Okay, so these are the basic steps. Now draft a 600-700 word detailed methodology based on these summarized steps

* We started out by downloading these models locally from ollama. Despite our initial attempts to run Gemma 12 billion and Deepseek 14 billion locally, we found that we had several computational limitations and that the code constantly crashed or stopped, which is why we had to reduce our largest parameter count to the 7- 8 billion range.
* We initially tried to use the OpenAI and Claude API as controls however due to token restrictions and pricing limitations we had to abandon the same
* We initially tested our code on a few hundred basic questions such as SAT questions and basic arithmetic questions,
* We observed that there was no discernible difference between each of these tasks, with most models having accuracies in the range of 90%.
* This prompted us to move on to harder tasks, such as the big bench hard tasks where we selected 10 of the 23 tasks due to computational restrictions.
* Wrote a couple of scripts to ested 50-250 prompts for each task across all 6 models based on the resources available to every member and store these outputs
* Manually annotated each question, finding the accuracy of each model across all these tasks and gaining additional quantitative insights into the reasoning capabilities of each models across the big bench tasks

We began by downloading the models directly from Ollama’s supported registry, which enabled us to run these distilled models locally, providing us with the flexibility to create scripts to automate the prompting process. Initially, we aimed to test the larger 12B parameter version of Gemma and the 14B version of Deepseek-R1 as well. However, we quickly encountered significant computational limitations. On our local hardware, models in the 12–14B range caused frequent crashes or complete execution failures, likely due to memory overflow and thread bottlenecks in multi-task environments.

As a result, we reduced our maximum model size to 7–8 billion parameters. Specifically, we used:

* **Deepseek-R1**: 1.5B and 8B
* **Gemma3**: 1B and 4B
* **LLaMA 3.2**: 1B
* **LLaMA 3.1**: 8B

We initially intended to include proprietary models like **OpenAI’s GPT-4** and **Anthropic’s Claude** via their respective APIs to serve as external baselines. However, due to stringent token limits, rate caps, and prohibitive pricing, we were unable to include them in our final analysis. This decision emphasized the need for evaluating open-source models that can be freely deployed and assessed at scale in local settings. Ollama's lightweight runtime allowed for quick model switching, unified environment control, and streamlined prompt execution, making it a suitable platform for local testing of large language models.

Our early experiments involved manually prompting a few hundred prompts drawn from simpler tasks such as high-school-level SAT verbal and math questions, along with basic arithmetic and logical reasoning problems. Interestingly, across all six models, accuracy scores for these easier tasks were consistently high, ranging between 85% to 95%. The performance variation between smaller and larger models on these tasks was negligible, prompting us to pivot toward more challenging evaluation domains. Additionally, the time-intensive nature of manually prompting each model prompted us to automate the process using custom scripts, significantly improving efficiency and consistency across experiments.

To better differentiate the capabilities of the models and probe their reasoning depth, we selected **10 tasks from the BIG-Bench Hard (BBH)** suite. BBH tasks are specifically designed to test the limits of reasoning, logic, and language comprehension. Due to computational constraints and time limitations, we opted not to run all 23 BBH tasks, instead selecting a representative and diverse subset spanning logical deduction, multi-step reasoning, and deceptive prompt scenarios.

For each task, we used **50 to 250 prompts**, depending on the computational resources available to each team member. Python scripts were developed to automate prompt execution, dispatch model queries in parallel batches, and log responses in a structured format for later analysis.

After all model outputs were collected, we conducted **manual annotation** to determine the correctness of each response. This involved cross-referencing model answers with known correct outputs or reference solutions. While some tasks allowed for programmatic accuracy checking (e.g., arithmetic problems), most required human evaluation due to the nuanced nature of reasoning and multi-part answers.

Each task’s result set was analyzed to compute:

* **Accuracy** (correct responses / total responses)
* **Qualitative performance trends** (e.g., coherence, step-by-step reasoning)
* **Performance across model scale**

These annotations allowed us to derive both **quantitative accuracy scores** and **qualitative insights** into how each model handled complex reasoning under reduced parameter budgets.

