```
## Response
<think>
Okay, I need to help the user write a short poem that ends with "was tentacles all
    along." Hmm, ending with that specific line suggests a twist or surprise, maybe
    something unexpected.

First, I'll think about the theme. Tentacles make me think of sea creatures, like
    octopuses or squids, so perhaps an underwater setting would work well. Maybe a
    treasure hunt or an adventure beneath the waves could set the scene.

I want the poem to build up anticipation. Starting with something mysterious, like
    shadows under the sea, could create intrigue. Then, introducing elements like a
    sunken chest or sailors' lore might add depth and keep the reader engaged.

Rhyme scheme is important for a poem. I'll probably go with something simple, like AABB
     or ABAB, to keep it flowing smoothly. Each stanza should have four lines to
    maintain structure.

Now, the twist at the end should be sudden but make sense. Perhaps the treasure they
    seek turns out to be something entirely different. The last line needs to be
    impactful, so building up the suspense through the previous lines will make the
    ending more surprising.

Let me outline the stanzas. The first stanza could set the scene with the ocean depths
    and a hidden secret. The second stanza can show the action, like diving for
    treasure, and then reveal the unexpected tentacles.

I'll make sure each line flows naturally and the imagery is vivid. Using words like "
    waves," "shadows," "caverns," and "mystery" can help paint the picture. Ending with
     the twist will leave the reader with a lasting impression.

Alright, putting it all together, I'll craft two stanzas that build up to the final
    line, ensuring the surprise feels satisfying and the poem feels complete.
</think>

Beneath the waves, where shadows creep,
A secret stirs in the ocean deep.
A sunken chest, a sailor's song,
The mystery whispered all along.

We dove for pearls, for gold, for lore,
Through caverns dark, to the ocean floor.
But what we found, so strange, so strong--
It wasn't treasure... it was tentacles all along.
```

B.4. Hyper-Parameters

B.4.1. Hyper-Parameters of DeepSeek-R1-Zero-Qwen-32B

To train DeepSeek-R1-Zero-Qwen-32B, we set the learning rate to 2e-6, the KL coefficient to 0.001, and the sampling temperature to 1 for rollout. For each question, we sample 16 outputs with a maximum length of 32,768. Each training step consists of 32 unique questions, resulting in a training batch size of 512 per step. Every 400 steps, we replace the reference model with the latest policy model. To accelerate training, each rollout generates 8,192 outputs, which are randomly split into 16 mini-batches and trained for only a single inner epoch.

B.4.2. Hyper-Parameters of SFT

For code-start SFT and the second-stage SFT, we fine-tune DeepSeek-V3-Base for 2-3 epochs using the curated dataset, as described in B.3. We employ a cosine decay learning rate scheduler, starting at 5×10^{-5} and gradually decreasing to 5×10^{-6}. The maximum context length is set to 32,768 tokens, and the batch size is 128.

B.4.3. Hyper-Parameters of Distillation

For distillation, we fine-tune the corresponding base model for 2–3 epochs using the 800k data described in Section B.3.3. The base model and initial learning rate are listed in Table 6. We employ a cosine decay learning rate scheduler that gradually decreases the learning rate to one-tenth of its initial value. The maximum context length is 32,768 tokens, and the batch size is 64.

Table 6 | DeepSeek-R1 Distilled Models, their corresponding Base Models, and Initial Learning Rates.

Distilled Model	Base Model	Initial Learning Rate
DeepSeek-R1-Distill-Qwen-1.5B	Qwen2.5-Math-1.5B	1×10^{-4}
DeepSeek-R1-Distill-Qwen-7B	Qwen2.5-Math-7B	8×10^{-5}
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B	7×10^{-5}
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B	6×10^{-5}
DeepSeek-R1-Distill-Llama-8B	Llama-3.1-8B	5×10^{-5}
DeepSeek-R1-Distill-Llama-70B	Llama-3.3-70B-Instruct	2×10^{-5}

B.4.4. Training Cost

Regarding our research on DeepSeek-R1, we utilized the A100 GPUs to prepare for the experiments with a smaller model (30B parameters). The results from this smaller model have been promising, which has allowed us to confidently scale up to 660B R1-Zero and R1. For the training of DeepSeek-R1-Zero, we employed 64*8 H800 GPUs, and the process required approximately 198 hours. Additionally, during the training phase of DeepSeek-R1, we utilized the same 64*8 H800 GPUs, completing the process in about 4 days, or roughly 80 hours. To create the SFT datasets, we use 5K GPU hours. The details are shown in Table 7.

