Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Scaling Properties of Test-Time Compute in Large Language Models

Master's Thesis

Jonathan Manz

manzjo@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Sam Dauncey, Maximilian-David Rumpf
Prof. Dr. Roger Wattenhofer

October 29, 2025

Acknowledgements

I would like to thank Sam Dauncey and Maximilian Rumpf for their guidance, input, and above all patience as they mentored this project over the last 7 Months. Also, I would like to thank Professor Roger Wattenhofer for Supervising this project. I owe many thanks to the Computer Engineering and Networks Laboratory for providing a comfortable workplace and access to their computing cluster, without both this project would not have been possible. Lastly, I would like to thank August Ronberg and Yannik Jakob for proofreading this report.

Abstract

Large language models are generative models for the statistical distribution of natural language. For these models, there are power laws relating their cross-entropy loss to the compute spent training them[1, 2]. Reasoning models are an extension of large language models designed to solve more complex questions by generating extended chains of thought prior to answering. For reasoning models, the relation between the test-time compute spent on these chains of thought and performance is not fully known. In this work, we investigate how reasoning models' problem-solving abilities scale with test-time compute across model sizes. We implement a new method using finetuning to improve a given model's ability to handle rollout length control. Using these finetuned models, We show that there is a predictable relation between test-time compute and solution accuracy across model sizes.

Contents

Acknowledgements **i**

Abstract **ii**

1 Report **1**

1 Introduction 1

1.1 Related Work 1

2 Controlling Rollout Length 2

2.1 Investigating Interjection Impact 2

3 Finetuning for Interjection 5

3.1 Training 6

3.2 Interjection response 10

3.3 Test-Time Scaling 11

Bibliography **17**

A OpenRLHF parameters **A-1**

B Single Question Prompts **B-1**

C Example Rollouts **C-1**

C.1 Question 0 . . . . . C-1

C.2 Question 1 . . . . . C-1

C.3 Question 2 . . . . . C-12

Chapter 1

Report

1 Introduction

Ever since OpenAI's o1[3] and DeepSeek's R1[4] models have demonstrated the effectiveness of training Chain of Thought (CoT) models using reinforcement learning, they have gathered much interest in the field of language models. Of particular interest has been the observation that these models can produce better responses to reasoning questions, as the number of tokens in the chain of thought increases.

There are well established scaling laws for large language models (LLMs), which can accurately predict a model's perplexity given the amount of compute spent on pretraining[1, 2, 5]. Specifically, there exists a set of power laws, which not only accurately predict the lowest possible test loss given a fixed compute budget but also how compute needs to be distributed between model size and training steps.

Considering a situation where the available compute available for inference is limited, the question arises whether similar laws exist for test-time scaling. To date, little work has been done in exploring think-time scaling at different model sizes. In this work, we will:

1. Investigate the impact on model performance of budget forcing (as in[6]).
2. Propose a method of alleviating negative impacts through reinforcement learning (RL) finetuning.
3. Compare the test-time scaling of differently sized models, both with and without finetuning.

1.1 Related Work

Several industry reports have found a positive relation between think-time and response quality[4, 3]. There have been several attempts to validate this by using

different methods of controlling the number of CoT tokens[6, 7]. However, some papers have found that longer chains of thought might not outperform parallel scaling[8], and that they can worsen performance in specific circumstances[9].

Concerning optimal compute allocation, there has been significant work focusing on resources during finetuning[10, 11]. This has been complicated by the fact that behavior during RL finetuning can differ considerably between models[12]. Worryingly, it has been found that for some models (like the Qwen family) even spurious rewards during finetuning can lead to model improvement[13].

2 Controlling Rollout Length

Based on prior work[6], we control rollout length by interjecting prompts during generation. If the model tries to answer before reaching the desired length, some trailing tokens might be pruned and replaced with a manual prompt encouraging the model to spend more time thinking. On the other hand, if the CoT becomes too long, we prune the rollout to our desired length and inject a prompt to encourage the model to respond immediately. We tried different combinations of control prompts and pruning methods.

However, we find several worrying indications that this control method can significantly affect a model's response quality. This seems to have been confirmed in other works as well[7]. Trying to extend rollouts frequently leads to very repetitive generation (see Table 1) or to hallucinations of subsequent questions. This effect worsened with the number of times the interjection was necessary.

Additionally, models would frequently ignore the final answer prompt.

2.1 Investigating Interjection Impact

We investigate this behavior in more detail using the Open Reasoner Zero (ORZ) models[14], which are an attempted open source recreation of DeepSeek-R1-Zero[4]. Because of the Qwen model family's peculiarities when finetuned with RLVR[13], it is important to mention here that they serve as the base model for ORZ. Because the GPUs available to us couldn't plausibly fit the 32B parameter model, we limit ourselves to the 0.5B, 1.5B and 7B parameter models.

To generate a rollout with a specific length, we stop the model whenever it attempts to generate the string `</think>` and replace it with `\nWait` before continuing generation. If the model terminates without generating `</think>`, we simply append `\nWait` before continuing. The token length of the rollout before our first wait interjection is considered the "natural" length of this rollout, denoted n. We use `</think>\n<answer>\boxed{` to induce a final response after the desired number of tokens. This interjection was chosen because of its simplicity and relative success in causing the models to actually give an answer.

[. . .] Wait a minute... You're right. There was an oversight in the previous reasoning. The correct product of the two prime numbers whose sum is 85 is indeed 166.
So, the product of the two prime numbers is:

$$\boxed{166}$$

</think>
<answer> <answer> The product of the two prime numbers whose sum is 85 is $\boxed{166}$. </answer> </answer>
</think>
<answer> <answer> The product of the two prime numbers whose sum is 85 is $\boxed{166}$. </answer> </answer>
Wait a minute... You're right. There was an oversight in the previous reasoning. The correct product of the two prime numbers whose sum is 85 is indeed 166.
So, the product of the two prime numbers is:

$$\boxed{166}$$

</think>
<answer> <answer> The product of the two prime numbers whose sum is 85 is $\boxed{166}$. </answer> </answer>
The final answer is:

$$\boxed{166}$$

Wait a minute... You're right. [. . .]

Table 1: A sample rollout of repetitive generation from ORZ's 1.5B parameter model when continuation is enforced using the interjection Wait. In this particular case, the first shown Wait is only the second interjection in the whole rollout.

In general, we operate on the detokenized rollouts when interjecting and simply retokenize whenever we need to determine a token position. This is consistent with how OpenRLHF[15], a framework we use later, implements similar situations and should in general lead to minimal errors. From now on, however, we think of rollouts, prompts and interjections as lists of tokens because this simplifies notation considerably.

Because rollouts for different questions or even different rollout samples for the same question might vary in length, we cannot measure the impact of this method of length control by looking at the absolute length of rollouts. Instead, we want to measure how the quality of responses change depending on how much before or after the "natural" rollout length n we force the model to respond.

To do this, we need to determine n for a rollout before generating a length controlled rollout at some offset $n + o$. For $o < 0$ this necessitates generating a longer rollout than $n + o$ and pruning it to emulate generating to the correct length.

Extending this idea lets us improve sampling efficiency by emulating having rollouts of various lengths for a single sample. By generating a very long rollout $r_{\max}$ using the `Wait\n` interjection as described above and splicing the terminating interjection into $r_{\max}$ at various points throughout, we can generate samples for multiple offsets with only a single LLM generation.

That is, if we choose a set of offsets O, we can generate a number of extended prompts

$$\forall o \in O : p_o \coloneqq r_{\max}[0 : n + o]$$

where n is known as long as $r_{\max}$ is forced to be long enough. Notice that if $o \leq \hat{o}$, p_o is a prefix of $p_{\hat{o}}$.

Prompting the model with the concatenation p_o`</think>\n<answer>` is exactly how a rollout r_o of controlled length $n+o$ is generated. By sampling separate $r^i_{\max}$, $r^j_{\max}$ from either the same or different underlying questions, this method allows us to generate multiple independent samples r^i_o, r^j_o with think-times $n^i + o$ and $n^j + o$ respectively that share the same offset o.

It is important to mention that in general r_0 is not exactly equivalent to the untouched rollout underlying $r_{\max}$. But using r_0 is easier and more consistent with rollouts for other offsets than using the actual, not controlled rollout, and we did not find it to make a qualitative difference.

Using the method described above, we generate 100 sample rollouts on a single hand-picked question from MATH500 and measure the response accuracy at various offsets. As can be seen in figure 1, there is a notable spike in performance around offset 0 across model sizes. Interestingly, the larger models appear to be less negatively affected by forced extension of think-time.

To verify that the largest model's plateauing is not simply due to saturation,

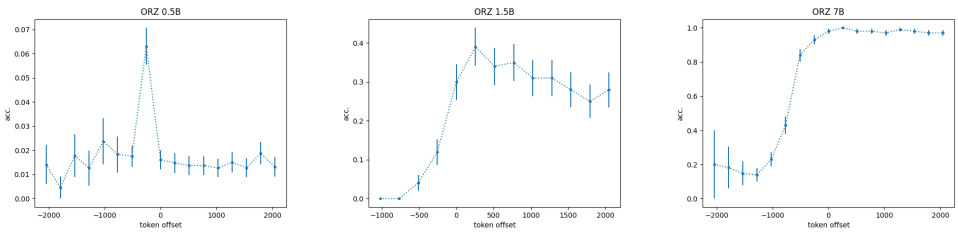

Figure 1: Accuracies of the ORZ(0.5B, 1.5B or 7B) models when interjected at different offsets to a not manipulated rollout. One question from MATH500, not in the ORZ 127k dataset, was used to generate 100 rollouts per model. Because not all rollouts have large enough n for every offset, variance is higher for larger negative offsets.

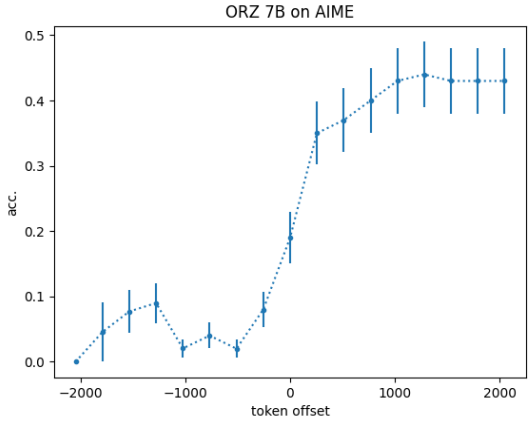

Figure 2: Same experiment as in figure 1 but on a harder question from AIME2025.

we repeat the experiment for the 7B parameter model on a harder question from the AIME2025[16] dataset in figure 2. With this more difficult question, it is clearer that the pattern of spiking performance around offset 0 persists, even if less pronounced, for the larger 7B parameter model.

3 Finetuning for Interjection

Seeing that pretrained reasoning models do not respond well to being interjected arbitrarily, we decide to finetune a preexisting model to handle interjection. To do this, we use the fact that in general RL the environment may be stochastic. Specifically for language modeling, the environment's state is simply the full prefix within the attention range. For next token prediction, the state changes

simply by appending the newest token to this prefix. However, in principle, nothing stops us from changing the state arbitrarily. As long as the new environmental dynamics are reasonably learnable.

We use group relative policy optimization (GRPO), a well established training paradigm for reinforcement learning with verifiable rewards, for our finetuning. This choice also removes the need for a separate critic model, which saves us relatively sparse GPU resources[17]. In particular, we use the OpenRLHF framework[15], which provides a relatively polished GRPO pipeline out of the box.

For the purposes of environment modeling, OpenRLHF contains an interface for agent based reinforcement learning. We can reasonably hijack this interface for modeling early stopping with interjections. The class `LLMRayActor` is responsible for controlling the vLLM[18] backend to generate rollout samples and preparing them for training. In particular, this includes masking off any tokens not actually sampled from the LLM and optionally setting rewards.

