Reasoning Vision Language Model on Toaster

Backprop Battalion

Aniket Suhas Borkar (210135)
Anuj (210166)
Apoorva Gupta (210179)
Divyansh (210355)
Rajeev Kumar (210815)
Sandeep Nitharwal (210921)

Introduction

What's our goal?

Run VLM on low-compute devices

VLM capability improvement by enabling reasoning

Benchmarking reasoning capability on VQA

Low Compute Device

+

Our work

Advanced Vision capabilities on low compute Devices

Importance of VLMs

Combine visual and textual data to perform multimodal tasks

Image Captioning

"Two young girls are playing with lego toy."

Text-to-Image Generation

"A brain riding a rocketship heading towards moon"

Visual Question Answering (VQA)

Q. Who is wearing glasses?

A. Man

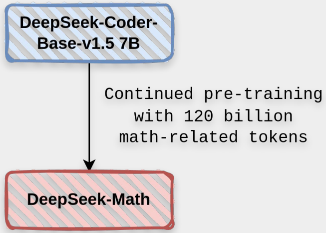
LLM capability improvement: recap

Improve/Fine-Tune

Idea

- Improve architecture
- Scale up parameters
- Fine-tune on specialized datasets

DeepSeek-Math



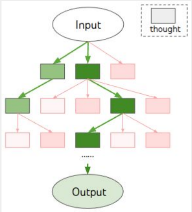
Spend More Time Reasoning

Idea

- Decompose problems into intermediate steps or "thoughts"

Tree of Thoughts (ToT)

- Structure reasoning process into a tree

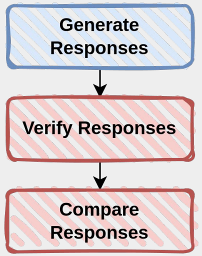


Search Over More Solutions

Idea

- Generate multiple candidate responses and select the best

Inference-Time Search



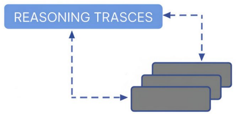
How did we achieve our goal?

Created reasoning data → Fine-tuned understanding → Aligned responses through RL

Dataset

Reasoning Data Distillation

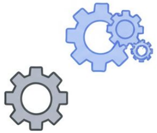
Augmented a part of ChartQA dataset with CoT instances generated using LLMs



SFT

Supervised Fine Tuning for CoT

Using the dataset generated in previous step, the base architecture VLM was fine-tuned to improve CoT predictions



DPO-RL

Reinforcement Learning

Using the fine-tuned model from the SFT phase, multiple responses were generated and paired up to be used for the DPO algorithm

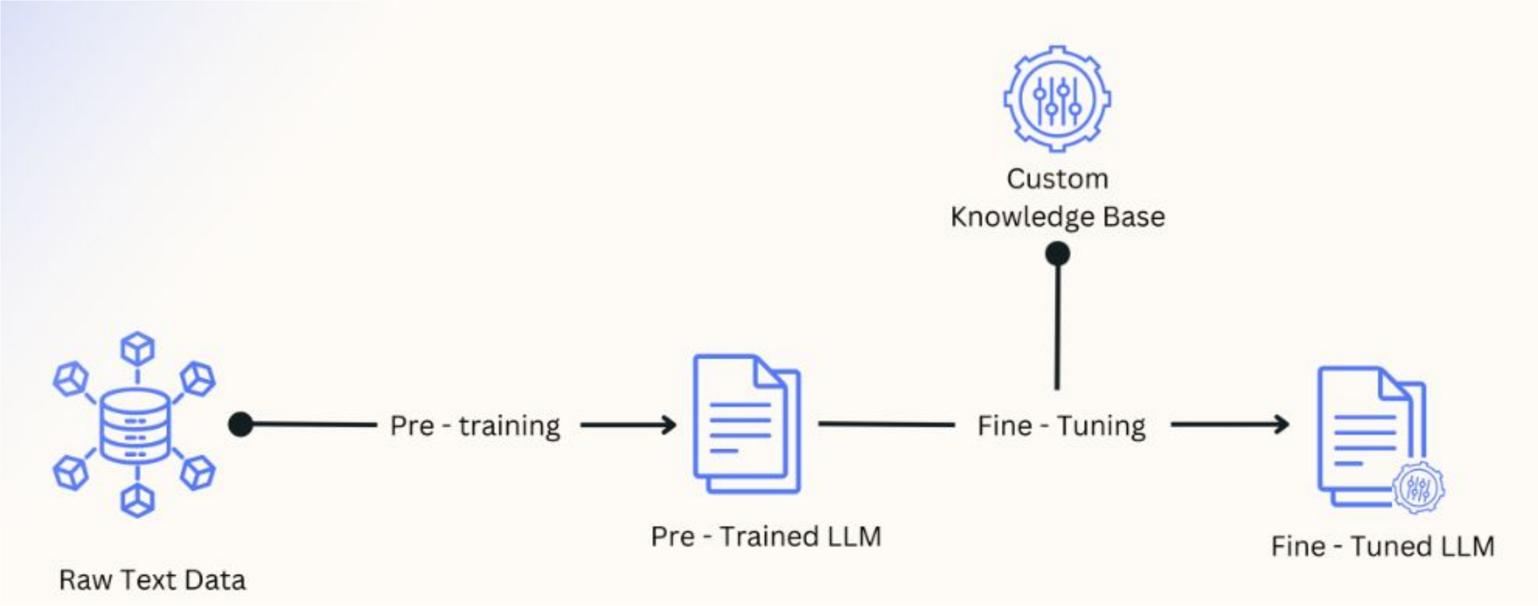
Technical Details

📚 Supervised Fine-Tuning (SFT)

✅ Trains model on (input, target) pairs using reasoning traces

✅ Our setup uses CoT traces from Gemini

✅ Lays the foundation before reinforcement learning (RL) fine-tuning



🧠 RLHF for Vision-Language Models

Why RLHF?