B.5. Reward Hacking

In the context of LLM training, reward hacking refers to the phenomenon wherein a model exploits flaws or biases in the reward function, thereby achieving high reward scores without truly aligning with the underlying human intent. In our work, we observe such reward hacking behavior when employing the helpful reward model. Specifically, if the reward model contains systematic biases or inaccuracies, the LLM may learn to generate responses that are rated highly by the model but diverge from authentic human preferences. This misalignment can manifest in performance degradation on tasks requiring complex reasoning, as illustrated in Figure 6.

B.6. Ablation Study of Language Consistency Reward

To study the impact of the Language Consistency (LC) Reward, we conduct an ablation experiment on DeepSeek-R1-Distill-Qwen-7B. This model uses the same cold start data as DeepSeek-R1

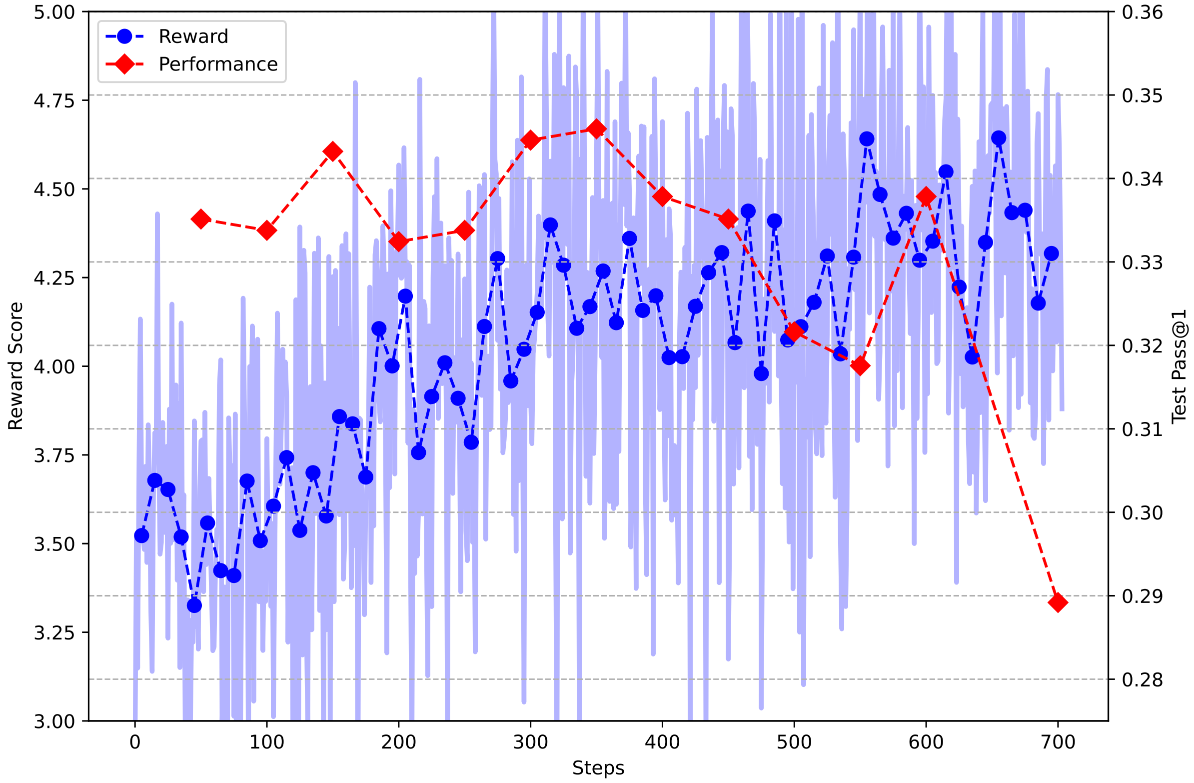

Figure 6 | Reward hacking: the reward exhibits an increasing trend as the performance on CodeForces decreases for training.