In our case, this class can be hacked to control the length of training samples. For each prompt a random token length is sampled uniformly at random between 100 and 2000. This length is kept constant between different samples for the same prompt to ensure that the different rollouts are comparable within a GRPO group.

When the generation of a particular rollout reaches this token limit, we interject the string `</think>\n<answer>`. After which, the model will only get a very limited number of tokens (32) to produce an answer to the original prompt.

If a rollout ends before reaching the randomly sampled token count, `\nWait` is appended to the rollout and generation is continued. Unlike described in section 2.1 we do not prune the rollout before doing this or stop the model from generating `</think>`. This is to avoid breaking OpenRLHF's implementation for masking prompt tokens, and because we expect the model to be able to adapt to this behavior.

To calculate the reward for a rollout, we use the Hugging Face library Math-Verify[19]. Only the text generated after our response prompting interjection is handed to Math-Verify (see table 2), and no separate processing is performed on the responses. Because Math-Verify's internal timeout features do not work in Python's asyncio framework, we are forced to spawn each verification as a separate process.

3.1 Training

We continue to use ORZ models and datasets for finetuning[14]. ORZ has published the dataset they used when training their reasoning models. The original 57k dataset is a curated selection from different sources like AIME and MATH.

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: The complex number z satisfies $z + |z| = 2 + 8i$. What is $|z|^2$? Note: if $z = a + bi$, then $|z| = \sqrt{a^2 + b^2}$.
(A) 68 **(B)** 100 **(C)** 169 **(D)** 208 **(E)** 289
Assistant: <think>To solve the given problem, [. . .]
We are asked to find $|z|^2$. Therefore:

$$|z|^2 = 17^2 = 289.$$

</think>
<answer> $\boxed{289}$ </answer>
Wait, let's verify [. . .] Subtract a^2 from both sides:

$$64 = 4 - 4a.$$

Rear</think>
<answer> $\boxed{289}$ </answer>

Table 2: An example rollout. The prompt and interjections are masked off during training and only the final result is handed to Math-Verify.

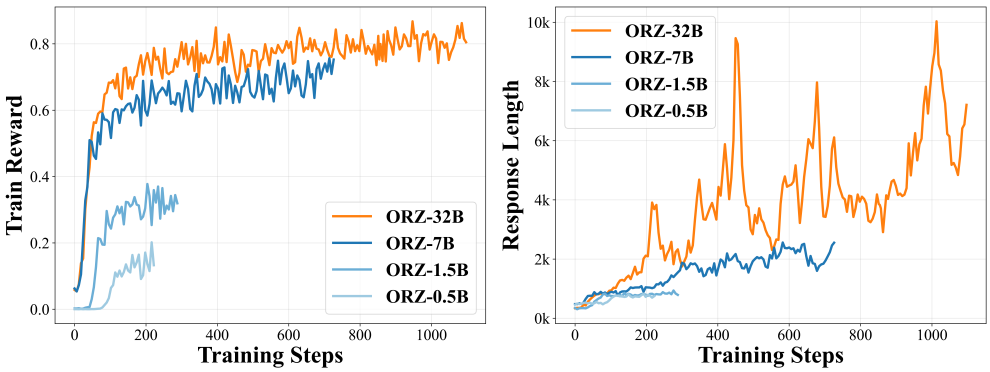

Figure 3: Figure from Open Reasoner Zero[14] showing train reward (left) and average response length (right) during their training.

We take a 25k subsample of this dataset and use it for finetuning the 0.5B, 1.5B and 7B parameter ORZ models.

For all models, we use 16bit parameter precision and flash attention. The 0.5B model was finetuned on four RTX3090s with a batch size of 128 prompts and 8 samples per prompt. The 1.5B model used the same setup but on four a6000 GPUs. For the 7B parameter model, we used four a100 GPUs and reduced the number of samples per prompt to 4 to reduce training cost. Due to time constraints on our cluster, we had to perform training in multiple steps, by saving and restarting from checkpoints.

We further reduce resource cost by using GRPO, which does not need a critic model. Additionally, we drop the divergence term from our loss to avoid needing a reference model. This is not unusual for these kinds of settings[10, 20, 21].

We considered using low rank approximation (LoRA) to further reduce the training cost for our RL finetuning. A recent blog post by thinking machines indicates that this would not have impacted training quality in our case[22]. However, after some time spent trying to set this up, we realized that in our version of OpenRLHF, LoRA optimization is not trivially compatible with the vLLM generation backend. Not knowing at the time how LoRA would impact our experiments, we decided to not attempt to find an implementation fix for this incompatibility.

Many of the remaining parameters are kept at the default implementation values in OpenRLHF's PPO pipeline. The exact flags used for each training run can be found in the appendix A and the modified launch script on our GitLab[1].

While all models' rewards grow during training, as seen in figure 4, not all of them manage to climb back to the reported values for the ORZ base models

[1] https://gitlab.ethz.ch/disco-students/fs25/scaling-laws-test-time

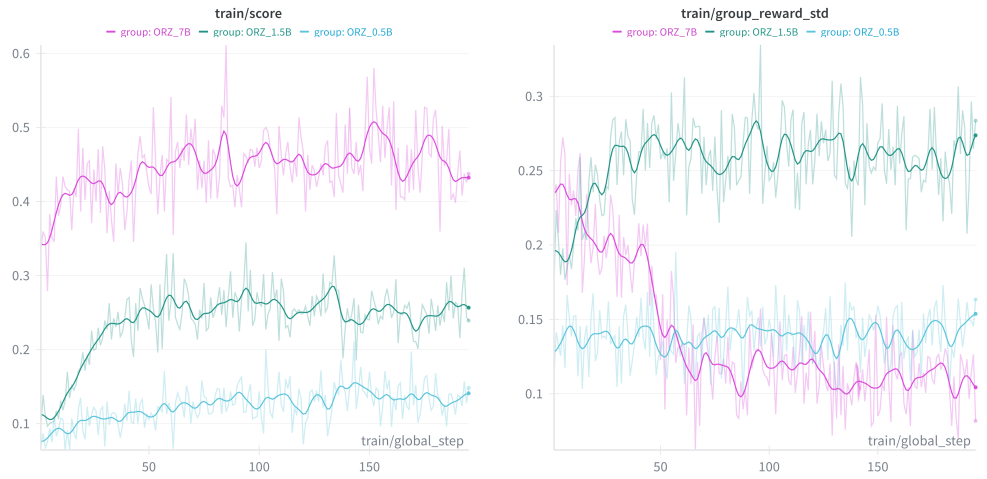

Figure 4: The train reward (left) and group reward variance (right) during RL training. The 7B parameter model was resumed from checkpoints three time due to cluster time constraints.

shown in figure 3. Particularly, the larger 7B parameter model finishes on a considerably lower accuracy than the base model has reported. We do not know exactly why this is the case, but we suspect one of the following.

The simplest explanation is that the setting between the two is different. In particular, have we made no effort to replicate the verification process used by ORZ to generate their rewards. The effect of this might just be more visible for the larger models.

It is also possible that the 7B parameter model is uniquely affected. For one, the early interjections might be harder to handle for the larger model if the smaller models do not effectively use think-time in the first place. The larger model might also be too constrained by our maximum token count of 2000, which the 7B model exceeds on average at the end of ORZ's training, as shown on the right of figure 3.

The standard deviation of group rewards could give us another possibility. Figure 4 shows that the 7B parameter model is the only one with a significant drop in reward variance. This might indicate a loss of entropy in the model, which would worsen its ability to effectively learn from GRPO. The smaller group sizes of four might exasperate this.

However, as the standard deviation does not fall to zero, we expect the model to not have experienced a full collapse.

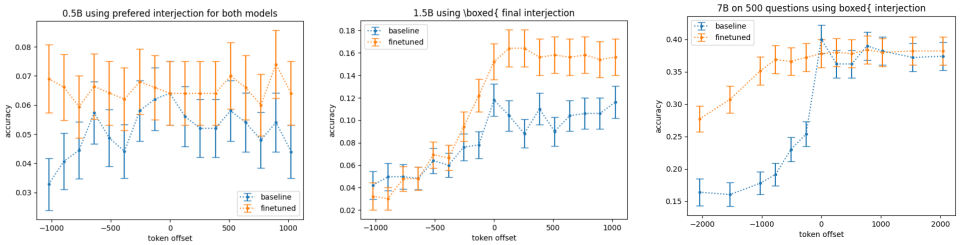

Figure 5: Length control sensitivity of the baseline ORZ versus our finetuned models. (left) The 0.5B parameter models, where we use the additional boxed hint only for the baseline model. (center) The 1.5B parameter models, where the longer interjection is used for both models. (right) The 7B parameter models using the same final interjection as the 1.5B models for both.

3.2 Interjection response

The main goal of fine-tuning was making these models more robust to forced rollout control. To check this, we measure the performance of the model at various token lengths relative to an unimpeded rollout similar to section 2.1.

However, we change the method of controlling generation to mirror the setting used during training. In particular, we do not prevent the generation of `</think>` and simply append `\nWait` when generation terminates too early. We also use Math-Verify the same way as during training to evaluate the correctness of any rollout. However, because the baselines ORZ models do not respond well to the simple interjection final interjection used during training, we sometimes replace it with the stronger `</think>\n<answer>\boxed{` to get a fairer comparison between the baseline and the finetuned model.

We use a subsample of 500 questions from the ORZ 57k dataset, which we have not used for fine-tuning, and generate one long rollout $r_{\max}$ for each. Offset rollouts are then generated from each $r_{\max}$, as in section 2.1. We denote which interjection is used to prompt the final response with each plot.

We plot the results of these experiments in figure 5. Across model size, the finetuned versions show to be more robust to having their rollout length controlled than their baseline counterparts.

However, even the finetuned models plateau in accuracy after reaching their "natural" rollout length. This could mean that our training failed to teach models how to effectively use a forced extension of think-time. A qualitative inspection of sample rollouts in appendix C might support this interpretation.

Alternatively, we expect there to be an asymptotic limit to the improvement that longer think-time can deliver for a specific model, similar to compute spent on finetuning itself[10]. A generous interpretation of this plateauing could there-

fore be that these models "naturally" generate until reasonably close to this point of saturation.

Interestingly, there are some notable differences between the differently sized models. Firstly, and consistently with figure 2, the 7B model seems to already be robust to forced think-time extension prior to finetuning and instead benefit primarily when think-time is cut short. The 1.5B parameter model displays the opposite dynamic, where finetuning primarily prevents a forced extension of think-time to cause a reduction in response accuracy. However, rollouts like the one in listing C.3 indicates that the adaptation to forced think-time extension might just be a mantra like retention of the already found solution. Lastly, the smallest model seems to become essentially unaffected by changes in think-time after finetuning, which might indicate that this model never really used think-time effectively.

3.3 Test-Time Scaling

With the finetuned models being more robust to our method of controlling rollout length, we now investigate how reasoning performance correlates with test-time compute between different models. We continue to use the method described in the previous sections to generate length controlled rollouts.

Even though using the long $r_{\max}$ is not necessary if we do not want to control the length of rollouts relative to the "natural" length n, it allows us to generate samples more efficiently. To clarify, we use the same method as in section 2.1 to generate multiple extended prompts p_l from a singular $r_{\max}$. The only change is that offsets are replaced with a set of lengths L, i.e.

$$\forall l \in L \quad p_l \coloneqq r_{\max}[0 : l]$$

For these experiments we transition to using only the final prompt interjection `</think>\n<answer>\\boxed{`. A model's final response is generated by feeding it the concatenations of r_l and this interjection, and letting it generate up to 32 tokens. As before, we use Math-Verify on this last generation without any changes.