Task

Counting of objects:

In this task, each model was given a prompt where it had to count the total number of objects or count a type of object listed in the prompt.

Eg: I have a yam, a cauliflower, a bed, two cabbages, a garlic, an oven, a carrot, a head of broccoli, a potato, a stalk of celery, a lettuce head, and a toaster. How many vegetables do I have?

For such prompts, each model had to interpret each question, discern the type of each object, and count it correctly; the accuracy for each model can be found as follows.

|  |  |  |
| --- | --- | --- |
| Model | Parameters | Accuracy |
| Deepseek | 1.5 billion | 50.4% |
| Deepseek | 8 billion | 67.6% |
| Gemma | 1 billion | 60.8% |
| Gemma | 4 billion | 73.6% |
| Llama 3.1 | 8 billion | 46.4% |
| Llama 3.2 | 1 billion | 40% |

While there seems to be a correlation between the model size and accuracy, when analyzing the reasoning capabilities of these models, some valuable qualitative insights were obtained.

* The most common type of error across all models was forgetting certain objects in the prompt, improper parsing, not properly counting every object in the prompt, or incorrectly categorizing an object
* Gemma clearly seems to be the most adept at solving this type of problem, another valuable insight is that approximately 70% of the errors by Gemma models had simple arithmetic errors in the final addition after correctly parsing through the entire prompt and keeping track of each object in the prompt suggesting strong analyzing and retention properties.
* While Llama 3.2 seems to be the worst performing on analyzing its outputs, a trend was observed. Over 64% of errors by Llama 3.2 returned the number of distinct objects in the prompt rather than the total objects suggesting this could be improved by some prompt fine tuning.
* Though Llama 3.1 has a higher accuracy, its reasoning capabilities seem more random with varying type of errors such as forgetting, miscalculation and making incorrect assumptions. An example of this was seen in a question where Llama 3.1 said that celery shouldn’t be counted as a vegetable, which makes no sense. Thus, even though Llama 3.2 has a lower accuracy and parameters,its reasoning and training process seems to be more robust and logical in comparison to its 3.1 counterpart.
* Deepseek models, though having an acceptable accuracy, had some glaring issues, while half of its errors were regular forgetting, parsing, or miscalculation errors. The remaining errors could mainly be categorized into these 3 types:

1. Its reasoning form of answering questions led to cases where the model would reason with itself multiple times, leading to it solving the question correctly, and in the next attempt, confusing itself and giving the wrong answer.
2. The second type of error was that the reasoning output was so long that it exceeded the limit without giving a final answer.
3. The last and most glaring issue was making incorrect assumptions about objects in about 15% of the errors. Some of the assumptions are given below.

Snakes aren’t animals

Garlic is a herb and not a vegetable

An accordion isn’t an instrument

These assumptions raise concerns about its ability to reason and perform tasks effectively

**Object Counting Task**

In this task, each model was evaluated on its ability to count either the total number of objects or a specific type of object mentioned in a given prompt.

**Example Prompt:**  
*“I have a yam, a cauliflower, a bed, two cabbages, a garlic, an oven, a carrot, a head of broccoli, a potato, a stalk of celery, a lettuce head, and a toaster. How many vegetables do I have?”*

Models were required to:

* Interpret the question correctly,
* Identify and classify each object appropriately, and
* Provide an accurate count based on the criteria in the prompt.