Reinforcement Learning from Human Feedback (RLHF) helps align model behavior with desired reasoning patterns, particularly useful in tasks requiring multi-step reasoning (e.g., **Chain-of-Thought**).

Common Methods Used:

- 🔁 **PPO** (Proximal Policy Optimization): Maximising clipped objective to ensure stable updates
- ⚖️ **DPO** (Direct Preference Optimization): Directly optimizes for human preferences.
- 🧩 **GRPO** (Group Relative Policy Optimization): Extends DPO with flexible reward structures.

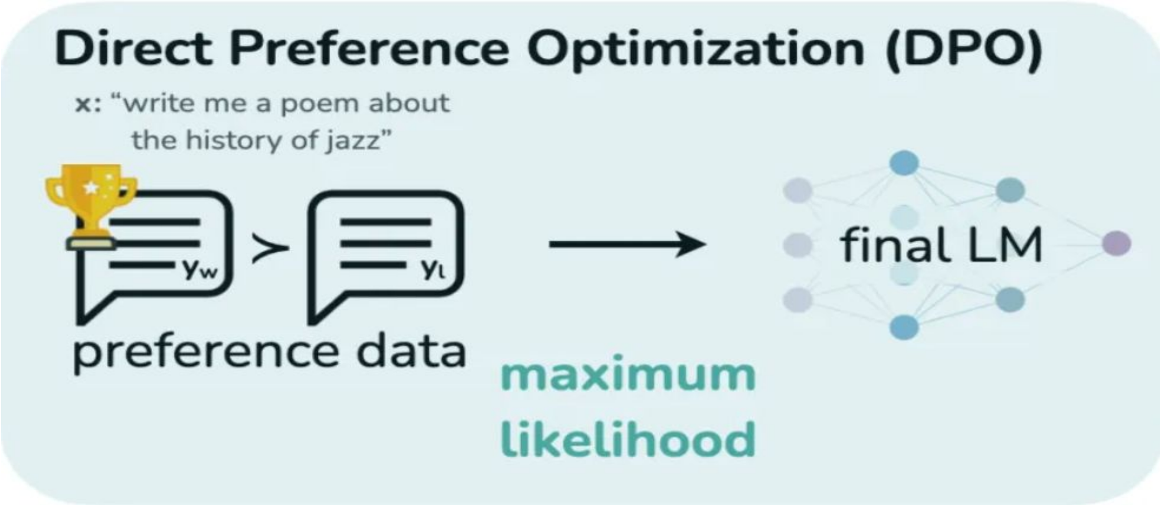
📘 DPO - Direct Preference Optimization

A simple and effective alternative to PPO for aligning models with human preferences

✅ Learns from preference pairs (preferred vs dispreferred outputs)

✅ No reward model needed — avoids reward modeling complexity

✅ Lightweight, scalable, and faster than PPO for alignment



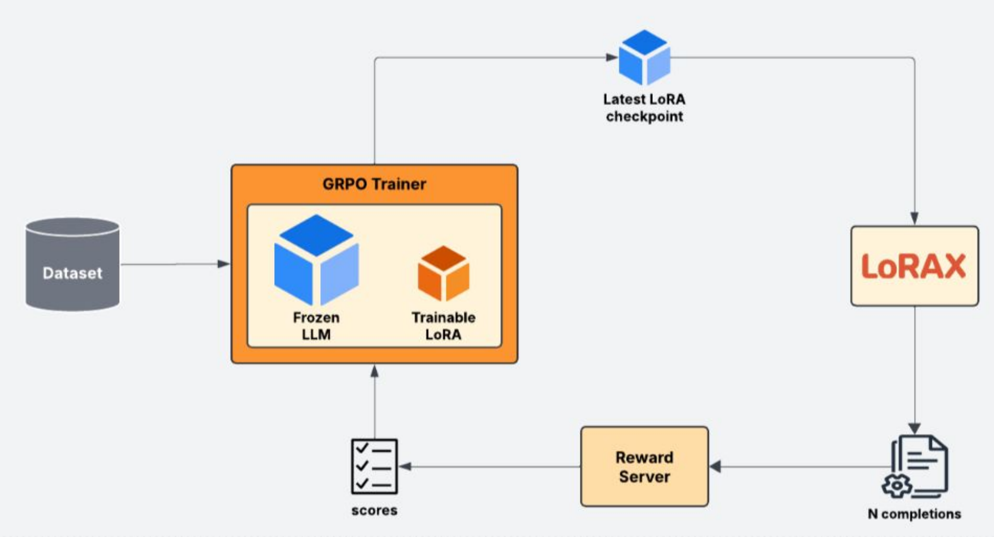
⚙️ GRPO - Group Relative Policy Optimization

A modern and flexible RLHF method enabling fine-grained control through ranked completions

✅ Optimizes model with multiple ranked completions (not just pairwise)

✅ Sample-efficient and avoids reward model training

✅ Suitable for fine-grained reasoning, structured preferences and complex reasoning tasks



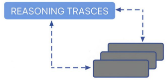
🚀 Our Pipeline

Applying CoT traces and modern RL to small vision-language models.

