterogeneous Hardware for LLM Services.

- Group 2-1
  - Member: Shiyi Liu and Yiran Xia
  - Paper: [\*H2-LLM: Hardware-Dataflow Co-Exploration for Heterogeneous Hybrid-Bonding-based Low-Batch LLM Inference\*](#)
  - Slot: Session 1-3 (2025/10/21)
- Group 2-2
  - Member: Ruyi Song and Qi Liu
  - Paper: [\*LIA: A Single-GPU LLM Inference Acceleration with Cooperative AMX-Enabled CPU-GPU Computation and CXL Offloading\*](#)
  - Slot: Session 1-4 (2025/10/21)



# Category 1. Architecture and Network for LLM

## Topic 3. Communication Optimizations in LLM Serving.

- Group 3-1
  - Member: Zihao Wang and Zhaoxiang Bao
  - Paper: *MCCS: A Service-based Approach to Collective Communication for Multi-Tenant Cloud*
  - Slot: Session 2-1 (2025/10/23)
- Group 3-2
  - Member: Yichen Liu and Jichen Zhang
  - Paper: *Rdma over ethernet for distributed training at meta scale*
  - Slot: Session 2-2 (2025/10/23)



# Category 2. LLM Training Inference and RL Systems

## Topic 4. Long Context LLM Training.

- Group 4-1
  - Member: Wong Hiu Tung and Ma Sum Yi
  - Paper: *DISTFLASHATTN: Distributed Memory-efficient Attention for Long-context LLMs Training*
  - Slot: Session 2-3 (2025/10/23)
- Group 4-2
  - Member: Dakai An and Tianyu Feng
  - Paper: *ByteScale: Communication-Efficient Scaling of LLM Training with a 2048K Context Length on 16384 GPUs*
  - Slot: Session 2-4 (2025/10/23)



# Category 2. LLM Training Inference and RL Systems

## Topic 5. Mixture of Expert Training.

- Group 5-1
  - Member: Zhenghong Huang and Wenxi Qiu
  - Paper: *Comet: Fine-grained computation-communication overlapping for mixture-of-experts*
  - Slot: Session 3-1 (2025/10/28)
- Group 5-2
  - Member: Wenkai Li and Mengming Li
  - Paper: *Hetermoe: Efficient training of mixture-of-experts models on heterogeneous gpus*
  - Slot: Session 3-2 (2025/10/28)



# Category 2. LLM Training Inference and RL Systems

## Topic 6. Prefill-Decoding Disaggregated Inference.

- Group 6-1
  - Member: Bu Jin and Yu Liu
  - Paper: Mooncake: Trading more storage for less computation—a KVCache-centric architecture for serving LLM chatbot
  - Slot: Session 3-3 (2025/10/28)
- Group 6-2
  - Member: Heyang Sun and Xu Xu
  - Paper: Taming the Chaos: Coordinated Autoscaling for Heterogeneous and Disaggregated LLM Inference
  - Slot: Session 3-4 (2025/10/28)



# Category 2. LLM Training Inference and RL Systems

## Topic 7. Attention-Fully Connected Layer Disaggregated Inference.

- Group 7-1
  - Member: Songrun Xie
  - Paper: [MegaScale-Infer: Efficient Mixture-of-Experts Model Serving with Disaggregated Expert Parallelism](#)
  - Slot: Session 4-1 (2025/10/30)



# Category 2. LLM Training Inference and RL Systems

## Topic 8. RL Systems.

- Group 8-1
  - Member: Yuguang Zhou and Jiazhi Mi
  - Paper: *Hybridflow: A flexible and efficient rlhf framework*
  - Slot: Session 4-2 (2025/10/30)
- Group 8-2
  - Member: Ding Pan and Jiayi Cheng
  - Paper: *AReAL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning*
  - Slot: Session 4-3 (2025/10/30)



# Category 3. Algorithmic Advances for LLM

## Topic 9. Diffusion Language Model.

- Group 9-1
  - Member: Shiyuan Song and Junjie Hou
  - Paper: *Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution*
  - Slot: Session 4-4 (2025/10/30)
- Group 9-2
  - Member: Chaolei Tan and Ziqi Jiang
  - Paper: *Large language diffusion models*
  - Slot: Session 5-1 (2025/11/04)



# Category 3. Algorithmic Advances for LLM

## Topic 10. LLM Inference Acceleration Algorithms.

- Group 10-1
  - Member: Caieus Moreign and Chenyu Liu
  - Paper: *H2o: Heavy-hitter oracle for efficient generative inference of large language models*
  - Slot: Session 5-2 (2025/11/04)
- Group 10-2
  - Member: Haodong Wang and Qianli Liu
  - Paper: *LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding*
  - Slot: Session 5-3 (2025/11/04)
- Group 10-3
  - Member: Jiangnan Yu and Zhenxiao Cao
  - Paper: *Native sparse attention: Hardware-aligned and natively trainable sparse attention*
  - Slot: Session 5-4 (2025/11/04)



# Category 3. Algorithmic Advances for LLM

## Topic 11. Efficient SFT Algorithms.

- Group 11-1
  - Member: Yejia Liu and Haoxian Liu
  - Paper: *GaLore: Memory-Efficient LLM Training by Gradient Low-Rank Projection*
  - Slot: Session 6-1 (2025/11/06)
- Group 11-2
  - Member: Vinayak Khurana and Eman Ansar
  - Paper: *Flora: Low-Rank Adapters Are Secretly Gradient Compressors*
  - Slot: Session 6-2 (2025/11/06)