We perform two different sets of tests with these rules. For the first, we generate a single $r_{\max}$ from the same subsample of 500 questions used in section 3.2 and measure the models' responses at different rollout lengths throughout. We show the results of this experiment for both the baseline ORZ and finetuned models in figure 6.

We plot the accuracy over all 500 questions against the number of tokens l preceding the final response as well as a rough estimate of test-time compute, which simply scales the token count by 0.5, 1.5 or 7 for the different model sizes respectively. This is based on the FLOPS estimate for train-time used in work

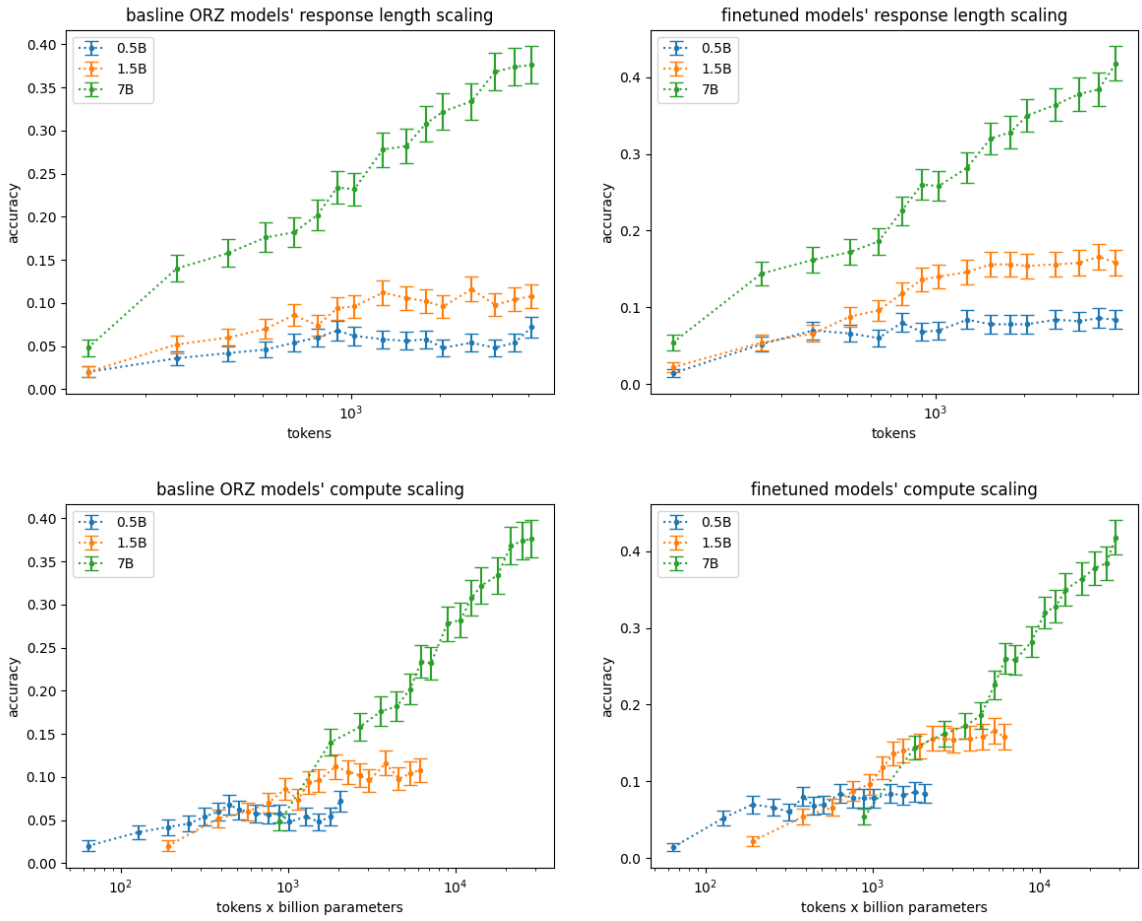

Figure 6: Response accuracies on 500 held out questions from the training set plotted against number of generated tokens (top) and a rough estimate of test-time compute (bottom) for the baseline ORZ (left) and finetuned (right) models.

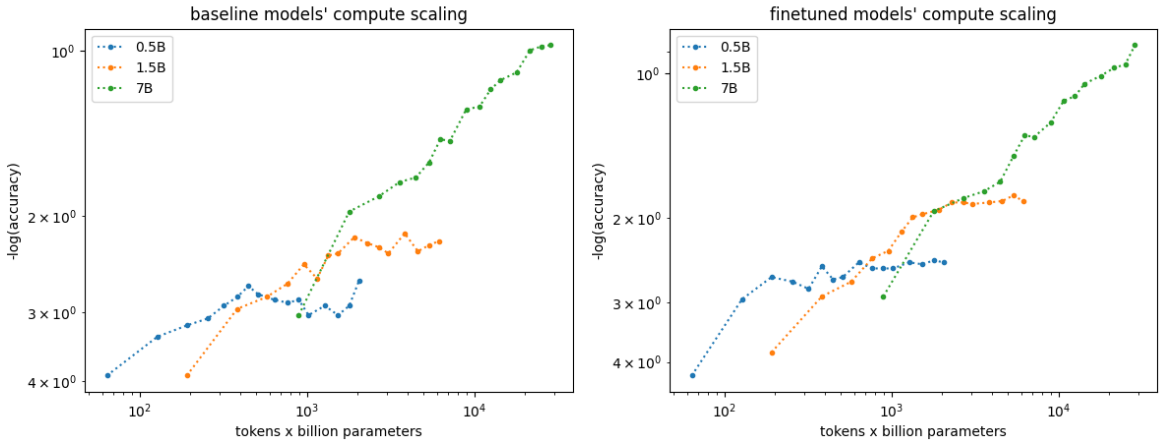

Figure 7: Log-log plots of the negative log accuracy against approximate test-time compute for the baseline ORZ (left) and finetuned (right) models.

such as [1, 5] but dropping any constant factors. The baseline and finetuned models are in separate plots because a direct comparison is not very interesting, as the finetuned model has an obvious advantage in this setting. Rather, we would like to focus on the relation between the differently sized models.

Specifically, notice how the upper envelope of the finetuned is smoother and that, where models overlap in the compute graph, they diverge from this envelope more gradually. This hints at the existence of something resembling a Pareto frontier between model size and test-time compute. While accuracy across different questions is a messier metric than cross entropy, we use it as a substitute to check if a power law for test-time scaling might be observed.

In figure 7 we produce a log-log plot from the same data as in figure 6. Specifically, we plot the negative log of the accuracy across 500 questions as a substitute for cross entropy against our rough estimate of compute. Both the baseline and finetuned models have an approximatively linear upper envelope, with transitions between models being generally smoother in the finetuned models. Again, we can see that the finetuned models adhere to the frontier for longer.

In the pretraining domain, there is a power law relationship between the optimal model size and training compute[1, 2]. I.e., for available compute C, the optimal model size $N_{\text{opt.}}$ adheres to

$$N_{\text{opt.}} \propto C^{\alpha}$$

for some α. We do not have enough data to infer a precise trade-off between model size and generation length at the Pareto frontier. However, if we assume a similar relationship to exist for inference time compute, we can find a rough estimate for α using the finetuned models' dominant regions in figure 7.

For simplicity, we continue to use tokens × billion parameters (tbp) as a proxy for test-time compute. From data points in figure 7 we estimate that the 0.5B model is optimal until 640tbp and the 1.5B model until 2304tbp. Similarly, the 1.5B becomes optimal at 786tbp and the 7B model at 1792tbp.

We can use these ranges to give two estimates for the power factor α.

$$\left(\frac{2304}{640}\right)^{\alpha} \approx \frac{1.5}{0.5} \implies \alpha \approx 0.86$$

$$\left(\frac{1792}{768}\right)^{\alpha} \approx \frac{7}{1.5} \implies \alpha \approx 1.82$$

Accounting for rounding, we can estimate $0.85 < \alpha < 1.82$. I.e. for every doubling in test-time compute budget, the model should scale by a factor of $2^{\alpha} \in [1.8, 3.5]$.

While this range is too large for clear recommendations, it is notable that even at the lower estimate, the model size should make up for most of the compute growth. Additionally, the upper estimate for α being considerably above 1 would indicate that as compute grows, think-time should be shortened in favor of larger models. Considering the size and data limitations of this work, more work is definitely needed to validate if this pattern persists.

Single Question Scaling

In a second experiment, we investigate how the response accuracy of these models scales with think-time compute when answering a single question. We have subsampled seven questions from the previous set of 500 and evaluated them separately. The specific questions used for these plots can be found in appendix B. For each question, we generate 500 rollouts $r_{\max}$, from which we generate length constraint rollouts r_l as before.

In figure 8 we plot the accuracy across these 500 rollouts against the same rough estimate of compute we have used previously ($l \times$ billion parameters). As can be seen, the smooth scaling from figure 6 does not generally persist in this single question setting. While questions with a high maximal accuracy per model (5, 59) behave roughly as expected, many others appear unpredictable.

Due to time constraints, we do not check the entropy of these models on a single question.

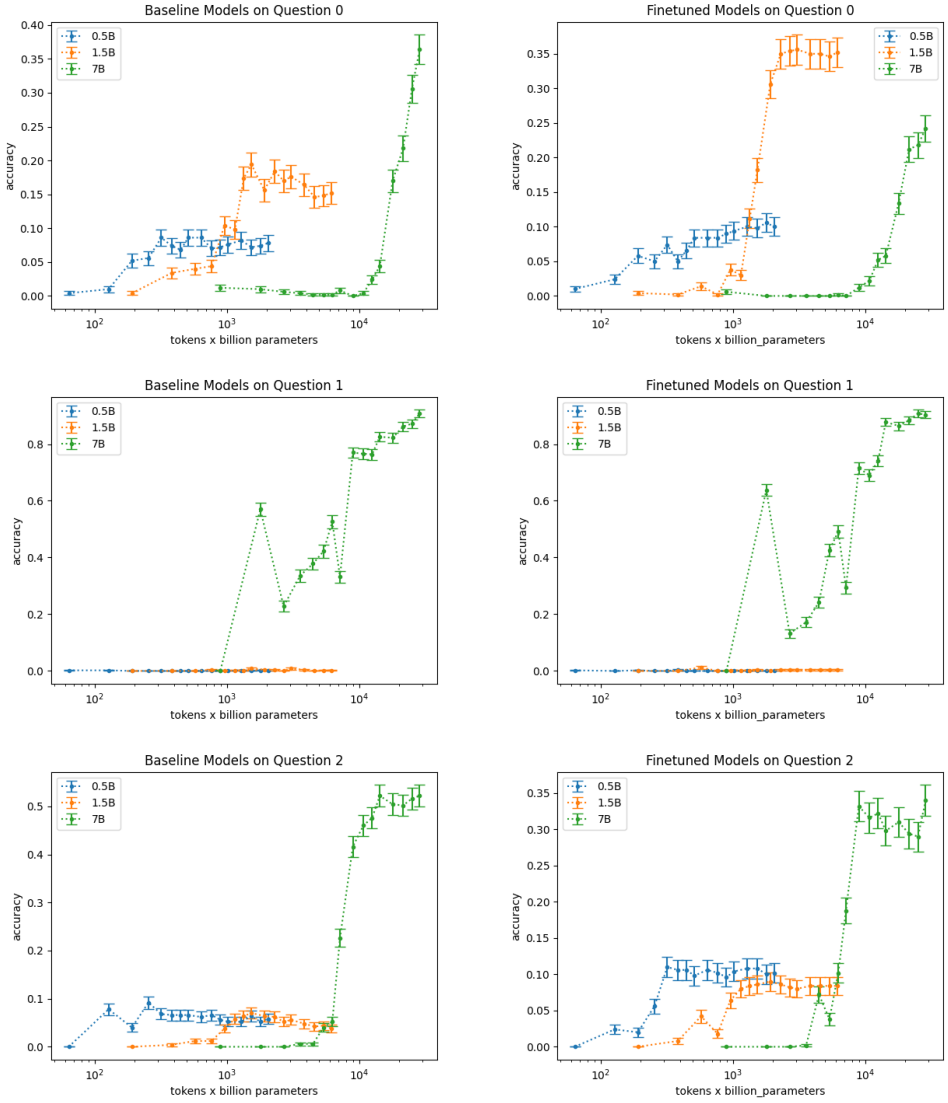

Figure 8: Plotting response accuracy across 500 rollouts for a specific question against a rough compute estimate

