lecture_12.py

```
from execute_util import text, link, image
from lecture_util import x_link, blog_link
from references import deepseek_r1, llama4, olmo2_32b, mmlu

def main():
```

Evaluation: given a **fixed model**, how "**good**" is it?

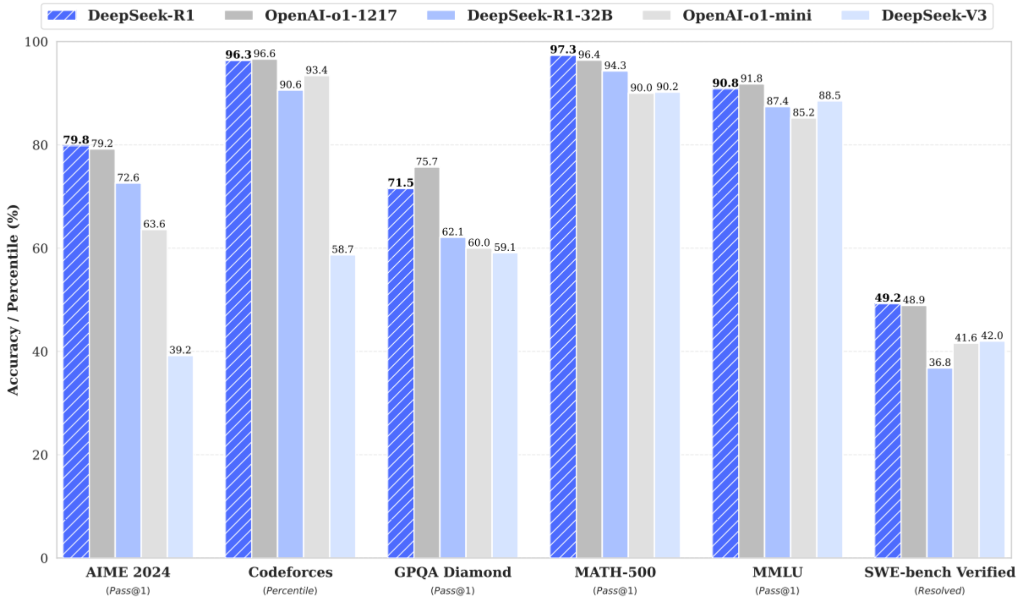
```
    what_you_see()
    how_to_think_about_evaluation()

    perplexity()

    knowledge_benchmarks()
    instruction_following_benchmarks()
    agent_benchmarks()
    pure_reasoning_benchmarks()
    safety_benchmarks()

    realism()
    validity()
    what_are_we_evaluating()
```

Takeaways

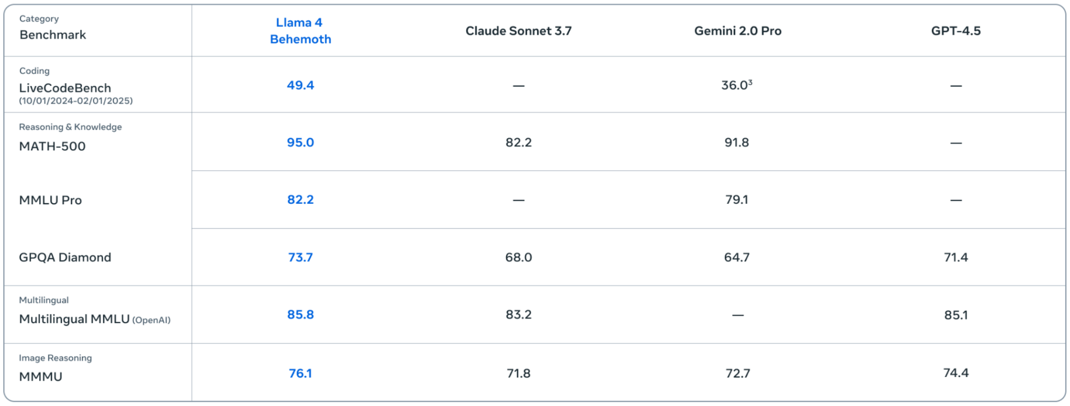
- There is no one true evaluation; choose the evaluation depending on what you're trying to measure.
- Always look at the individual instances and the predictions.
- There are many aspects to consider: capabilities, safety, costs, realism.
- Clearly state the rules of the game (methods versus models/systems).

```
def what_you_see():
```