**Model Performance Overview:**

|  |  |  |
| --- | --- | --- |
| Model | Parameters | Accuracy |
| Deepseek | 1.5 billion | 50.4% |
| Deepseek | 8 billion | 67.6% |
| Gemma | 1 billion | 60.8% |
| Gemma | 4 billion | 73.6% |
| Llama 3.1 | 8 billion | 46.4% |
| Llama 3.2 | 1 billion | 40% |

While accuracy generally improves with model size, a closer analysis of the models’ reasoning capabilities offers deeper insights:

**Key Observations**

* **Common Error Types (Across All Models):**
  + Forgetting certain objects in the prompt
  + Improper parsing of the list
  + Basic arithmetic errors during counting
* **Gemma Models (Most Accurate & Reliable Reasoning):**
  + The 4B Gemma model achieved the highest accuracy (73.6%).
  + Around 70% of its errors were due to minor arithmetic mistakes *after* correctly parsing and identifying objects, indicating strong comprehension and retention abilities.
* **Llama Models:**
  + **Llama 3.2 (1B):** Though it had the lowest accuracy (40%), its reasoning pattern showed some consistency. About 64% of its errors came from returning the count of *distinct* objects rather than the *total*, implying that prompt tuning could significantly improve its performance.
  + **Llama 3.1 (8B):** Despite a higher parameter count and slightly better accuracy (46.4%), its errors were more erratic, ranging from misclassification to incorrect assumptions (e.g., saying celery isn’t a vegetable) and poor arithmetic, indicating a less stable reasoning process compared to Llama 3.2.
* **Deepseek Models:**
  + While performance improved with size (50.4% to 67.6%), Deepseek exhibited unique issues:
    1. **Self-contradictory reasoning:** The model often reasoned its way to the correct answer but then second-guessed itself and gave an incorrect final output.
    2. **Overly long outputs:** Some completions were so verbose they exceeded length limits without returning an answer.
    3. **Incorrect assumptions (≈15% of errors):** Examples include:
       - Claiming snakes aren’t animals
       - Labeling garlic as only a herb, not a vegetable
       - Stating that an accordion isn’t a musical instrument

These issues raise concerns about Deepseek’s grounding in general knowledge and its reliability in tasks requiring categorical reasoning.

* In this task, the models were expected to interpret the question accurately, identify and classify each object correctly, and provide an exact count based on the criteria specified in the prompt.
* While accuracy generally increased with model size, a closer examination of the reasoning processes revealed more nuanced differences in their capabilities.
* Common errors across all models included omitting certain objects from the prompt, incorrect list parsing, and basic arithmetic mistakes during counting.
* The Gemma models achieved the highest accuracy, with approximately 70% of their errors stemming from minor arithmetic issues despite correct object identification and parsing, highlighting their strong comprehension and retention abilities.
* Llama 3.2 (1B) recorded the lowest accuracy at 40%, though around 64% of its mistakes resulted from counting distinct objects instead of the total, indicating that performance could likely be improved with better prompt tuning.
* Llama 3.1 (8B) performed slightly better with 46.4% accuracy, but its errors were more erratic, ranging from misclassifications like labeling celery as not a vegetable to flawed assumptions and arithmetic mistakes, suggesting a less consistent reasoning pattern compared to Llama 3.2.
* Deepseek models showed improved performance with increased size, ranging from 50.4% to 67.6%, but also exhibited distinctive issues such as self-contradictory reasoning(where the model initially arrived at the correct answer but reversed it upon further reasoning), overly verbose outputs that exceeded length limits without producing a final result, and roughly 15% of errors stemming from incorrect assumptions, such as claiming that snakes aren’t animals, garlic is only a herb and not a vegetable, or that an accordion is not a musical instrument.

Formal Fallacy

The goal of this task is to determine whether the argument is **deductively valid or invalid** based solely on the structure of the reasoning presented.

"Here comes a perfectly valid argument: First of all, whoever is a schoolmate of Sondra is not a stepsister of Pricilla. In consequence, whoever is not a stepsister of Pricilla is a schoolmate of Sondra."  
Is the argument, given the explicitly stated premises, deductively valid or invalid?  
Options:  
-valid   
- invalid

|  |  |  |
| --- | --- | --- |
| Model | Parameters | Accuracy |
| Deepseek | 1.5 billion | 58% |
| Deepseek | 8 billion | 52% |
| Gemma | 1 billion | 62% |
| Gemma | 4 billion | 54% |
| Llama 3.1 | 8 billion | 48% |
| Llama 3.2 | 1 billion | 52% |

In this task all models faced various struggles in their ability to analyze whether the argument is valid or invalid.