Big/Highly Capable CoT Models

Vision aided-deepseek-R1
Gemini 2.0 Flash Thinking

Generate CoT reasoning traces



Feed input samples images to get the reasoning traces

Base Vision Language Model

SmolVLM-256M, SmolVLM-500M

Supervised Fine Tuning

We can then fine tune the vision language model on generated reasoning traces.

RL on SFT model

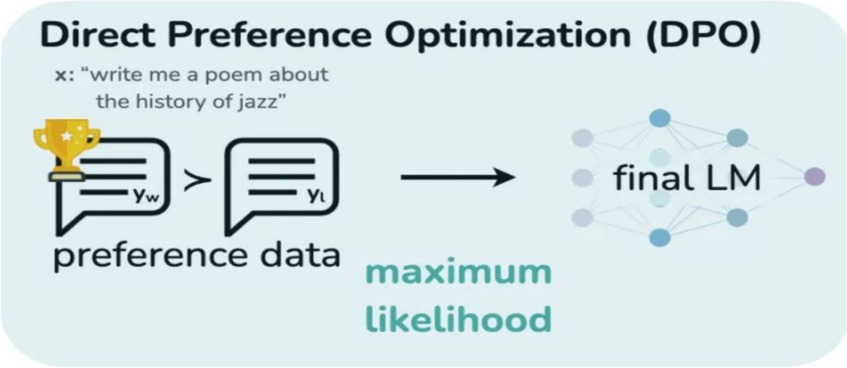
We can also apply RL on the fine-tuned model after SFT to further improve the reasoning capabilities of the model.

Reasoning SmolVLM

Training

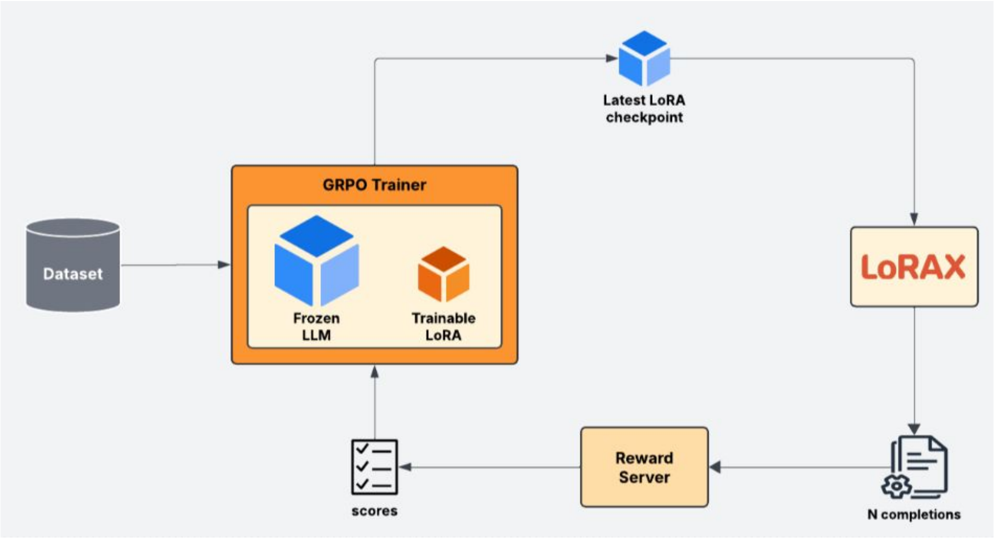
Challenges with DPO training

1. Requires human annotated chosen and rejected answers from a model, for the model to learn preferences
2. Experimented with Gemini's CoT as chosen and current generation as rejected answers
3. Reducing DPO's Memory is an active research area, for us it meant, resource limitation
4. More about it in Experiments Section



Challenges with GRPO training

1. Improvement over DPO in terms of memory, but two many training steps required for the to model to learn anything plausible
2. No plausible implementation to incorporate vision training in popular libraries