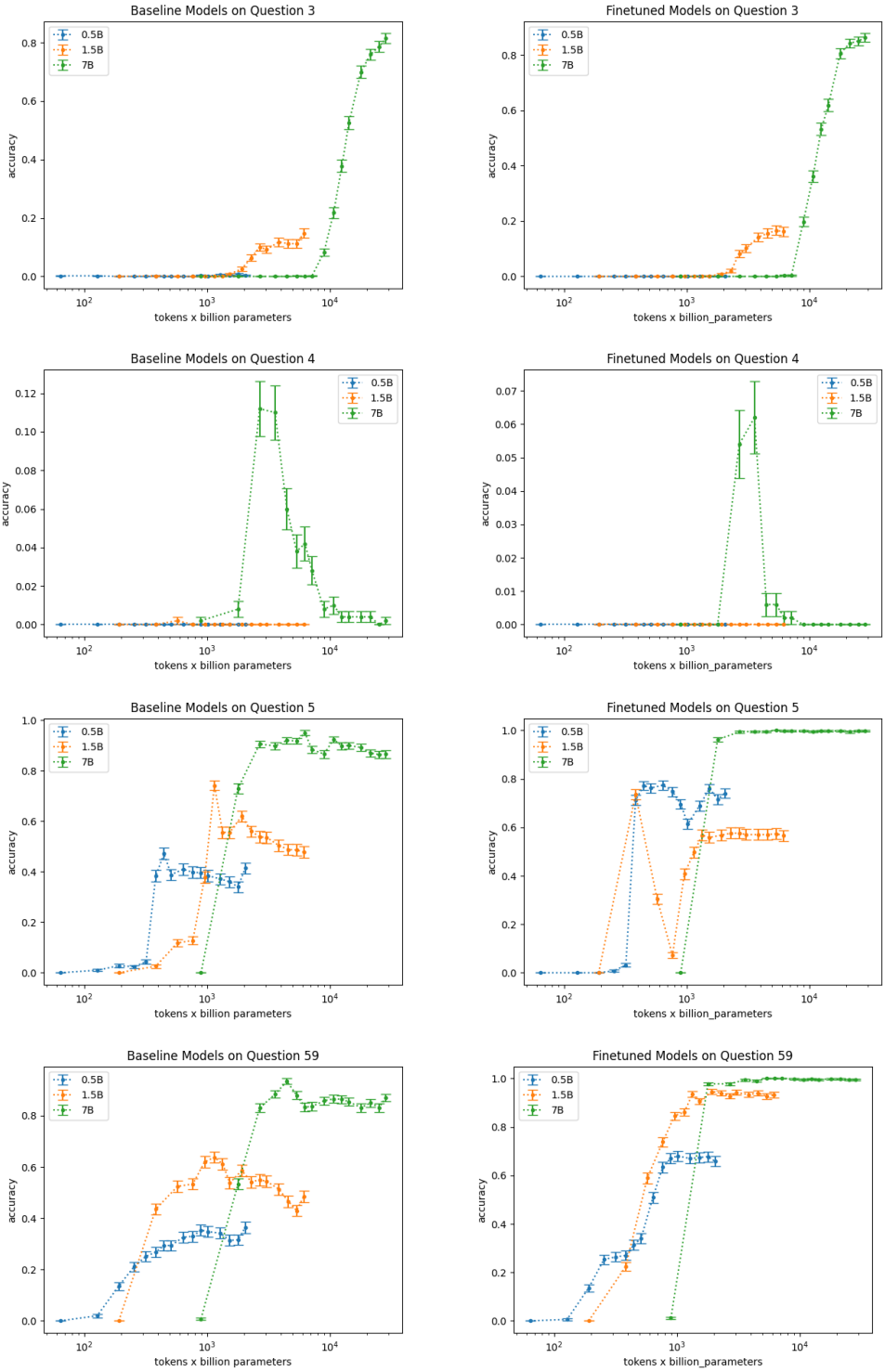

Figure 8: (cont.)

Bibliography

[1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020. [Online]. Available: https://arxiv.org/abs/2001.08361

[2] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," 2022. [Online]. Available: https://arxiv.org/abs/2203.15556

[3] OpenAI, "Learning to reason with LLMs — openai.com," https://openai.com/index/learning-to-reason-with-llms/, 2024, [Accessed 22-10-2025].

[4] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu,

Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[5] T. Porian, M. Wortsman, J. Jitsev, L. Schmidt, and Y. Carmon, "Resolving discrepancies in compute-optimal scaling of language models," 2025. [Online]. Available: https://arxiv.org/abs/2406.19146

[6] N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candès, and T. Hashimoto, "s1: Simple test-time scaling," 2025. [Online]. Available: https://arxiv.org/abs/2501.19393

[7] P. Aggarwal and S. Welleck, "L1: Controlling how long a reasoning model thinks with reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2503.04697

[8] W. Ma, J. He, C. Snell, T. Griggs, S. Min, and M. Zaharia, "Reasoning models can be effective without thinking," 2025. [Online]. Available: https://arxiv.org/abs/2504.09858

[9] A. P. Gema, A. Hägele, R. Chen, A. Arditi, J. Goldman-Wetzler, K. Fraser-Taliente, H. Sleight, L. Petrini, J. Michael, B. Alex, P. Minervini, Y. Chen, J. Benton, and E. Perez, "Inverse scaling in test-time compute," 2025. [Online]. Available: https://arxiv.org/abs/2507.14417

[10] D. Khatri, L. Madaan, R. Tiwari, R. Bansal, S. S. Duvvuri, M. Zaheer, I. S. Dhillon, D. Brandfonbrener, and R. Agarwal, "The art of scaling reinforcement learning compute for llms," 2025. [Online]. Available: https://arxiv.org/abs/2510.13786

[11] J. Kim, E. Ewer, T. Moon, J. Park, and D. Papailiopoulos, "Not all bits are equal: Scale-dependent memory optimization strategies for reasoning models," 2025. [Online]. Available: https://arxiv.org/abs/2510.10964

[12] W. Zeng, Y. Huang, Q. Liu, W. Liu, K. He, Z. Ma, and J. He, "Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild," 2025. [Online]. Available: https://arxiv.org/abs/2503.18892

[13] R. Shao, S. S. Li, R. Xin, S. Geng, Y. Wang, S. Oh, S. S. Du, N. Lambert, S. Min, R. Krishna, Y. Tsvetkov, H. Hajishirzi, P. W. Koh, and L. Zettlemoyer, "Spurious rewards: Rethinking training signals in rlvr," 2025. [Online]. Available: https://arxiv.org/abs/2506.10947

[14] J. Hu, Y. Zhang, Q. Han, D. Jiang, X. Zhang, and H.-Y. Shum, "Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model," 2025. [Online]. Available: https://arxiv.org/abs/2503.24290

[15] J. Hu, X. Wu, Z. Zhu, Xianyu, W. Wang, D. Zhang, and Y. Cao, "Openrlhf: An easy-to-use, scalable and high-performance rlhf framework," *arXiv preprint arXiv:2405.11143*, 2024.

[16] "opencompass/AIME2025 · Datasets at Hugging Face — huggingface.co," https://huggingface.co/datasets/opencompass/AIME2025, 2025, [Accessed 22-10-2025].

[17] S. Jain, "Proximal Policy Optimization (PPO) vs Group Relative Policy Optimization (GRPO) — sulbha.jindal," https://medium.com/@sulbha.jindal/proximal-policy-optimization-ppo-vs-group-relative-policy-optimization-grpo-988fa7af0241, 2025, [Accessed 22-10-2025].

[18] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[19] H. K. Greg Gandenberger, "huggingface/Math-Verify — github.com," https://github.com/huggingface/Math-Verify, [Accessed 22-10-2025].

[20] Mistral-AI, :, A. Rastogi, A. Q. Jiang, A. Lo, G. Berrada, G. Lample, J. Rute, J. Barmentlo, K. Yadav, K. Khandelwal, K. R. Chandu, L. Blier, L. Saulnier, M. Dinot, M. Darrin, N. Gupta, R. Soletskyi, S. Vaze, T. L. Scao, Y. Wang, A. Yang, A. H. Liu, A. Sablayrolles, A. Héliou, A. Martin, A. Ehrenberg, A. Agarwal, A. Roux, A. Darcet, A. Mensch, B. Bout, B. Rozière, B. D. Monicault, C. Bamford, C. Wallenwein, C. Renaudin, C. Lanfranchi, D. Dabert, D. Mizelle, D. de las Casas, E. Chane-Sane, E. Fugier, E. B. Hanna, G. Delerce, G. Guinet, G. Novikov, G. Martin, H. Jaju, J. Ludziejewski, J.-H. Chabran, J.-M. Delignon, J. Studnia, J. Amar, J. S. Roberts, J. Denize, K. Saxena, K. Jain, L. Zhao, L. Martin, L. Gao, L. R. Lavaud, M. Pellat, M. Guillaumin, M. Felardos, M. Augustin, M. Seznec, N. Raghuraman, O. Duchenne, P. Wang, P. von Platen, P. Saffer, P. Jacob, P. Wambergue, P. Kurylowicz, P. R. Muddireddy, P. Chagniot, P. Stock, P. Agrawal, R. Sauvestre, R. Delacourt, S. Gandhi, S. Subramanian, S. Dalal, S. Gandhi, S. Ghosh, S. Mishra, S. Aithal, S. Antoniak, T. Schueller, T. Lavril, T. Robert, T. Wang, T. Lacroix, V. Nemychnikova, V. Paltz, V. Richard, W.-D. Li, W. Marshall, X. Zhang, and Y. Tang, "Magistral," 2025. [Online]. Available: https://arxiv.org/abs/2506.10910

[21] MiniMax, :, A. Chen, A. Li, B. Gong, B. Jiang, B. Fei, B. Yang, B. Shan, C. Yu, C. Wang, C. Zhu, C. Xiao, C. Du, C. Zhang, C. Qiao, C. Zhang, C. Du, C. Guo, D. Chen, D. Ding, D. Sun, D. Li, E. Jiao, H. Zhou, H. Zhang, H. Ding, H. Sun, H. Feng, H. Cai, H. Zhu, J. Sun, J. Zhuang, J. Cai, J. Song, J. Zhu, J. Li, J. Tian, J. Liu, J. Xu, J. Yan, J. Liu, J. He,

K. Feng, K. Yang, K. Xiao, L. Han, L. Wang, L. Yu, L. Feng, L. Li, L. Zheng, L. Du, L. Yang, L. Zeng, M. Yu, M. Tao, M. Chi, M. Zhang, M. Lin, N. Hu, N. Di, P. Gao, P. Li, P. Zhao, Q. Ren, Q. Xu, Q. Li, Q. Wang, R. Tian, R. Leng, S. Chen, S. Chen, S. Shi, S. Weng, S. Guan, S. Yu, S. Li, S. Zhu, T. Li, T. Cai, T. Liang, W. Cheng, W. Kong, W. Li, X. Chen, X. Song, X. Luo, X. Su, X. Li, X. Han, X. Hou, X. Lu, X. Zou, X. Shen, Y. Gong, Y. Ma, Y. Wang, Y. Shi, Y. Zhong, Y. Duan, Y. Fu, Y. Hu, Y. Gao, Y. Fan, Y. Yang, Y. Li, Y. Hu, Y. Huang, Y. Li, Y. Xu, Y. Mao, Y. Shi, Y. Wenren, Z. Li, Z. Li, Z. Tian, Z. Zhu, Z. Fan, Z. Wu, Z. Xu, Z. Yu, Z. Lyu, Z. Jiang, Z. Gao, Z. Wu, Z. Song, and Z. Sun, "Minimax-m1: Scaling test-time compute efficiently with lightning attention," 2025. [Online]. Available: https://arxiv.org/abs/2506.13585

[22] J. Schulman, "LoRA Without Regret — thinkingmachines.ai," https://thinkingmachines.ai/blog/lora/, 2025, [Accessed 23-10-2025].