Benchmark scores

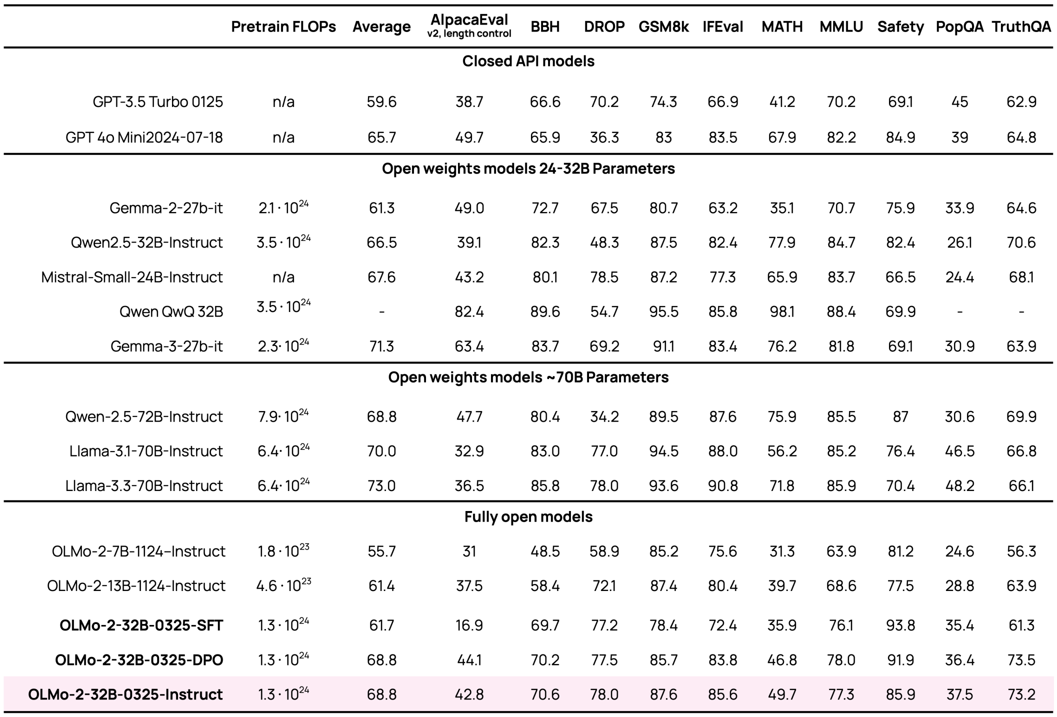


[DeepSeek-AI+ 2025]



Category Benchmark	Llama 4 Behemoth	Claude Sonnet 3.7	Gemini 2.0 Pro	GPT-4.5
Coding LiveCodeBench (10/01/2024-02/01/2025)	**49.4**	—	36.0[3]	—
Reasoning & Knowledge MATH-500	**95.0**	82.2	91.8	—
MMLU Pro	**82.2**	—	79.1	—
GPQA Diamond	**73.7**	68.0	64.7	71.4
Multilingual Multilingual MMLU (OpenAI)	**85.8**	83.2	—	85.1
Image Reasoning MMMU	**76.1**	71.8	72.7	74.4

