Democratizing LLMs

An Exploration of Cost-Performance Trade-Offs In Self Refined, Open-Source Models

Introduction

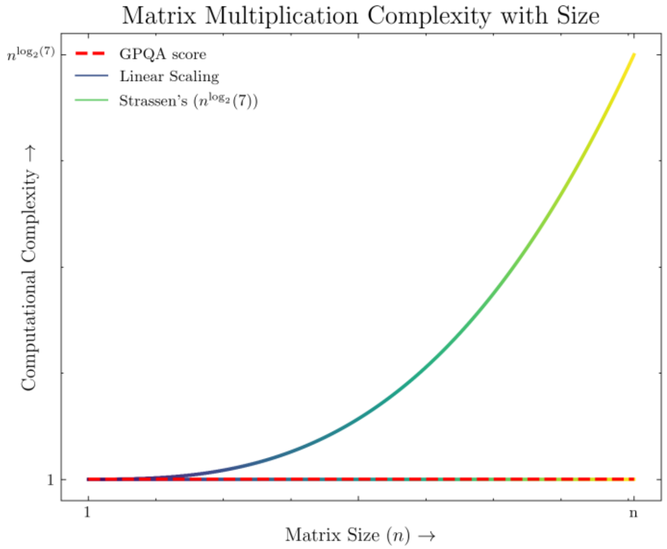
- Large language models have revolutionized and changed the world.
- However, the foundational models are massive and require enormous amounts of compute to train or perform inference on.
- This has led to large companies **pay-walling** access to capable models.
  - For example, GPT-4 is only accessible via API, or subscription, which is $20/month.
  - This represents 0.4% of median US income but 13% of median Indian income. [1]

- **This results in a large portion of the world, price-walled from using today's intelligent models, limiting innovation, especially in crucial domains like fundamental research.**

[1] https://www.statista.com/statistics/802122/india-net-national-income-per-capita/

What solutions exist?

- GPU compute is a finite resource. We need innovative, fresh solutions to make the VRAM, Compute Time Tradeoff effectively.
  - LLM inference is matrix multiplication.
  - Matrix multiplication complexity: $O(n^{2.807})$ via Strassen's algorithm. [1]
  - **<u>Consequence:</u>** Bigger models require exponentially greater amounts of compute, while having sublinear capability increases. (e.g: GPQA)



- **Can we somehow "think longer" with smaller models?**

Our Solution

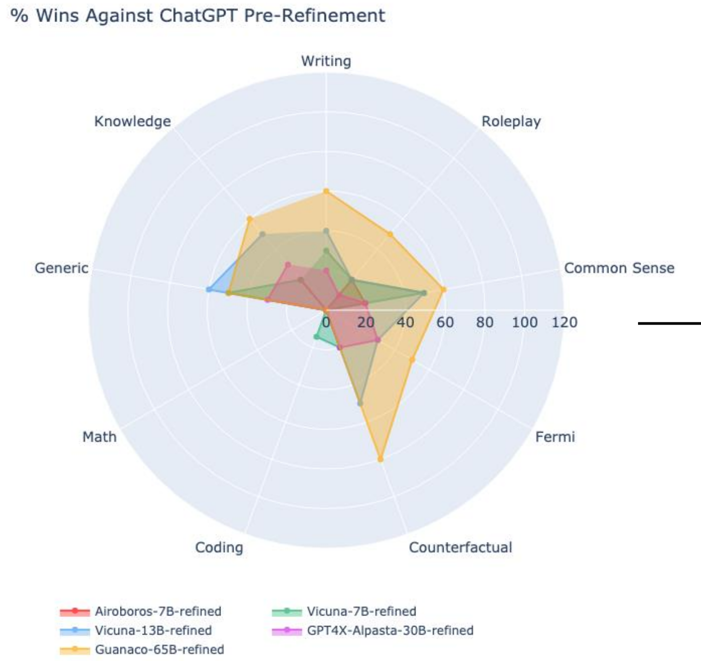
- Domain Agnostic Self Refinement.
  - We use generic critiques to self improve models.
  - We rank various open-source models on the Performance, Refinement and Inference Cost Score **(PeRFICS),** which includes factors like**:**
    - the cost to run inference,
    - total improvement achieved,
    - baseline performance, etc.
- Our results show compute-performance optimality at the ~30B parameter mark.