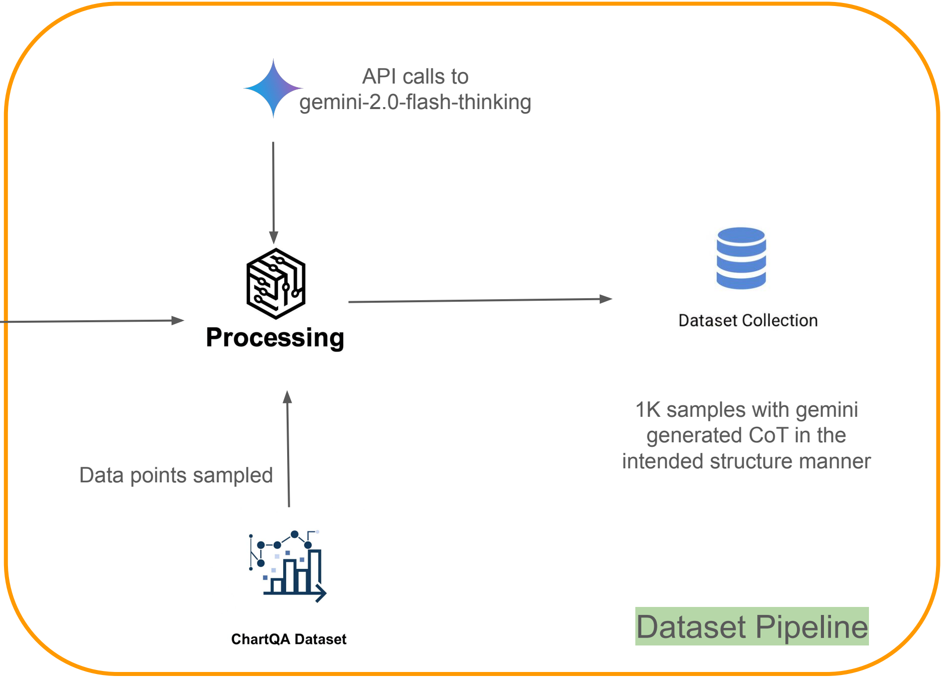


Phase 1: Structured CoT on 1k Samples

Prompted Gemini-2.0-flash-thinking model to generate structured thinking Chain of Thoughts before generating answer for 1k data points sampled from the ChartQA dataset

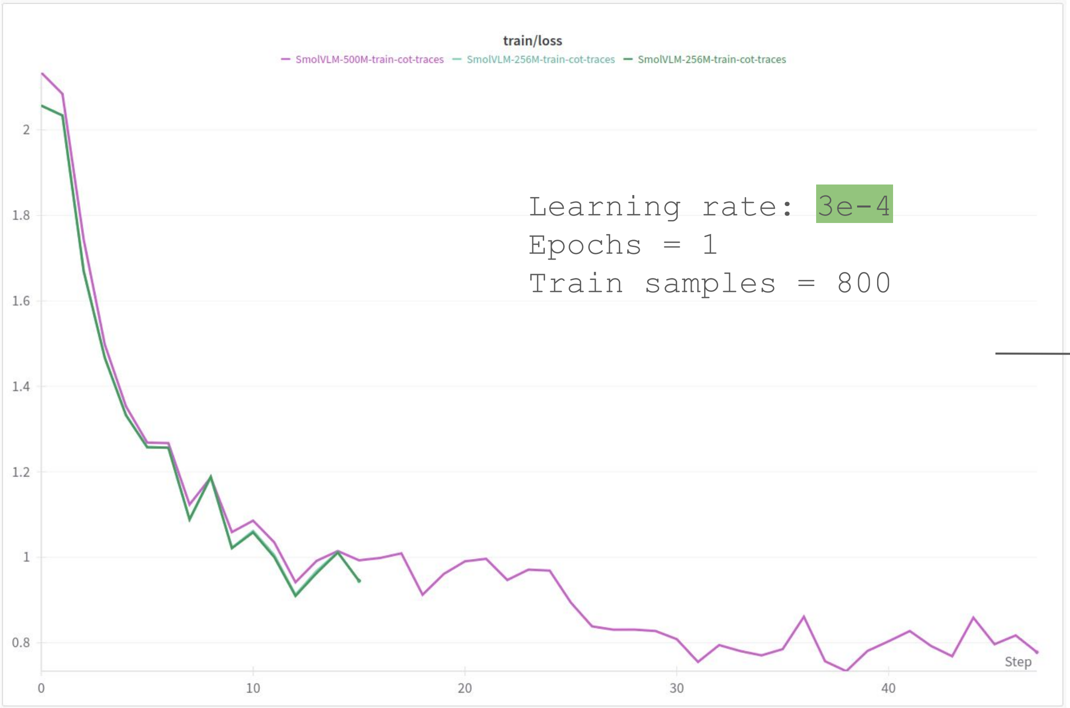
~~~
"""
Put your reasoning process inside <think> tags and the final answer
inside <answer> tags. Follow this format:
```
<think>
**Goal**: Identify the goal of the query
**Requirements**: information from the image needed
**Observation**: put out your observations based on the image
**Conclusion**: Based on the observations, put conclusion
**Recheck**: Recheck the conclusion with the query
**Final Answer**: Put out the final answer
</think>
<answer>
[Your final answer goes here]
</answer>
```
"""
~~~

SYSTEM PROMPT



Phase 1: Continued

Training on the data from phase:1



<think>
Goal: Calculate the difference between…
Requirements: Identify the highest and lowest egg consuming countries from the chart and subtract the lowest from the highest.
Image Observations:
- The chart displays egg consuming countries on the Y-axis.
- The highest egg consuming country is "United Kingdom" with a value of 16.9.
- The lowest egg consuming country is "South Africa" with a value of 1.1.
Conclusion: Subtract the lowest egg consuming country value (1.1) from the highest egg consuming country value (16.9): 16.9 - 1.1 = 15.8.
Recheck: The question asks for the difference in the value of the highest egg consuming country and the lowest egg consuming country. The values are correctly identified and subtracted.
Final Answer: The difference is 15.8.
</think>
<answer>
15.8
</answer>

Model Response

Accuracy decreased from ~ 45% to 17%

Phase 2: LR fix + Scaling Data?

Even though both the eval loss and train loss were flowing nicely, it was hard to figure out what is going wrong.

