```
## Response
<think>

I need to add the numbers 5 and 4. Starting with 5, if I add 4 to it, the total will be 9. Therefore, the sum of 5 and 4 is 9.

</think>

**Solution:**

We are asked to calculate the sum of 5 and 4.

**Step 1:** Start with the number 5.

**Step 2:** Add 4 to it.

\[
5 + 4 = 9
\]

**Final Answer:** \(\boxed{9}\)

---

## Question
what is 1 plus 2

## Response
<think>

I need to determine the sum of 1 and 2.

Adding these two numbers together, 1 plus 2 equals 3.

Therefore, the answer is 3.

</think>

Sure! Let's solve the problem step by step.

**Problem:** What is \(1 + 2\)?

**Solution:**

To find the sum of 1 and 2, simply add the two numbers together:

\[
1 + 2 = 3
\]

**Answer:**

\(\boxed{3}\)

---

## Question
{question}

## Response
```

Listing 4 | An example prompt of using DeepSeek-V3 as a judge.

```
As an advanced reasoning problem evaluation assistant, your primary responsibility is
    to assess the accuracy of provided answers. You will be presented with a reasoning-
    related question, its corresponding reference answer, and an answer requiring
    evaluation.

## Answer Quality Classification
You have to carefully analyze and classify the answer into one of the following two
    levels:
1. **correct**: The answer fully aligns with the reference answer in both reasoning
    process and final conclusion, and address the question without any errors or
    omissions.
2. **incorrect**: The answer contains major errors in key reasoning steps or the final
    conclusion, or completely deviates from the core of the question. This indicates a
    fundamental misunderstanding or error in comprehending the question.

## Question
{question}

## Reference Answer
{reference}

## Answer to be Evaluated
{answer}

## Output Format
You need to combine the question and reference answer, first provide a detailed
    explanation of your analysis of the answer to be evaluated, then conclude with the
    final answer quality classification.
Output the following content in **JSON** format, including two key:
1. 'analysis': analysis of the answer's correctness;
2. 'correctness': correct/incorrect
```

B.3.3. 800K Supervised Data

Reasoning Data We curate a large set of reasoning prompts and generate reasoning trajectories by performing rejection sampling from the checkpoint of the first-stage RL training. In the previous stage, we only included data that could be evaluated using rule-based rewards. However, in this stage, we expand the dataset by incorporating additional data, some of which uses a generative reward model by feeding the ground-truth and model predictions into DeepSeek-V3 for judgment, an example prompt is provided in Listing 4. Additionally, because the model output is sometimes chaotic and difficult to read, we have filtered out chain-of-thought with mixed languages, long paragraphs, and code blocks. For each prompt, we sample multiple responses and retain only the correct ones. In total, we collect about 600k reasoning-related training samples.

Non-Reasoning Data For non-reasoning data, such as writing, factual QA, self-cognition, and translation, we adopt the DeepSeek-V3 pipeline and reuse portions of the SFT dataset of DeepSeek-V3. We also incorporate software engineering-focused data, including program repair and front-end web development, to enhance the model's ability to solve real-world problems. For certain non-reasoning tasks, we call DeepSeek-V3 to generate a potential chain-of-thought before answering the question by prompting. However, for simpler queries, such as "hello" we do not provide a CoT in response. In the end, we collected a total of approximately 200k training samples that are unrelated to reasoning.

When designing our thinking process style, we ask the model to follow key principles: First, keep each paragraph concise and digestible. Short paragraphs make ideas clearer and easier to follow. Second, adopt a conversational tone that feels natural and engaging. We avoid technical formatting like markdown to maintain a smooth reading experience. Third, and most importantly, the thinking process begins by understanding the complete user context. This means analyzing who our users are, what situations they're dealing with, and what they truly need - including those unstated needs that may lie beneath the surface of their initial request.

After eliciting these thinking processes from the model, human annotators meticulously verify the accuracy of the outputs. Our findings indicate that these artificial reasoning traces enhance the model's precision in interpreting user queries. Specifically, they effectively highlight format constraints, clarify user intentions, and elucidate the requisite structure of outputs. This methodological approach facilitates more accurate and responsive interactions between the model and users.

Table 5 | Data Statistics of SFT Data.

Domain	Num Samples	Avg Rounds	Avg Tokens
Math	395285	1.0	6094.2
Code	211129	1.1	7435.7
STEM	10124	1.0	4928.8
Logic	10395	1.0	2739.0
General	177812	1.1	1419.8
Total	804745	1.0	5355.3

SFT Data Statistics Table 5 summarizes the data statistics across various domains, based on approximately 800,000 supervised samples. It is worth noting that the majority of the data consists of single-turn interactions, which may limit the multi-turn conversational capabilities of DeepSeek-R1. We leave the expansion to multi-turn dialogue data as future work. The math-related data are primarily in Chinese and English, spanning a wide range of topics and difficulty levels. These questions are verifiable, either through deterministic rules or by reference to specific ground-truth answers. The code dataset encompasses not only competitive programming problems but also debugging tasks and project-oriented coding queries. STEM and logic-related questions, although smaller in volume compared to mathematics and code, are sourced from publicly available textbooks and online repositories. The general-domain queries are diverse, predominantly in Chinese and English, and include tasks such as creative writing, open-ended question answering, role-playing scenarios, and more.

Examples of SFT Trajectories We provide examples of SFT trajectories related to math, code, and non-reasoning in Listings 5, 6, and 7.

Listing 5 | An example SFT trajectory from reasoning data related to mathematics.

```
## Question
For positive integer $n$, define $S_n$ to be the minimum value of the sum \[ \sum_{k
    =1}^n \sqrt{(2k-1)^2+a_k^2}, \] where $a_1,a_2,\ldots,a_n$ are positive real
    numbers whose sum is 17. There is a unique positive integer $n$ for which $S_n$ is
    also an integer. Find this $n$.

## Response
```