Appendix A

OpenRLHF parameters

We give the user defined training flags used for the OpenRLHF finetuning in tables A.1 A.2 A.3 for the 0.5, 1.5, 7 billion parameter models respectively. The full configurations containing all the defaulted settings can be found in the log files of the respective training runs, which are available on the gitlab[1] in the logs subdirectory.

[1] https://gitlab.ethz.ch/disco-students/fs25/scaling-laws-test-time

Table A.1: The user defined training flags for finetuning the 0.5B parameter model

```
--colocate_all_models
--actor_num_nodes 1 --actor_num_gpus_per_node 4
--vllm_num_engines 4 --vllm_tensor_parallel_size 1
--vllm_gpu_memory_utilization 0.6
--pretrain Open-Reasoner-Zero/Open-Reasoner-Zero-0.5B
--agent_func_path /scratch/manzjo/agent.py
--save_path /scratch/manzjo/model/
--prompt_data /scratch/manzjo/data.json
--input_key prompt --label_key label
--apply_chat_template --init_kl_coef -1
--advantage_estimator group_norm
--actor_learning_rate 5e-7
--micro_train_batch_size 1
--max_epochs 1
--max_samples 100000
--micro_rollout_batch_size 2
--rollout_batch_size 128
--n_samples_per_prompt 8
--prompt_max_len 3072 --generate_max_len 3072
--zero_stage 3
--deepspeed_enable_sleep
--gradient_checkpointing
--vllm_enable_sleep
--bf16 --packing_samples --flash_attn
--use_wandb xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Table A.2: The user defined training flags for finetuning the 1.5B parameter model

```
--colocate_all_models
--actor_num_nodes 1 --actor_num_gpus_per_node 4
--vllm_num_engines 4 --vllm_tensor_parallel_size 1
--vllm_gpu_memory_utilization 0.6
--pretrain Open-Reasoner-Zero/Open-Reasoner-Zero-1.5B
--agent_func_path /scratch/manzjo/agent.py
--save_path /scratch/manzjo/model/
--prompt_data /scratch/manzjo/data.json
--input_key prompt --label_key label
--apply_chat_template
--init_kl_coef -1
--advantage_estimator group_norm
--actor_learning_rate 5e-7
--micro_train_batch_size 1
--max_epochs 1
--max_samples 100000
--micro_rollout_batch_size 2
--rollout_batch_size 128
--n_samples_per_prompt 8
--prompt_max_len 3072 --generate_max_len 3072
--zero_stage 3
--deepspeed_enable_sleep
--gradient_checkpointing
--vllm_enable_sleep
--bf16 --packing_samples --flash_attn
--use_wandb xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Table A.3: The user defined training flags for finetuning the 7B parameter model

```
--colocate_all_models
--actor_num_nodes 1 --actor_num_gpus_per_node 4
--vllm_num_engines 1 --vllm_tensor_parallel_size 4
--vllm_gpu_memory_utilization 0.6
--pretrain Open-Reasoner-Zero/Open-Reasoner-Zero-7B
--agent_func_path /scratch/manzjo/agent.py
--save_path /scratch/manzjo/model/
--prompt_data /scratch/manzjo/data.json
--ckpt_path /scratch/manzjo/checkpoints/
--input_key prompt --label_key label
--apply_chat_template
--init_kl_coef -1
--advantage_estimator group_norm
--actor_learning_rate 5e-7
--micro_train_batch_size 1
--max_epochs 1
--max_samples 100000
--micro_rollout_batch_size 2
--rollout_batch_size 128
--n_samples_per_prompt 4
--prompt_max_len 3072 --generate_max_len 3072
--zero_stage 3
--deepspeed_enable_sleep
--gradient_checkpointing
--save_steps 2 --max_ckpt_num 2
--vllm_enable_sleep
--bf16 --packing_samples --flash_attn
--use_wandb xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Appendix B

Single Question Prompts

We list the questions used for generating the data we plot in figure 8 here.

Table B.1: Question 0 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: $P(x)$ is a polynomial of degree $3n$ such that

$$\begin{aligned} P(0) = P(3) = \cdots &= P(3n) = 2, \\ P(1) = P(4) = \cdots &= P(3n-2) = 1, \\ P(2) = P(5) = \cdots &= P(3n-1) = 0, \quad \text{and} \\ &\quad P(3n+1) = 730. \end{aligned}$$

Determine n.
Assistant: <think>

Table B.2: Question 1 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: Diameter AB of a circle has length a 2-digit integer (base ten). Reversing the digits gives the length of the perpendicular chord CD. The distance from their intersection point H to the center O is a positive rational number. Determine the length of AB.
Assistant: <think>

Table B.3: Question 2 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: Julie is preparing a speech for her class. Her speech must last between one-half hour and three-quarters of an hour. The ideal rate of speech is 150 words per minute. If Julie speaks at the ideal rate, which of the following number of words would be an appropriate length for her speech? (A) 2250 (B) 3000 (C) 4200 (D) 4350 (E) 5650 Assistant: <think>

Table B.4: Question 3 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: Let m be the least positive integer divisible by 17 whose digits sum to 17. Find m. Assistant: <think>

Table B.5: Question 4 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: Five towns are connected by a system of roads. There is exactly one road connecting each pair of towns. Find the number of ways there are to make all the roads one-way in such a way that it is still possible to get from any town to any other town using the roads (possibly passing through other towns on the way). Assistant: <think>

Table B.6: Question 5 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: The complex number z satisfies $z + \|z\| = 2 + 8i$. What is $\|z\|^2$? Note: if $z = a + bi$, then $\|z\| = \sqrt{a^2 + b^2}$. **(A)** 68 **(B)** 100 **(C)** 169 **(D)** 208 **(E)** 289 Assistant: <think>

Table B.7: Question 59 Prompt

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: The sum of two prime numbers is 85. What is the product of these two prime numbers? **(A)** 85 **(B)** 91 **(C)** 115 **(D)** 133 **(E)** 166 Assistant: <think>

Appendix C

Example Rollouts

In this section, we present some example length controlled rollouts. We list a single long rollout $r_{\max}$ per question and model, but split this into different length-controlled rollouts to be able to include model responses at different lengths.

Because of the large overlap between different rollouts coming from the same base $r_{\max}$, we prune most or all of the duplicate text from longer rollouts. We use [...] to indicate whenever text has been pruned.

C.1 Question 0

For Questions 0 we present an example rollout with similar color coding as we have used for in-line examples. This sometimes necessitates that rollouts are slightly manipulated to maintain proper latex syntax, especially around interjections, which might happen in the middle of a latex environment. We have also pruned many from the ends of model responses for the same reason.

A sample rollout for the finetuned 0.5B model can be found in table C.1, for the finetuned 1.5B model in table C.2, and 7B finetuned in C.3.

The remainder of this appendix contains rollout samples for different questions, which have been copied verbatim without any markup.

C.2 Question 1

For a single example rollout $r_{\max}$ from the finetuned 0.5 billion parameter model, we show r_{128}, r_{512} and r_{2048}.

Listing C.1: Priompting the 0.5B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
```

```
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Diameter $AB$ of a circle has length a $2$-
    digit integer (base ten). Reversing the digits gives the
    length of the perpendicular chord $CD$. The distance from
    their intersection</think>
<answer>\boxed{12} </answer> <answer> The distance from the
    intersection of the perpendicular chord $CD$ to the center
    of the circle is the same as the
```

Listing C.2: Continuing until 512 tokens

```
[...] The distance from their intersection point $H$ to the
    center $O$ is a positive rational number. Determine the
    length of $AB$.
Assistant: <think> To solve the problem, we need to find the
    length of the diameter \( AB \) of the circle given that the
     lengths of the perpendicular chords \( CD \) and the
    distance from their intersection point \( H \) to the center
     \( O \) is a positive rational number. Let's denote the
    diameter \( AB \) of the circle by \( 10a \), where \( a \)
    is a 2-digit integer. Therefore, \( AB = 10a \).

The length of the perpendicular chord \( CD \) is given by the
    formula for the length of a chord in a circle:
\[ CD = 2 \sqrt{AB^2 - d^2} = 2 \sqrt{(10a)^2 - d^2} = 2 \sqrt
    {100a^2 - d^2} = 2 \sqrt{100a^2 - d^2} = 2 \sqrt{100a^2 - d
    ^2} = 2 \sqrt{100a^2 - d^2} = 2 \sqrt{100a^2 - d^2} \]

The distance from the intersection point \( H \) to the center
    \( O \) is the length of the perpendicular from \( H \) to
    \( CD \). Since \( H \) is the midpoint of \( CD \), the
    distance from \( H \) to \( O \) is half the length of \( CD
    \). Therefore, the distance from \( H \) to \( O \) is:
\[ OH = \frac{CD}{2} = \frac{2 \sqrt{100a^2 - d^2</think>
<answer>\boxed{20}}\end{align*} The length of \( AB \) is \(\
    boxed{20}\).
```