Ilya Sutskever
Founder SSI,
Founder & Chief Scientist OpenAI

“I had a very strong belief that bigger is better”

“You need to just scale it and it will work fine”

Incorporating

Dataset Pipeline

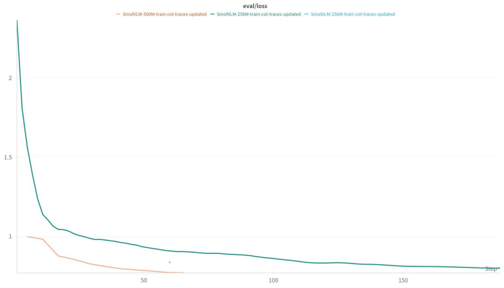
ChartQA Dataset

Processing

Generate CoT on 4k samples, after 12 hours of calls

+

LR changed to 3e-5



Still, the same problem

Phase 2: Towards Simple CoT

- We tried to figure out what went wrong
- Two main reasons for the failure could have been:
  - Structure Learning is hard task
  - Longer generations leads to squashing of last of the tokens in transformers

````
SYSTEM_PROMPT = """
Put your reasoning process inside <think> tags in or
two sentence and the final answer inside <answer>
tags. Follow this format:
```
<think>
[Your Reasoning process goes here]
</think>
<answer>
[Your final answer goes here]
</answer>
```
"""
````

Dataset Pipeline

Dataset Collection

3k smolcot dataset after 4 hours

"**output**": "75 / 10 = 7.5\nThe ratio is 7.5."

"**output**": "224 + 22 = 246."

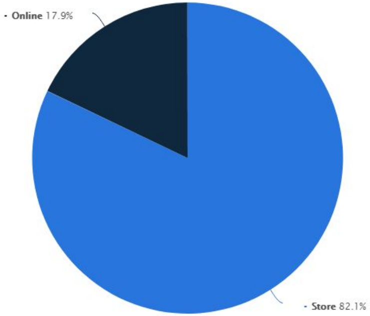
“**Output**”: "25.3% + 2.5% = 2.8%\nThe sum is 2.8%."

Phase3: Massively Scaled Simple CoT Data

What if we scale the CoT data even further

Dataset Pipeline

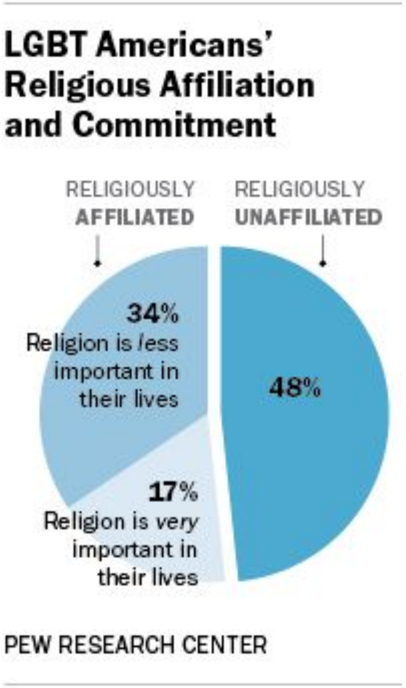
Generated ~8k samples



e stores higher than Online?"

Ilya Sutskever
Founder SSI,
Founder & Chief Scientist OpenAI

Example



How many parts is the graph divided into?

500M Base: The survey also found that 48% of LGBT Americans say religion is very important in their lives, while 17% say it is not important at all.

500M-3k: 3

500M-7k: The graph has three parts: Religiously Affiliated, Unaffiliated, and Religiously Unaffiliated. Answer: 3

Experiments

Experimentation we did

Large Chain-of-Thought traces

- We originally started out with training data consisting of long Chain-of-thought traces
- The CoTs were generated using Google **Gemini**, which is a very large model with large context
- SFT with this data allowed the model to successfully **learn the answer format**
- Also teaches model to reason - correctly captures **“goal”** and “**plan**” of reason
- But this means the model generates a large number of tokens in this process of reasoning
- But SmolVLM fails to capture such a **long context**, tends to hallucinate
- Results were not always satisfactory

Experimentation we did

DPO on these CoTs

- DPO is primarily a method to introduce a **“preference signal”** and align outputs to the same
- It is **difficult** to inculcate “totally new” behaviour (e.g. reasoning) and DPO only yields marginal changes to the original model’s ability
- Since the base model is in essence **too small** to do long chain-of-thoughts, DPO doesn’t help much
- As noticed in the previous case, the model loses track of the context and **hallucinates**

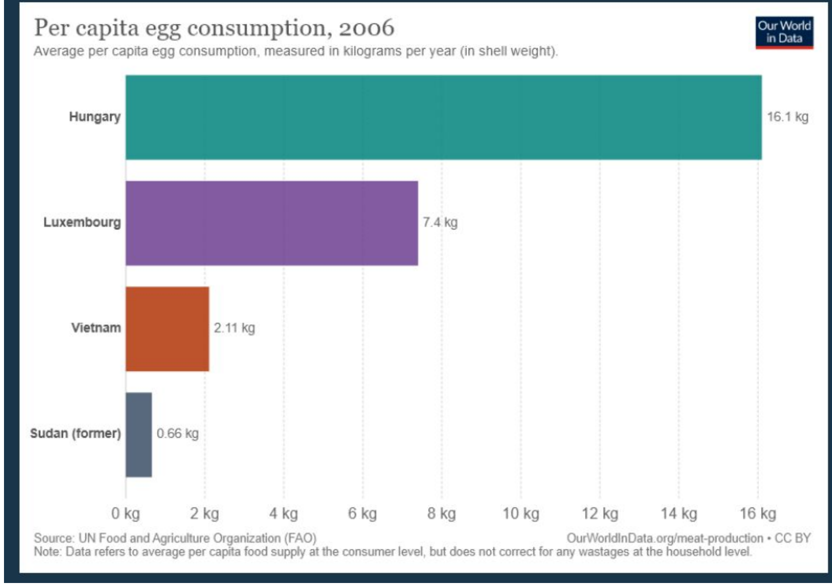
- Thus, we moved on to modifying our data pipeline and generated smaller reasoning steps
- More concise reasoning allowed SmolVLM to capture context and input info better