# Category 3. Algorithmic Advances for LLM

## Topic 12. RL Algorithms for LLM reasoning.

- Group 12-1
  - Member: Runze Zhang and Mingyang Zhao
  - Paper: *Proximal policy optimization algorithms*
  - Slot: Session 6-3 (2025/11/06)
- Group 12-2
  - Member: Haochen Shi and Baixuan Xu
  - Paper: *Deepseekmath: Pushing the limits of mathematical reasoning in open language models*
  - Slot: Session 6-4 (2025/11/06)
- Group 12-3
  - Member: Zibin Meng and Pengfei Wu
  - Paper: *Group sequence policy optimization*
  - Slot: Session 7-1 (2025/11/11)



# Category 4. Multimodal Foundation Model

## Topic 13. Multimodal Information Modeling and Reasoning.

- Group 13-1
  - Member: Pusen Gao and Yiyao Peng
  - Paper: *Qwen2.5-vl technical report*
  - Slot: Session 7-2 (2025/11/11)
- Group 13-2
  - Member: Tse Wai Chung and Tuan An To
  - Paper: *Mmada: Multimodal large diffusion language models*
  - Slot: Session 7-3 (2025/11/11)
- Group 13-3
  - Member: Tianci Yin and Yu Foon Darin Chau
  - Paper: *RoboMonkey: Scaling Test-Time Sampling and Verification for Vision-Language-Action Models*
  - Slot: Session 7-4 (2025/11/11)
- Group 13-4
  - Member: Yubo Zhao and Dongjie Yang
  - Paper: *Qwen2.5-Omni Technical Report*
  - Slot: Session 8-1 (2025/11/13)



# Category 4. Multimodal Foundation Model

## Topic 14. Image- Video- Generation and Acceleration.

- Group 14-1
  - Member: Zhizhou Zhong and Zhenyuan Zhang
  - Paper: *CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer*
  - Slot: Session 8-2 (2025/11/13)
- Group 14-2
  - Member: Gongye Liu and Zixuan Ye
  - Paper: *Seedance 1.0: Exploring the Boundaries of Video Generation Models*
  - Slot: Session 8-3 (2025/11/13)
- Group 14-3
  - Member: Meng Chu and Mingzhe Zheng
  - Paper: *Radial Attention:  $O(n \log n)$  Sparse Attention with*

## *Energy Decay for Long Video Generation*

- Slot: Session 8-4 (2025/11/13)
- Group 14-4
  - Member: Jianxin Huang and Chenran Huang
  - Paper: *Sparse VideoGen2: Accelerate Video Generation with Sparse Attention via Semantic-Aware Permutation*
  - Slot: Session 9-1 (2025/11/18)
- Group 14-5
  - Member: Hanlin Wang and Xuanhua He
  - Paper: *SpurgeAttention: Accurate and Training-free Sparse Attention Accelerating Any Model Inference*
  - Slot: Session 9-2 (2025/11/18)



# Category 5. LLM Agent, Evaluation and Applications

RELAXED  
SYSTEM LAB

## Topic 15. LLM for Coding.

- Group 15-1
  - Member: Zhantong Xue and Le Xu
  - Paper: [Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence](#)
  - Slot: Session 9-3 (2025/11/18)
- Group 15-2
  - Member: Danxuan Liang and Yu Kei Jian
  - Paper: [Qwen2.5-xCoder: Multi-Agent Collaboration for Multilingual Code Instruction Tuning](#)
  - Slot: Session 9-4 (2025/11/18)



# Category 5. LLM Agent, Evaluation and Applications

## Topic 16. LLM for Math.

- Group 16-1
  - Member: Zhaochen Su and Junteng Liu
  - Paper: [\*Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition\*](#)
  - Slot: Session 10-1 (2025/11/20)



# Category 5. LLM Agent, Evaluation and Applications

RELAXED  
SYSTEM LAB

## Topic 17. GUI Agent.

- Group 17-1
  - Member: Yuxuan Cao and Junlong Li
  - Paper: *AgentTrek: Agent Trajectory Synthesis via Guiding Replay with Web Tutorials*
  - Slot: Session 10-2 (2025/11/20)
- Group 17-2
  - Member: Shijue Huang and Changxuan Fan
  - Paper: *Opencua: Open foundations for computer-use agents*
  - Slot: Session 10-3 (2025/11/20)



# Category 5. LLM Agent, Evaluation and Applications

## Topic 18. Retrieval Augmented Generation.

- Group 18-1
  - Member: Bowen Liu and Siqi Wang
  - Paper: [\*Chameleon: A Heterogeneous and Disaggregated Accelerator System for Retrieval-Augmented Language Models\*](#)
  - Slot: Session 10-4 (2025/11/20)
- Group 18-2
  - Member: Chunyin Li and Junle Chen
  - Paper: [\*ReasonIR: Training Retrievers for Reasoning Tasks\*](#)
  - Slot: Session 11-1 (2025/11/25)
- Group 18-3
  - Member: Xiaoyu Han and Kwok Tsun On
  - Paper: [\*Search-r1: Training llms to reason and leverage search engines with reinforcement learning\*](#)
  - Slot: Session 11-2 (2025/11/25)



# Category 5. LLM Agent, Evaluation and Applications

## Topic 19. LLM Evaluation and Benchmarks.

- Group 19-1
  - Member: Ling Liang and Xinyu Geng
  - Paper: *The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models*
  - Slot: Session 11-3 (2025/11/25)
- Group 19-2
  - Member: Yuwei Wu and Longge Deng
  - Paper: *DeepScholar-Bench: A Live Benchmark and Automated Evaluation for Generative Research Synthesis*
  - Slot: Session 11-4 (2025/11/25)



**RELAXED**  
SYSTEM LAB

Thank you!