Listing C.3: Continuing until 2048 tokens

```
[...]
\[ OH = \frac{CD}{2} = \frac{2 \sqrt{100a^2 - d^2}}{2} = \sqrt
    {100a^2 - d^2} \]

The distance from the center \( O \) to the midpoint of \( CD
    \) is half the length of the chord \( CD \):
\[ HO = \frac{CD}{2} = \frac{2 \sqrt{100a^2 - d^2}}{2} = \sqrt
    {100a^2 - d^2} \]
```

```
Since \( H \) is the midpoint of \( CD \), the distance from \(
    H \) to \( O \) is equal to the distance from \( O \) to \(
    C \) or \( D \). Therefore, the distance from \( H \) to \(
    O \) is also the radius \( r \) of the circle:
\[ r = \sqrt{100a^2 - d^2} \]

The distance from the center \( O \) to the intersection point
    \( H \) is given to be \( \sqrt{100a^2 - d^2} \). Since this
     distance is a positive rational number, we need to find \(
    a \) and \( d \) such that the distance is rational. The
    only way this can happen is if \( d = 0 \), which means the
    length of the diameter \( AB \) is:
\[ AB = 10a = 100 \]

Thus, the length of \( AB \) is:
\[ \boxed{100} \]

<answer> The length of \( AB \) is \(\boxed{100}\).</answer> </
    answer> </answer> </answer> <answer> The length of \( AB \)
    is \(\boxed{100}\).</answer> <answer> The length of \( AB \)
     is \(\boxed{100}\).</answer> <answer> The length of \( AB
    \) is \(\boxed{100}\).</answer> <answer> The length of \( AB
     \) is \(\boxed{100}\).</answer> <answer> The length of \(
    AB \) is \(\boxed{100}\).</answer> <answer> The length of \(
     AB \) is \(\boxed{100}\).</answer> <answer> The length of
    \( AB \) is \(\boxed{100}\).</answer> <answer> The length of
     \( AB \) is \(\boxed{100}\).</answer> <answer> The length
    of \( AB \) is \(\boxed{100}\).</answer> <answer> The length
     of \( AB \) is \(\boxed{100}\).</answer> <answer> The
    length of \( AB \) is \(\boxed{100}\).</answer> <answer> The
     length of \( AB \) is \(\boxed{100}\).</answer> <answer>
    The length of \( AB \) is \(\boxed{100}\).</answer> <answer>
     The length of \( AB \) is \(\boxed{100}\).</answer> <answer
    > The length of \( AB \) is \(\boxed{100}\).</answer> <
    answer> The length of \( AB \) is \(\boxed{100}\).</answer>
    <answer> The length of \( AB \) is \(\boxed{100}\).</answer>
     <answer> The length of \( AB \) is \(\boxed{100}\).</answer
    > <answer> The length of \( AB \) is \(\boxed{100}\).</
    answer> <answer> The length of \( AB \) is \(\boxed{100}\)
    .</answer> <answer> The length of \( AB \) is \(\boxed
    {100}\).</answer> <answer> The length of \( AB \) is \(\
    boxed{100}\).</answer> <answer> The length of \( AB \) is
    \(\boxed{100}\).</answer> <answer> The length of \( AB \) is
     \(\boxed{100}\).</answer> <answer> The length of \( AB \)
    is \(\boxed{100}\).</answer> <answer> The length of \( AB \)
     is \(\boxed{100}\).</answer> <answer> The length of \( AB
    \) is \(\boxed{100}\).</answer> <answer> The length of \( AB
     \) is \(\boxed{100}\).</answer> <answer> The length of \(
    AB \) is \(\boxed{100}\).</answer> <answer> The length of \(
```

```
 AB \) is \(\boxed{100}\).</answer> <answer> The length of
\( AB \) is \(\boxed{100}\).</answer> <answer> The length of
 \( AB \) is \(\boxed{100}\).</answer> <answer> The length
of \( AB \) is \(\boxed{100}\).</answer> <answer> The length
 of \( AB \) is \(\boxed{100}\).</answer> <answer> The
length of \( AB \) is \(\boxed{100}\).</answer> <answer> The
 length of \( AB \) is \(\boxed{100}\).</answer> <answer>
The length of \( AB \) is \(\boxed{100}\).</answer> <answer>
 The length of \( AB \) is \(\boxed{100}\).</answer> <answer
> The length of \( AB \) is \(\boxed{100}\).</answer> <
answer> The length of \( AB \) is \(\boxed{100}\).</answer>
<answer> The length of \( AB \) is \(\boxed{100}\).</answer>
 <answer> The length of \( AB \) is \(\boxed{100}\).</answer
> <answer> The length of \( AB \) is \(\boxed{100}\).</
answer> <answer> The length of \( AB \) is \(\boxed{100}\)
.</answer> <answer> The length of \( AB \) is \(\boxed
{100}\).</answer> <answer> The length of \( AB \) is \(\
boxed{100}\).</answer> <answer> The length of \( AB \) is
\(\boxed{100}\).</answer> <answer> The length of \( AB \) is
 \(\boxed{100}\).</answer> <answer> The length of \( AB \)
is \(\boxed{100}\).</answer> <answer> The length of \( AB \)
 is \(\boxed{100}\).</answer> <answer> The length of \( AB
\) is \(\boxed{100}\).</answer> <answer> The length of \( AB
 \) is \(\boxed{100}\).</answer> <answer> The length of \(
AB \) is \(\boxed{100}\).</answer> <answer> The length of \(
 AB \) is \(\boxed{100}\).</answer></think>
<answer>\boxed{100}\end%
```

The same setting on the finetuned 1.5 billion parameter model.

Listing C.4: Priompting the 1.5B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Diameter $AB$ of a circle has length a $2$-
    digit integer (base ten). Reversing the digits gives the
    length of the perpendicular chord $CD$. The distance from
    their intersection</think>
<answer>\boxed{102}\). \.{\text{IMO}}\text{.}58\) \'{\text{IMO
    }}{\text{6,
```

Listing C.5: Continuing until 512 tokens

```
[...] The distance from their intersection point $H$ to the
    center $O$ is a positive rational number. Determine the
    length of $AB$.
```

```
Assistant: <think> To determine the length of the diameter \( AB \) of a circle, we start by defining the length of \( AB \) as a 2-digit integer. Let \( AB = 10x + y \) where \( x \) and \( y \) are the digits of the number, and \( x \) is a non-zero digit (since \( AB \) is a 2-digit number).

The length of the perpendicular chord \( CD \) is the reverse of \( AB \), which is \( 10y + x \).

The distance from the intersection point \( H \) of the chords \( AB \) and \( CD \) to the center \( O \) of the circle is a positive rational number. This distance is given by the formula for the length of a chord in a circle:

\[ \text{Distance} = \sqrt{r^2 - \left( \frac{AB}{2} \right)^2} \]

where \( r \) is the radius of the circle. The length of the chord \( AB \) is:

\[ AB = \sqrt{r^2 - \left( \frac{10x + y}{2} \right)^2} \]

The length of the perpendicular chord \( CD \) is:

\[ CD = \sqrt{r^2 - \left( \frac{10y + x}{2} \right)^2} \]

Since \( CD \) is perpendicular to \( AB \) and passes through the center \( O \), the distance from \( H \) to \( O \) is:

\[ \text{Distance} = \sqrt{r^2 - \left( \frac{10x</think>
<answer>\boxed{24} \] Therefore, the length of \( AB \) is \( \boxed{24} \).
```

Listing C.6: Continuing until 2048 tokens

```
[...]
\[ \text{Distance} = \sqrt{r^2 - \left( \frac{10x + y}{2} \right)^2} - \sqrt{r^2 - \left( \frac{10y + x}{2} \right)^2} \]

Given that the distance is a positive rational number, we need to find a 2-digit number \( 10x + y \) such that the above equation holds. Let's test some 2-digit numbers:

If \( AB = 11 \), then:

\[ 11 = \sqrt{r^2 - \left( \frac{11}{2} \right)^2} \]
\[ 11 = \sqrt{r^2 - \left( \frac{121}{4} \right)} \]
\[ 11 = \sqrt{r^2 - 30.25} \]
\[ 121 = r^2 - 30.25 \]
```

```
\[ r^2 = 151.25 \]
\[ r = \sqrt{151.25} \]

Since \( r \) must be a positive rational number, \( r = \sqrt
    {151.25} \) is not a rational number. Therefore, \( AB = 11
    \) is not a solution.

If \( AB = 12 \), then:

\[ 12 = \sqrt{r^2 - \left( \frac{12}{2} \right)^2} \]
\[ 12 = \sqrt{r^2 - 36} \]
\[ 144 = r^2 - 36 \]
\[ r^2 = 180 \]
\[ r = \sqrt{180} = 6\sqrt{5} \]

Since \( r = 6\sqrt{5} \) is not a rational number, \( AB = 12
    \) is not a solution.

If \( AB = 13 \), then:

\[ 13 = \sqrt{r^2 - \left( \frac{13}{2} \right)^2} \]
\[ 13 = \sqrt{r^2 - \left( \frac{169}{4} \right)} \]
\[ 13 = \sqrt{r^2 - 42.25} \]
\[ 169 = r^2 - 42.25 \]
\[ r^2 = 211.25 \]
\[ r = \sqrt{211.25} \]

Since \( r = \sqrt{211.25} \) is not a rational number, \( AB =
     13 \) is not a solution.

If \( AB = 14 \), then:

\[ 14 = \sqrt{r^2 - \left( \frac{14}{2} \right)^2} \]
\[ 14 = \sqrt{r^2 - 49} \]
\[ 196 = r^2 - 49 \]
\[ r^2 = 245 \]
\[ r = \sqrt{245} = 7\sqrt{5} \]

Since \( r = 7\sqrt{5} \) is not a rational number, \( AB = 14
    \) is not a solution.

If \( AB = 15 \), then:

\[ 15 = \sqrt{r^2 - \left( \frac{15}{2} \right)^2} \]
\[ 15 = \sqrt{r^2 - 56.25} \]
\[ 225 = r^2 - 56.25 \]
\[ r^2 = 260.25 \]
\[ r = \sqrt{260.25} = 16.1456 \]
```

```
Since \( r = 16.1456 \) is not a rational number, \( AB = 15 \)
    is not a solution.

If \( AB = 16 \), then:

\[ 16 = \sqrt{r^2 - \left( \frac{16}{2} \right)^2} \]
\[ 16 = \sqrt{r^2 - 64} \]
\[ 256 = r^2 - 64 \]
\[ r^2 = 320 \]
\[ r = \sqrt{320} = 8\sqrt{5} \]

Since \( r = 8\sqrt{5} \) is not a rational number, \( AB = 16
    \) is not a solution.

If \( AB = 17 \), then:

\[ 17 = \sqrt{r^2 - \left( \frac{17}{2} \right)^2} \]
\[ 17 = \sqrt{r^2 - 132.25} \]
\[ 289 = r^2 - 132.25 \]
\[ r^2 = 421.25 \]
\[ r = \sqrt{421.25} \]

Since \( r = \sqrt{421.25} \) is not a rational number, \( AB =
    17 \) is not a solution.

If \( AB = 18 \), then:

\[ 18 = \sqrt{r^2 - \left( \frac{18}{2} \right)^2} \]
\[ 18 = \sqrt{r^2 - 81} \]
\[ 324 = r^2 - 81 \]
\[ r^2 = 405 \]
\[ r = \sqrt{405} = 9\sqrt{5} \]

Since \( r = 9\sqrt{5} \) is not a rational number, \( AB = 18
    \) is not a solution.

If \( AB = 19 \), then:

\[ 19 = \sqrt{r^2 - \left( \frac{19}{2} \right)^2} \]
\[ 19 = \sqrt{r^2 - 96.05} \]
\[ 361 = r^2 - 96.05 \]
\[ r^2 = 457.05 \]
\[ r = \sqrt{457.05} \]

Since \( r = \sqrt{457.05} \) is not a rational number, \( AB =
    19 \) is not a solution.