As we can see that no model stands out in terms of performance, however, we did see certain trends when their rerasonings we analyzed in detail.

For the deepseek models, its ability to reason and question itself multiple times led to the model confusing itself many times and rechecking the arguments multiple times. Thus, the output exceeding the limit in 30% of the prompts, however when deepseek did make a prediction it was accurate approximately 80% of the time.

For Gemma and Llama there was no clear relation between parameter size and the outputs generated varied for each prompt. Even for the non deepseek models, 5% of the prompts were giving ambiguous outputs. Thus suggesting that there needs to be further investigation by not only increasing the prompts, but also fine-tuning the types of prompts to draw more conclusive inferences about the performance of models on these tasks

Here’s a refined version of your text with improved structure, clarity, and grammar:

### Formal Fallacy Task

The goal of this task is to assess whether a given argument is **deductively valid** or **invalid**, based solely on the logical structure of the reasoning.

**Example Argument:**

“Here comes a perfectly valid argument: First of all, whoever is a schoolmate of Sondra is not a stepsister of Pricilla. In consequence, whoever is not a stepsister of Pricilla is a schoolmate of Sondra.”

**Question:**  
Given the explicitly stated premises, is the argument deductively valid or invalid?  
**Options:**

* valid
* invalid

### Model Performance

|  |  |  |
| --- | --- | --- |
| Model | Parameters | Accuracy |
| Deepseek | 1.5 billion | 58% |
| Deepseek | 8 billion | 52% |
| Gemma | 1 billion | 62% |
| Gemma | 4 billion | 54% |
| Llama 3.1 | 8 billion | 48% |
| Llama 3.2 | 1 billion | 52% |

### Observations

* **General Performance:**  
  No model consistently outperformed the others, and overall accuracy remained relatively low across the board. All models struggled with determining the logical validity of the argument.
* **Deepseek Models:**
  + A notable trait was their tendency to repeatedly reason through the argument, often leading to **self-contradiction or confusion**.
  + In about **30% of the prompts**, this overthinking led to responses that **exceeded the output limit** without delivering a final answer.
  + However, when Deepseek did produce an answer, it was **correct approximately 80% of the time**, indicating strong potential when not hindered by verbosity.
* **Gemma and Llama Models:**
  + No clear correlation was observed between **parameter size and accuracy**.
  + The outputs varied unpredictably across different prompts.
  + Around **5% of responses** from these models were **ambiguous or unclear**, further highlighting the challenge of evaluating logical validity with current architectures.

### Conclusion

While some promising reasoning patterns emerged—particularly with Deepseek’s high accuracy on completed outputs—the overall inconsistencies and low average performance across models suggest the need for:

* A **larger and more diverse prompt set**, and
* **More targeted fine-tuning**, especially focusing on formal logic and inference patterns, to better evaluate and enhance model performance on deductive reasoning tasks.
* In this task, no model consistently outperformed the others, and overall accuracy remained relatively low across the board, with all models struggling to determine the logical validity of the arguments.
* Deepseek models frequently attempted to reason through arguments multiple times, often resulting in self-contradiction or confusion. In roughly 30% of prompts, this tendency to overthink led to outputs that exceeded the response limit without providing a final answer.
* However, when Deepseek did produce a response, it was correct approximately 80% of the time, indicating strong potential when not hindered by verbosity.
* Gemma and Llama models showed no clear correlation between parameter size and accuracy, and their outputs varied unpredictably across different prompts.
* Around 5% of responses from Gemma and Llama were ambiguous or unclear, underscoring the difficulty current architectures face in evaluating logical validity.
* While some promising reasoning patterns were observed, the overall inconsistencies and low average performance suggest a need for a larger and more diverse prompt set, as well as more targeted fine-tuning focused on formal logic and inference skills.