Llama 4



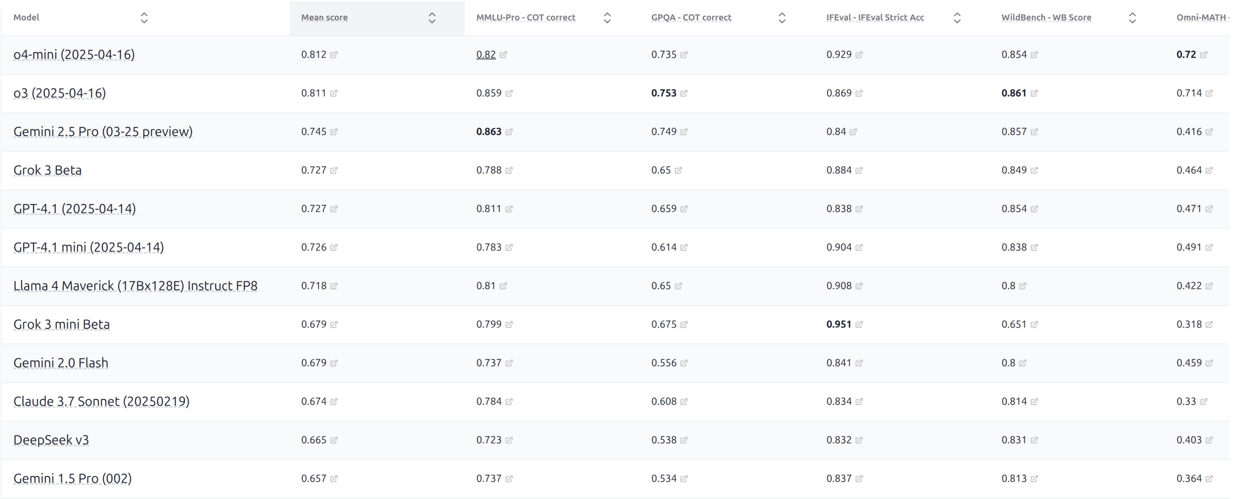
	Pretrain FLOPs	Average	AlpacaEval v2, length control	BBH	DROP	GSM8k	IFEval	MATH	MMLU	Safety	PopQA	TruthQA
Closed API models												
GPT-3.5 Turbo 0125	n/a	59.6	38.7	66.6	70.2	74.3	66.9	41.2	70.2	69.1	45	62.9
GPT 4o Mini2024-07-18	n/a	65.7	49.7	65.9	36.3	83	83.5	67.9	82.2	84.9	39	64.8
Open weights models 24-32B Parameters												
Gemma-2-27b-it	$2.1 \cdot 10^{24}$	61.3	49.0	72.7	67.5	80.7	63.2	35.1	70.7	75.9	33.9	64.6
Qwen2.5-32B-Instruct	$3.5 \cdot 10^{24}$	66.5	39.1	82.3	48.3	87.5	82.4	77.9	84.7	82.4	26.1	70.6
Mistral-Small-24B-Instruct	n/a	67.6	43.2	80.1	78.5	87.2	77.3	65.9	83.7	66.5	24.4	68.1
Qwen QwQ 32B	$3.5 \cdot 10^{24}$	-	82.4	89.6	54.7	95.5	85.8	98.1	88.4	69.9	-	-
Gemma-3-27b-it	$2.3 \cdot 10^{24}$	71.3	63.4	83.7	69.2	91.1	83.4	76.2	81.8	69.1	30.9	63.9
Open weights models ~70B Parameters												
Qwen-2.5-72B-Instruct	$7.9 \cdot 10^{24}$	68.8	47.7	80.4	34.2	89.5	87.6	75.9	85.5	87	30.6	69.9
Llama-3.1-70B-Instruct	$6.4 \cdot 10^{24}$	70.0	32.9	83.0	77.0	94.5	88.0	56.2	85.2	76.4	46.5	66.8
Llama-3.3-70B-Instruct	$6.4 \cdot 10^{24}$	73.0	36.5	85.8	78.0	93.6	90.8	71.8	85.9	70.4	48.2	66.1
Fully open models												
OLMo-2-7B-1124–Instruct	$1.8 \cdot 10^{23}$	55.7	31	48.5	58.9	85.2	75.6	31.3	63.9	81.2	24.6	56.3
OLMo-2-13B-1124-Instruct	$4.6 \cdot 10^{23}$	61.4	37.5	58.4	72.1	87.4	80.4	39.7	68.6	77.5	28.8	63.9
OLMo-2-32B-0325-SFT	$1.3 \cdot 10^{24}$	61.7	16.9	69.7	77.2	78.4	72.4	35.9	76.1	93.8	35.4	61.3
OLMo-2-32B-0325-DPO	$1.3 \cdot 10^{24}$	68.8	44.1	70.2	77.5	85.7	83.8	46.8	78.0	91.9	36.4	73.5
OLMo-2-32B-0325-Instruct	$1.3 \cdot 10^{24}$	68.8	42.8	70.6	78.0	87.6	85.6	49.7	77.3	85.9	37.5	73.2

OLMo 2 (32B)

Recent language models are evaluated on similar, but not entirely identical, benchmarks (MMLU, MATH, etc.).

What are these benchmarks?

What do these numbers mean?



Model	Mean score	MMLU-Pro - COT correct	GPQA - COT correct	IFEval - IFEval Strict Acc	WildBench - WB Score	Omni-MATH -
o4-mini (2025-04-16)	0.812	0.82	0.735	0.929	0.854	**0.72**
o3 (2025-04-16)	0.811	0.859	**0.753**	0.869	**0.861**	0.714
Gemini 2.5 Pro (03-25 preview)	0.745	**0.863**	0.749	0.84	0.857	0.416
Grok 3 Beta	0.727	0.788	0.65	0.884	0.849	0.464
GPT-4.1 (2025-04-14)	0.727	0.811	0.659	0.838	0.854	0.471
GPT-4.1 mini (2025-04-14)	0.726	0.783	0.614	0.904	0.838	0.491
Llama 4 Maverick (17Bx128E) Instruct FP8	0.718	0.81	0.65	0.908	0.8	0.422
Grok 3 mini Beta	0.679	0.799	0.675	**0.951**	0.651	0.318
Gemini 2.0 Flash	0.679	0.737	0.556	0.841	0.8	0.459
Claude 3.7 Sonnet (20250219)	0.674	0.784	0.608	0.834	0.814	0.33
DeepSeek v3	0.665	0.723	0.538	0.832	0.831	0.403
Gemini 1.5 Pro (002)	0.657	0.737	0.534	0.837	0.813	0.364