Example: Issues with long CoT

```
What is the difference in the value of highest egg consuming country and lowest egg consuming country?
<think>
The user is asking for the difference between the highest and lowest egg consumption values among the countries listed in the bar chart.
I need to identify the highest and lowest values and subtract them to get the difference.
Highest value is 16.1 kg (Hungary) and the lowest is 0.66 kg (Sudan (former)).
Subtracting the lowest from the highest value: 16.1 - 0.66 = 15.44.
</think>

<answer>
15.44
</answer>
```

Gemini CoT



smolvlm-smolcot-sft.ipynb | smolvlm-smolcot-sft.ipynb (output)

```
----------------
BASE Model:
----------------
User:What is the difference in the value of highest egg consuming country and lowest egg consuming country?
Assistant: 15.
==========================
----------------
SFT Model:
----------------
<think>
- **Goal**: Calculate the difference between the highest egg consuming country and the lowest egg consuming country from the chart.
- **Requirements**: Identify the highest and lowest egg consuming countries from the chart and subtract the lowest from the highest.
- **Image Observations**:
    - The chart displays egg consuming countries on the Y-axis.
    - The highest egg consuming country is "United Kingdom" with a value of 16.9.
    - The lowest egg consuming country is "South Africa" with a value of 1.1.
- **Conclusion**: Subtract the lowest egg consuming country value (1.1) from the highest egg consuming country value (16.9): 16.9 - 1.1 = 15.8.
- **Recheck**: The question asks for the difference in the value of the highest egg consuming country and the lowest egg consuming country. The values are correctly identified and subtracte
- **Final Answer**: The difference is 15.8.
</think>

<answer>
15.8
</answer>
==========================
----------------
DPO Model:
----------------
<think>
- **Goal**: Calculate the difference between the highest egg consuming country and the lowest egg consuming country from the bar chart.
- **Requirements**: Identify the highest and lowest egg consuming countries from the chart and calculate their difference.
- **Image Observations**:
    - The chart displays egg consumption in grams for different countries.
    - The highest egg consuming country is "Chad" with 16.99 grams.
    - The lowest egg consuming country is "Nigerian" with 1.18 grams.
- **Conclusion**: Subtract the lowest egg consuming country value from the highest egg consuming country value: 16.99 - 1.18 = 15.81.
- **Recheck**: The question asks for the difference in egg consuming country values. 15.81 is the difference between the highest and lowest egg consuming country values.
- **Final Answer**: The difference is 15.81.
</think>

<think>
15.81
</think>
```

Base Model lacks reasoning capacity!

Wrong Numbers

Arbitrary Country Names

Tag format is however now learned!

Experimentation we did

Adding “Let’s think step by step” prompt

- Large Language Models are Zero-Shot Reasoners : Takeshi et.al. demonstrate that simple additions to prompts improve results
- LLMs are efficient “zero-shot reasoners”: **reasoning performance improves** based on prompts
- We try modifying the system prompt and adding the simple instructions at the end
- E.g. “First think and then answer”, “Start your answer with thinking …”, etc.
- In particular: we append **“Let’s think step by step”** to the user prompt
- This significantly boosts the performance of the models, totally out of the box

Effect of “Let’s think step by step”: reasoning promoted

```
[26]  ✓ 10.9s
...
----------------
BASE Model:
----------------
User:Do men or women have more COVID number of of cornavirus patients in intensive care in Sweden?
Assistant: Men had a higher number of COVID-19 patients in intensive care than women in Sweden. In total, 4,272 men and 3,099 women had COVID-19 patients in intensive care in Sweden in March 2020.
==========================
----------------
SFT Model:
----------------
Men.
==========================
```

Without prompt modification: direct answer

```
----------------
BASE Model:
----------------
User:Do men or women have more COVID number of of cornavirus patients in intensive care in Sweden?Let's think step by step
Assistant: The number of patients in intensive care in Sweden was higher for men than for women. In total, 4,272 men and 3,099 women were in intensive care.
==========================
----------------
SFT Model:
----------------
The bar chart shows the number of COVID patients in intensive care in Sweden for men and women.

For men, the number of COVID patients in intensive care is 3,099.

For women, the number of COVID patients in intensive care is 1,216.

Comparing the numbers, we can see that men have more COVID patients in intensive care than women.
==========================
```

Modified Prompt

Increased Reasoning expressed!

Experimentation we did

Adding “Let’s think step-by-step” to query prompt.

Motivation: Promote reasoning behaviour.

Results: Model generates longer think tags, improves model’s accuracy

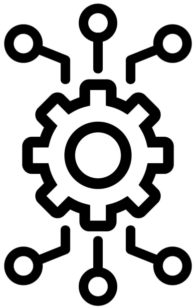
Example:

Evaluation Pipeline

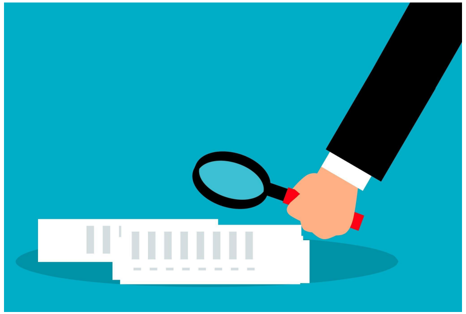
Evaluation Methodology

It is hard to define a completely objective metric like accuracy in case of VLM since the model outputs natural language text.