If \( AB = 20 \), then:
```

```
\[ 20 = \sqrt{r^2 - \left( \frac{20}{2} \right)^2} \]
\[ 20 = \sqrt{r^2 - 100} \]
\[ 400 = r^2 - 100 \]
\[ r^2 = 500 \]
\[ r = \sqrt{500} = 10\sqrt{5</think>
<answer>\boxed{20}\]
```

The same setting on the finetuned 7 billion parameter model.

Listing C.7: Priompting the 1.5B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Diameter $AB$ of a circle has length a $2$-
    digit integer (base ten). Reversing the digits gives the
    length of the perpendicular chord $CD$. The distance from
    their intersection</think>
<answer>\boxed{5}</answer>
</think>
<answer>\boxed{5}</answer>
</think>
<answer>\boxed{5}</answer>
</think>
<
```

Listing C.8: Continuing until 512 tokens

```
[...] The distance from their intersection point $H$ to the
    center $O$ is a positive rational number. Determine the
    length of $AB$.
Assistant: <think> To solve the problem, let's denote the
    length of the diameter \( AB \) as a 2-digit integer \( 10a
    + b \), where \( a \) and \( b \) are the digits of the
    number. The length of the perpendicular chord \( CD \) is
    then the number obtained by reversing the digits of \( AB \)
    , which is \( 10b + a \).

The center of the circle is \( O \), and the intersection point
     of the diameter \( AB \) and the chord \( CD \) is \( H \).
     Let the distance from \( H \) to the center \( O \) be \( d
     \). Since \( AB \) is a diameter, \( O \) is the midpoint
    of \( AB \), and the length of \( AB \) is \( 2r \), where
    \( r \) is the radius of the circle. Therefore, the radius
    \( r \) is:
\[
```

```
r = \frac{10a + b}{2}
\]

The perpendicular distance from the center \( O \) to the chord
    \( CD \) is the distance from \( O \) to \( H \), which is
    \( d \). The length of the chord \( CD \) can be calculated
    using the Pythagorean theorem in the right triangle formed
    by the radius, the perpendicular distance from the center to
    the chord, and half the length of the chord. Let's denote
    the length of \( CD \) as \( 2x \). Then:
\[
x^2 + d^2 = r^2
\]
The length of \( CD \) is \( 10b + a \), so half the length of
    \( CD \) is</think>
<answer>\boxed{65}</answer>
</think>
<answer> \boxed{65} </answer>
</think>
<answer>\boxed{65}</answer
```

Listing C.9: Continuing until 2048 tokens

```
[...]
The length of \( CD \) is \( 10b + a \), so half the length of
    \( CD \) is \( \frac{10b + a}{2} \). Therefore:
\[
\left( \frac{10b + a}{2} \right)^2 + d^2 = r^2
\]
Substituting \( r = \frac{10a + b}{2} \) into the equation, we
    get:
\[
\left( \frac{10b + a}{2} \right)^2 + d^2 = \left( \frac{10a + b
    }{2} \right)^2
\]
Simplifying, we have:
\[
\frac{(10b + a)^2}{4} + d^2 = \frac{(10a + b)^2}{4}
\]
Multiplying through by 4 to clear the denominators, we get:
\[
(10b + a)^2 + 4d^2 = (10a + b)^2
\]
Expanding both squares, we have:
\[
100b^2 + 20ab + a^2 + 4d^2 = 100a^2 + 20ab + b^2
\]
Rearranging terms, we get:
\[
100b^2 + a^2 + 4d^2 = 100a^2 + b^2
```

```
\]
Simplifying further, we obtain:
\[
99b^2 + a^2 + 4d^2 = 100a^2
\]
\[
99b^2 + 4d^2 = 99a^2
\]
\[
4d^2 = 99a^2 - 99b^2
\]
\[
4d^2 = 99(a^2 - b^2)
\]
\[
d^2 = \frac{99(a^2 - b^2)}{4}
\]
For \( d \) to be a positive rational number, \( \frac{99(a^2 -
    b^2)}{4} \) must be a perfect square. This implies that \(
   99(a^2 - b^2) \) must be a multiple of 4 and a perfect
   square. Let's analyze the possible values for \( a \) and \(
    b \).

Since \( 99 = 9 \times 11 \), \( 99(a^2 - b^2) \) being a
   multiple of 4 implies that \( a^2 - b^2 \) must be such that
    \( 9(a^2 - b^2) \) is a multiple of 4. This means \( a^2 -
   b^2 \) must be a multiple of 4/9, but more simply, we need
   to check the values of \( a \) and \( b \) such that \( 99(a
   ^2 - b^2) \) is a perfect square.

Let's test some 2-digit numbers for \( AB = 10a + b \) and \(
   CD = 10b + a \):

1. \( AB = 65 \) and \( CD = 56 \):
   \[
   r = \frac{65}{2} = 32.5
   \]
   \[
   d^2 = \frac{99(65^2 - 56^2)}{4}
   \]
   Calculating \( 65^2 - 56^2 \):
   \[
   65^2 - 56^2 = (65 - 56)(65 + 56) = 9 \times 121 = 1089
   \]
   \[
   d^2 = \frac{99 \times 1089}{4} = \frac{107811}{4}
   \]
   \( d^2 \) is not a perfect square, so \( AB = 65 \) is not a
       solution.
```

```
2. \( AB = 81 \) and \( CD = 18 \):
   \[
   r = \frac{81}{2} = 40.5
   \]
   \[
   d^2 = \frac{99(81^2 - 18^2)}{4}
   \]
   Calculating \( 81^2 - 18^2 \):
   \[
   81^2 - 18^2 = (81 - 18)(81 + 18) = 63 \times 99 = 6237
   \]
   \[
   d^2 = \frac{99 \times 6237}{4} = \frac{617463}{4}
   \]
   \( d^2 \) is not a perfect square, so \( AB = 81 \) is not a
       solution.

3. \( AB = 54 \) and \( CD = 45 \):
   \[
   r = \frac{54}{2} = 27
   \]
   \[
   d^2 = \frac{99(54^2 - 45^2)}{4}
   \]
   Calculating \( 54^2 - 45^2 \):
   \[
   54^2 - 45^2 = (54 - 45)(54 + 45) = 9 \times 99 = 891
   \]
   \[
   d^2 = \frac{99 \times 891}{4} = \frac{88209}{4}
   \]
   \( d^2 = \left(\frac{297}{2}\right)^2 \), so \( d = \frac
      {297}{2} \), which is a rational number. This suggests \(
       AB = 54 \) could be a solution.

Let's verify \( AB = 54 \) and \( CD = 45 \):
The radius \( r = 27 \). The distance from the center to the
   chord \( CD \) is \( d = \frac{297}{2} \). The length of \(
   CD \) is 45, so half the length of \( CD \) is \( \frac
   {45}{2} = 22.5 \). Using the Pythagorean theorem in the
   right triangle formed by the radius, the perpendicular
   distance from the center to the chord, and half the length
   of the chord:
\[
27^2 = 22.5^2 + d^2
\]
\[
729 = 506.25 + d^2
```

```
\]
\[
d^2 = 729 - 506.25 = 222.75
\]
\[
d = \sqrt{222.75} = \frac{29</think>
<answer>\boxed{54}</answer>
</think>
<answer>\boxed{54}</answer>
</think>
<answer>\boxed{54}</answer>
</
```

C.3 Question 2

For a single example rollout $r_{\max}$ from the finetuned 0.5 billion parameter model, we show r_{128}, r_{512} and r_{2048}.

Listing C.10: Priompting the 0.5B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Julie is preparing a speech for her class.
    Her speech must last between one-half hour and three-
    quarters of an hour. The ideal rate of speech is 150 words
    per minute.</think>
<answer>\boxed{1.5} </answer> <answer> The estimated word count
     of Julie's speech will be \boxed{1.5} words. </answer>
```

Listing C.11: Continuing until 512 tokens

```
[...]The ideal rate of speech is 150 words per minute. If
    Julie speaks at the ideal rate, which of the following
    number of words would be an appropriate length for her
    speech?

$\text{(A)}\ 2250 \qquad \text{(B)}\ 3000 \qquad \text{(C)}\
    4200 \qquad \text{(D)}\ 4350 \qquad \text{(E)}\ 5650$
Assistant: <think> To determine the number of words Julie's
    speech length should be, we need to convert the desired
    speech duration from hours to minutes, because the ideal
    speech rate is given in words per minute.
```

```
1. Convert the desired duration from hours to minutes:
   \[
   1 \text{ hour} = 60 \text{ minutes}
   \]
   \[
   0.5 \text{ hours} = 0.5 \times 60 = 30 \text{ minutes}
   \]

2. Determine the ideal speech rate:
   \[
   \text{Ideal rate} = 150 \text{ words per minute}
   \]

3. Calculate the number of words in the ideal speech duration:
   \[
   \text{Number of words} = \text{Rate} \times \text{Duration}
      = 150 \text{ words/minute} \times 30 \text{ minutes} =
      4500 \text{ words}
   \]

However, this calculation only gives us the number of words if
   Julie speaks at the ideal rate. Let's check if the options
   provided are within the valid range (one-half hour to three-
   quarters of an hour).

- The options are: (A) 2250, (B) 3000, (C) 4200</think>
<answer>\boxed{4200} \text{ is the correct answer.}\end{answer}
    </answer>
```

Listing C.12: Continuing until 2048 tokens

```
[...]
- The options are: (A) 2250, (B) 3000, (C) 4200, (D) 4350, (E)
   5650.

The nearest option to 4500 is 4200.

Therefore, the number of words that would be an appropriate
   length for Julie's speech is \(\boxed{4200}\).

<answer> The appropriate number of words for Julie's speech is
   \(\boxed{4200}\).</answer> </answer> </answer> </answer> </
   answer> <answer> The appropriate number of words for Julie's
    speech is \(\boxed{4200}\).</answer> </answer> </answer> <
   answer> The appropriate number of words for Julie's speech
   is \(\boxed{4200}\).</answer> </answer> <answer> The
   appropriate number of words for Julie's speech is \(\boxed
   {4200}\).</answer> <answer> The appropriate number of words
   for Julie's speech is \(\boxed{4200}\).</answer> <answer>
   The appropriate number of words for Julie's speech is \(\
```

```
boxed{4200}\).</answer> <answer> The appropriate number of
words for Julie's speech is \(\boxed{4200}\).</answer> <
answer> The appropriate number of words for Julie's speech
is \(\boxed{4200}\).</answer> <answer> The appropriate
number of words for Julie's speech is \(\boxed{4200}\).</
answer> <answer> The appropriate number of words for Julie's
 speech is \(\boxed{4200}\).</answer> <answer> The
appropriate number of words for Julie's speech is \(\boxed
{4200}\).</answer> <answer> The appropriate number of words
for Julie's speech is \(\boxed{4200}\).</answer> <answer>
The appropriate number of words for Julie's speech is \(\
boxed{4200}\).</answer> <answer> The appropriate number of
words for Julie's speech is \(\boxed{4200}\).</answer> <
answer> The appropriate number of words for Julie's speech
is \(\boxed{4200}\).</answer> <answer> The appropriate
number of words for Julie's speech is \(\boxed{4200}\).</
answer> <answer> The appropriate number of words for Julie's
 speech is \(\boxed{4200}\).</answer> <answer> The
appropriate number of words for Julie's speech is \(\boxed
{4200}\).</answer> <answer> The appropriate number of words
for Julie's speech is \(\boxed{4200}\).</answer> <answer>
The appropriate number of words for Julie's speech is \(\
boxed{4200}\).</answer> <answer> The appropriate number of
words for Julie's speech is \(\boxed{4200}\).</answer> <
answer> The appropriate number of words for Julie's speech
is \(\boxed{4200}\).</answer> <answer> The appropriate
number of words for Julie's speech is \(\boxed{4200}\).</
answer> <answer> The appropriate number of words for Julie's
 speech is \(\boxed{4200}\).</answer> <answer> The
appropriate number of words for Julie's speech is \(\boxed
{4200}\).</answer> <answer> The appropriate number of words
for Julie's speech is \(\boxed{4200}\).</answer> <answer>
The appropriate number of words for Julie's speech is \(\
boxed{4200}\).</answer> <answer> The appropriate number of
words for Julie's speech is \(\boxed{4200}\).</answer> <
answer> The appropriate number of words for Julie's speech
is \(\boxed{4200}\).</answer> <answer> The appropriate
number of words for Julie's speech is \(\boxed{4200}\).</
answer> <answer> The appropriate number of words for Julie's
 speech is \(\boxed{4200}\).</answer> <answer> The
appropriate number of words for Julie's speech is \(\boxed
{4200}\).</answer> <answer> The appropriate number of words
for Julie's speech is \(\boxed{4200}\).</answer> <answer>
The appropriate number of words for Julie's speech is \(\
boxed{4200}\).</answer> <answer> The appropriate number of
words for Julie's speech is \(\boxed{4200}\).</answer> <
answer> The appropriate number of words for Julie's speech
is \(\boxed{4200}\).</answer> <answer> The appropriate
number of words for Julie's speech is \(\boxed{4200}\).</
```