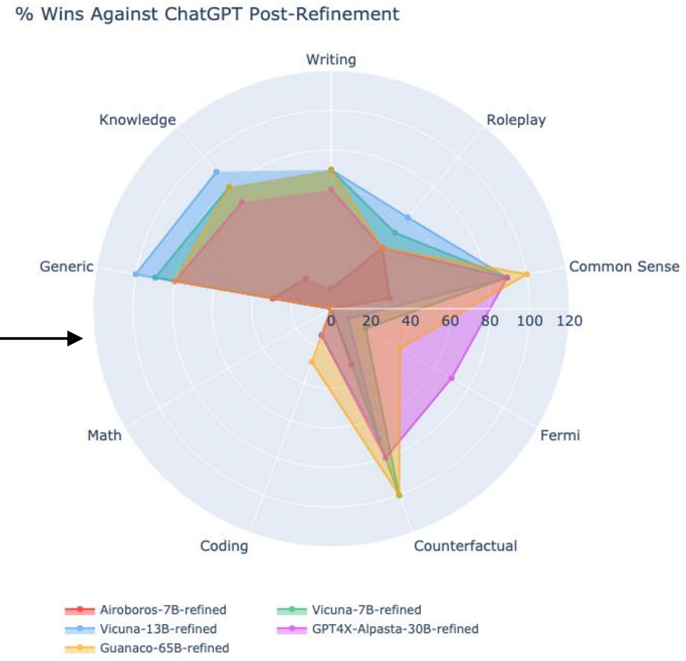
$$y_0 = \mathcal{M}(x_{i,j} \mid \mathcal{I}_{\text{zero}})$$
$$c_0 = \mathcal{M}(x_{i,j}, y_0 \mid \mathcal{I}_{\text{critique}})$$
$$y_1 = \mathcal{M}(x_{i,j}, y_0, c_0 \mid \mathcal{I}_{\text{refiner}})$$

$$\Psi(m) = \frac{\eta \cdot \exp(\kappa \cdot (\alpha \cdot \mathcal{B}(m) + \beta \cdot \mathcal{I}(m))) + \rho \cdot \mathcal{E}(m)}{\exp(\gamma \cdot \mathcal{C}(m)) + \delta} \quad (5)$$

Results

- With our domain-agnostic self-refinement technique, we achieve equivalent or better performance compared to ChatGPT with local, open-source models, by expending more inference compute.
  - This way, we can achieve equivalent performance with lesser upfront hardware investment.





	Airoboros-7B		Vicuna-7B		Vicuna-13B		GPT4X-Alpasta-30B		Guanaco-65B	
Task	Zero Shot	Self Refined	Zero Shot	Self Refined	Zero Shot	Self Refined	Zero Shot	Self Refined	Zero Shot	Self Refined
Writing	89.91%	86.74%	98.11%	104.79%	101.30%	106.80%	94.70%	101.53%	101.98%	104.98%
Roleplay	94.46%	100.12%	94.24%	102.54%	96.86%	105.25%	92.28%	103.88%	100.96%	103.12%
Common-sense	94.75%	93.65%	102.16%	116.48%	99.99%	113.70%	96.32%	107.17%	101.79%	111.46%
Fermi	82.53%	67.27%	76.60%	82.29%	92.50%	85.69%	89.25%	105.55%	94.20%	97.25%
Counterfactual	87.92%	96.45%	92.10%	117.49%	99.23%	112.67%	95.23%	112.14%	111.12%	116.68%
Coding	74.35%	59.72%	69.42%	65.33%	78.57%	78.08%	84.89%	97.79%	81.6%	90.03%
Math	31.67%	23.33%	31.67%	26.67%	26.67%	33.33%	64.81%	56.85%	53.33%	51.67%
Generic	92.88%	92.53%	98.01%	112.66%	101.09%	114.43%	97.09%	100.67%	102.49%	109.65%
Knowledge	85.98%	96.91%	95.20%	108.38%	102.29%	110.70%	97.95%	104.11%	100.24%	106.15%
Mean (Eq Weight)	**81.60%**	**79.64%**	**84.18%**	**92.96%**	**88.72%**	**95.62%**	**90.28%**	**98.85%**	**94.19%**	**98.99%**
Mean (Vicuna)	**86.24%**	**85.31%**	**89.31%**	**99.80%**	**94.53%**	**101.72%**	**92.71%**	**102.57%**	**98.24%**	**103.48%**

Table 3: Single Refinement Scores as a % of ChatGPT Performance.

Conclusion

- We motivate the need for compute-efficient techniques to extract superior performance from smaller parameter open-source models.
- We develop a domain-agnostic self refinement method, and a novel ranking metric, PeRFICS, to score the self-improvement capability of various open-source models. We also demonstrate that that there exists a clear demarcation of compute-optimality for various tasks.
- We provide one of the first demonstrations in academic literature of the use of LLM judges as reliable evaluators.
- We demonstrate that it is indeed possible for compute constrained environments to achieve comparable performance with inexpensive and/or open-source technology.