Therefore, we evaluated the model in two ways:



Automated Evaluation



Human Evaluation

Automated Evaluation

- We wrote a Python script to match the model's answer with the actual answer.
- Uses a combination of approaches to determine whether the model's output contains the correct answer:
  - Use Regex to extract numbers, symbols and other words.
  - Ignore punctuations and helping words and converted both the texts to lowercase.
  - Use **word2num** library for comparing outputs like "7" and "seven".
  - Refrained from using other LLMs to evaluate the results due to efficiency concerns.
- Major Challenges that require human evaluation:
  - Comparing synonyms ("**The statement is correct**" and "**The statement is True**" means the same).
  - Comparing unsolved fractions to floating points **(like 3/2 to 1.5)**.
  - Comparing different form of words like "**increases**" and "**increasing**"
  - Not ideal for comparing which response is better in cases when both models give correct or both give incorrect answers.

Human Evaluation

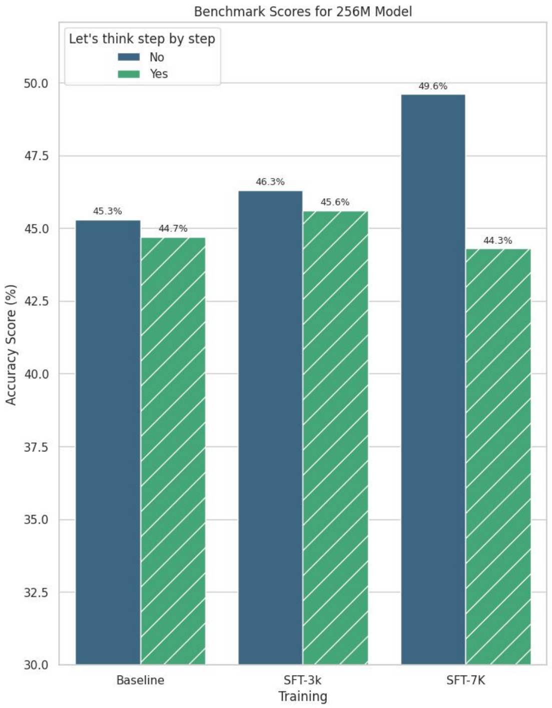
- Human evaluation tackles the major problems faced by automated evaluation.
- We ensured there is no bias in human evaluation by:
  - Anonymizing the model's responses when asking for comparison (user doesn't get to know which is ours and which one is baseline)
  - Asked friends to evaluate the responses.
- Made a script that shows the user the query and the image with two model responses, user chooses the better response or claims both are equivalent.
- This series of human preferences is then used to find the relative performance of various models using the well-known ELO rating system.
- **ELO rating system**: A relative rating system that is used in sports like chess to compare the strength of players, ELO ratings are calculated based on the matches the players played between each other.

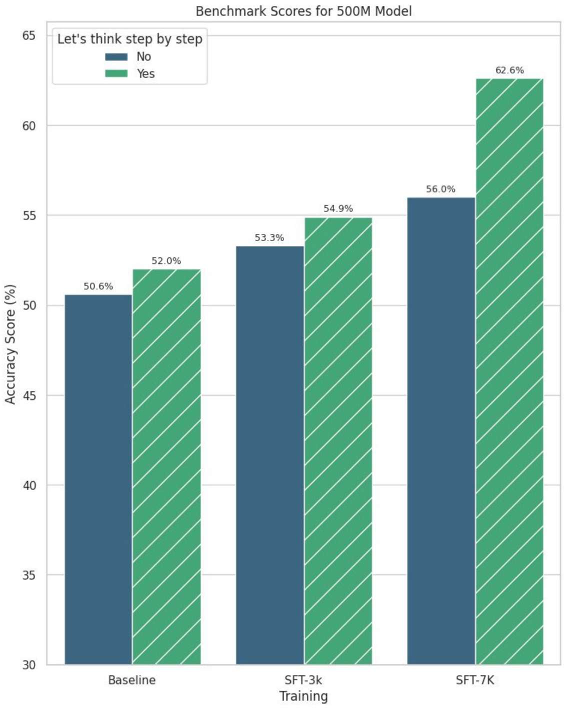
🏆 **Chatbot Arena LLM Leaderboard: Community-driven Evaluation for Best LLM and AI chatbots**

Discord | Twitter | 小红书 | Blog | GitHub | Paper | Dataset | Kaggle Competition

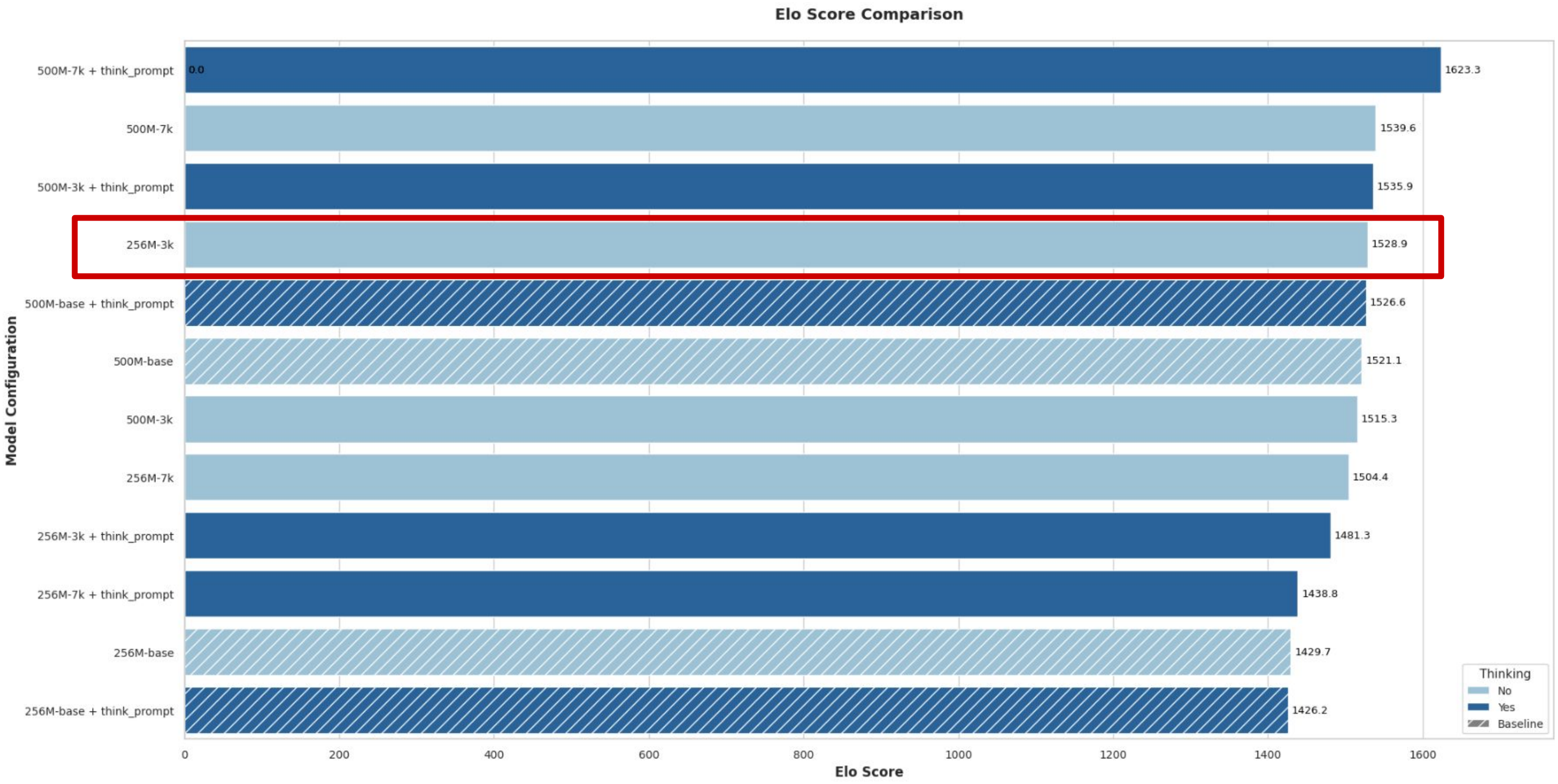
Results

Automatic Evaluation





Human Evaluation



What’s up with 256M-3k

- Interesting fights of 256M-3k with other models
- We offer no explanation why is it performing so good

GLU Variants Improve Transformer

Noam Shazeer
Google
noam@google.com

February 14, 2020

Abstract

Gated Linear Units [Dauphin et al., 2016] consist of the component-wise product of two linear projections, one of which is first passed through a sigmoid function. Variations on GLU are possible, using

Noam Shazeer
Gemini Lead @Google

4 Conclusions

We have extended the GLU family of layers and proposed their use in Transformer. In a transfer-learning setup, the new variants seem to produce better perplexities for the de-noising objective used in pre-training, as well as better results on many downstream language-understanding tasks. These architectures are simple to implement, and have no apparent computational drawbacks. We offer no explanation as to why these architectures seem to work; we attribute their success, as all else, to divine benevolence.

Conclusion

Cost Estimation

GPU Usage

```
A40 @ $0.4/hour
Total: $35.2
```

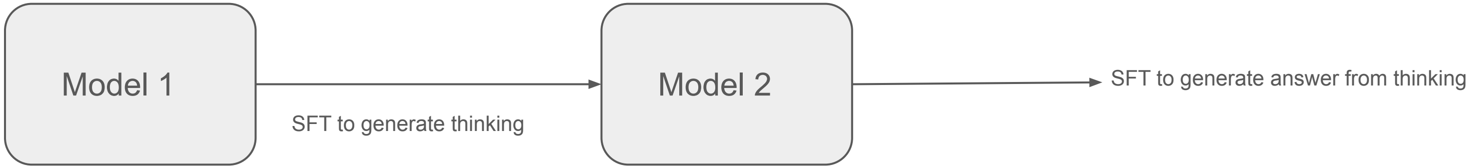
Dataset Pipeline

API Calls

```
$0.1/$0.4 per million
input/output tokens
Total: $20.48
```

Future Improvements

Employ Multistage training



Employ Multistage training

Group Contribution

All the members are equally contributing in all the project work and discussions.

Member Name	Roll No.	Contributions
Aniket Suhas Borkar	210135	- DPO, Training, Experiments
Anuj	210166	- Eval, Data, Experiments
Apoorva Gupta	210179	- GRPO, Training, Experiments
Divyansh	210355	- SFT, Data, Experiments
Rajeev Kumar	210815	- Data, Eval, Experiments
Sandeep Nitharwal	210921	- Eval, Data, Experiments

Demo

Thank You!