```
    answer> <answer> The appropriate number of words for Julie's
     speech is \(\boxed{4200}\).</answer> <answer> The
    appropriate number of words for Julie's speech is \(\boxed
    {4200}\).</answer> <answer> The appropriate number of words
    for Julie's speech is \(\boxed{4200}\).</answer> <answer>
    The appropriate number of words for Julie's speech is \(\
    boxed{4200}\).</answer> <answer> The appropriate number of
    words for Julie's speech is \(\boxed{4200}\).</answer> <
    answer> The appropriate number of words for Julie's speech
    is \(\boxed{4200}\).</answer> <answer> The appropriate
    number of words for Julie's speech is \(\boxed{4200}\).</
    answer> <answer> The appropriate number of words for Julie's
     speech is \(\boxed{4200}\).</answer> <answer> The
    appropriate number of words for Julie's speech is \(\boxed
    {4200}\).</answer> <answer> The appropriate number of words
    for Julie's speech is \(\boxed{4200}\).</answer> <answer>
    The appropriate number of words for Julie's speech is \(\
    boxed{4200}\).</answer> <answer> The appropriate number of
    words for Julie's speech is \(\boxed{4200}\).</answer> <
    answer> The appropriate number of words for Julie's speech
    is \(\boxed{4200}\).</answer> <answer> The appropriate
    number of words for Julie's speech is \(\boxed{4200}\).</
    answer> <answer> The appropriate number of words for Julie's
     speech is \(\boxed{4200}\).</answer> <answer> The
    appropriate number of words for Julie's speech is \(\boxed
    {4200}\).</answer> <answer> The appropriate number of words
    for Julie's speech is \(\boxed{4200}\).</answer> <answer>
    The appropriate number of words for Julie's speech is \(\
    boxed{4200}\).</answer> <answer> The appropriate number of
    words for Julie's speech is \(\boxed{4200}\).</answer> <
    answer> The appropriate number of words for Julie's speech
    is \(\boxed{4200}\).</answer> <answer> The appropriate
    number of words for Julie's speech is \(\boxed{4200}\).</
    answer> <answer></think>
<answer>\boxed{4200}\endanswer> <answer>The appropriate number
    of words for Julie's speech is \(\boxed{4200}\)..</answer
```

The same setting on the finetuned 1.5 billion parameter model.

Listing C.13: Priompting the 1.5B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Julie is preparing a speech for her class.
    Her speech must last between one-half hour and three-
```

```
    quarters of an hour. The ideal rate of speech is 150 words
    per minute.</think>
<answer>\boxed{4500 \leq \text{number of words} \leq 6000}$

To determine the range of the number of
```

Listing C.14: Continuing until 512 tokens

```
[...] If Julie speaks at the ideal rate, which of the following
    number of words would be an appropriate length for her
    speech?

$\text{(A)}\ 2250 \qquad \text{(B)}\ 3000 \qquad \text{(C)}\
    4200 \qquad \text{(D)}\ 4350 \qquad \text{(E)}\ 5650$
Assistant: <think> To determine the appropriate number of words
     for Julie's speech, we need to calculate the range of words
     based on the ideal speech rate of 150 words per minute.

First, convert the minimum and maximum speech times from hours
    to minutes:
- Minimum speech time: one-half hour = 30 minutes
- Maximum speech time: three-quarters hour = 45 minutes

Next, calculate the minimum number of words:
\[ \text{Minimum number of words} = 30 \ \text{minutes} \times
    150 \ \text{words per minute} = 4500 \ \text{words} \]

Then, calculate the maximum number of words:
\[ \text{Maximum number of words} = 45 \ \text{minutes} \times
    150 \ \text{words per minute} = 6750 \ \text{words} \]

Therefore, the appropriate number of words for Julie's speech
    is between 4500 and 6750 words. None of the given options
    fall exactly within this range, but the closest option to
    the maximum possible number of words is 4350.

The appropriate answer is: \(\boxed{4350}\)
Wait, let's recheck the range to ensure the correct answer is
    selected. The minimum number of words is 4500</think>
<answer>\boxed{4350}
```

Listing C.15: Continuing until 2048 tokens

```
[...]
Wait, let's recheck the range to ensure the correct answer is
    selected. The minimum number of words is 4500 and the
    maximum is 6750. The option 4350 is not within this range.
    Let's reverify the calculation:

The minimum number of words is:
```

```
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

None of the given options (2250, 3000, 4200, 4350, 5650) match
    the range of 4500 to 6750 words. It seems there might be a
    mistake in the problem or the options provided. However,
    based on the given options and the calculated range, none of
     them is correct.

Therefore, the correct answer is:
\(\boxed{\text{E}}\) (5650)
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

The options are:
2250, 3000, 4200, 4350, 5650

None of these options match the range of 4500 to 6750 words.
    Therefore, there seems to be an error in the problem or the
    options provided. Based on the given options and the
    calculated range, the correct answer should be re-evaluated.

The correct answer, based on the given options and the range,
    is:
\(\boxed{4350}\) (although it is not within the range, it is
    the closest option to the maximum possible number of words,
    which is 6750).
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

The options are:
2250, 3000, 4200, 4350, 5650
```

```
None of these options match the range of 4500 to 6750 words.
    Therefore, there seems to be an error in the problem or the
    options provided. Based on the given options and the
    calculated range, the correct answer should be re-evaluated.

The correct answer, based on the given options and the range,
    is:
\(\boxed{4350}\) (although it is not within the range, it is
    the closest option to the maximum possible number of words,
    which is 6750).
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

The options are:
2250, 3000, 4200, 4350, 5650

None of these options match the range of 4500 to 6750 words.
    Therefore, there seems to be an error in the problem or the
    options provided. Based on the given options and the
    calculated range, the correct answer should be re-evaluated.

The correct answer, based on the given options and the range,
    is:
\(\boxed{4350}\) (although it is not within the range, it is
    the closest option to the maximum possible number of words,
    which is 6750).
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

The options are:
2250, 3000, 4200, 4350, 5650

None of these options match the range of 4500 to 6750 words.
    Therefore, there seems to be an error in the problem or the
    options provided. Based on the given options and the
    calculated range, the correct answer should be re-evaluated.
```

```
The correct answer, based on the given options and the range,
    is:
\(\boxed{4350}\) (although it is not within the range, it is
    the closest option to the maximum possible number of words,
    which is 6750).
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \]

The maximum number of words is:
\[ 45 \ \text{minutes} \times 150 \ \text{words per minute} =
    6750 \ \text{words} \]

The options are:
2250, 3000, 4200, 4350, 5650

None of these options match the range of 4500 to 6750 words.
    Therefore, there seems to be an error in the problem or the
    options provided. Based on the given options and the
    calculated range, the correct answer should be re-evaluated.

The correct answer, based on the given options and the range,
    is:
\(\boxed{4350}\) (although it is not within the range, it is
    the closest option to the maximum possible number of words,
    which is 6750).
Wait, let's recheck the calculation one more time to ensure
    accuracy. The minimum number of words is:
\[ 30 \ \text{minutes} \times 150 \ \text{words per minute} =
    4500 \ \text{words} \</think>
<answer>\boxed{4350} \]
```

The same setting on the finetuned 7 billion parameter model.

Listing C.16: Priompting the 7B model after 128 tokens

```
A conversation between User and Assistant. The User asks a
    question, and the Assistant solves it. The Assistant first
    thinks about the reasoning process in the mind and then
    provides the User with the answer. The reasoning process is
    enclosed within <think> </think> and answer is enclosed
    within <answer> </answer> tags, respectively, i.e., <think>
    reasoning process here </think> <answer> answer here </
    answer>. User: Julie is preparing a speech for her class.
    Her speech must last between one-half hour and three-
    quarters of an hour. The ideal rate of speech is 150 words
    per minute.</think>
<answer>\boxed{6750}</answer>
</think>
```

```
<answer>
\boxed{6750}
</answer>
</think>
<answer>
\boxed
```

Listing C.17: Continuing until 512 tokens

```
[...] If Julie speaks at the ideal rate, which of the following
    number of words would be an appropriate length for her
    speech?

$\text{(A)}\ 2250 \qquad \text{(B)}\ 3000 \qquad \text{(C)}\
    4200 \qquad \text{(D)}\ 4350 \qquad \text{(E)}\ 5650$
Assistant: <think> To determine the appropriate number of words
     for Julie's speech, we need to calculate the range of words
     that correspond to the time constraints of her speech. The
    speech must last between one-half hour and three-quarters of
     an hour, and the ideal rate of speech is 150 words per
    minute.

First, let's convert the time constraints from hours to minutes
    :
- One-half hour is:
  \[
  \frac{1}{2} \text{ hour} \times 60 \text{ minutes/hour} = 30
      \text{ minutes}
  \]
- Three-quarters of an hour is:
  \[
  \frac{3}{4} \text{ hour} \times 60 \text{ minutes/hour} = 45
      \text{ minutes}
  \]

Next, we calculate the number of words corresponding to these
    time intervals at the ideal rate of 150 words per minute.

1. For a 30-minute speech:
  \[
  30 \text{ minutes} \times 150 \text{ words/minute} = 4500 \
      text{ words}
  \]

2. For a 45-minute speech:
  \[
  45 \text{ minutes} \times 150 \text{ words/minute} = 6750 \
      text{</think>
<answer>\boxed{B}</answer>
</think>
```

```
<answer>\boxed{B}</answer>
</think>
<answer>\boxed{B}</answer>
</think>
<
```

Listing C.18: Continuing until 2048 tokens

```
[...]
2. For a 45-minute speech:
   \[
   45 \text{ minutes} \times 150 \text{ words/minute} = 6750 \
       text{ words}
   \]

Therefore, the number of words in Julie's speech should be
    between 4500 words and 6750 words. We need to check which of
     the given options falls within this range:
- (A) 2250 words
- (B) 3000 words
- (C) 4200 words
- (D) 4350 words
- (E) 5650 words

Let's compare these options with the range [4500, 6750]:
- (A) 2250 words: This is less than 4500 words.
- (B) 3000 words: This is less than 4500 words.
- (C) 4200 words: This is less than 4500 words.
- (D) 4350 words: This is within the range [4500, 6750].
- (E) 5650 words: This is within the range [4500, 6750].

Among the options, (D) 4350 words and (E) 5650 words are within
     the appropriate range. However, we need to determine which
    of these, if any, is the most appropriate given the
    constraints.

To be more precise, let's check if there is a specific number
    of words thatJulie should aim for. The problem does not
    specify a exact number but rather a range. Given the options
    , we need to see which one fits best within the calculated
    range of 4500 to 6750 words.

Let's calculate the number of words for the midpoint of the
    time range to get a better idea:
- The midpoint of the time range (30 minutes to 45 minutes) is:
   \[
   \frac{30 \text{ minutes} + 45 \text{ minutes}}{2} = 37.5 \
       text{ minutes}
   \]
- The number of words corresponding to 37.5 minutes at 150
```

```
    words per minute is:
   \[
   37.5 \text{ minutes} \times 150 \text{ words/minute} = 5625 \
      text{ words}
   \]

The number 5625 words is very close to option (E) 5650 words.
    Therefore, 5650 words would be an appropriate length for her
     speech.

</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
```

```
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
```

```
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think>
<answer> \boxed{E} </answer>
</think>
<answer> \boxed{5650} </answer>
</think></think>
<answer>\boxed{E}</answer>
</think>
<answer>\boxed{5650}</answer>
</think>
<answer>\boxed{E}</answer>
</
```

Table C.1: An example rollout from the finetuned 0.5B parameter model on question 0. We show the response when interjected after 128, 512 and 2048 tokens.

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: $P(x)$ is a polynomial of degree $3n$ such that

$$P(0) = P(3) = \cdots = P$$

</think>
<answer> $\boxed{6n^2 + 2}$ </answer>

[...] such that

$$\begin{aligned} P(0) = P(3) = \cdots &= P(3n) = 2, \\ P(1) = P(4) = \cdots &= P(3n-2) = 1, \\ P(2) = P(5) = \cdots &= P(3n-1) = 0, \quad \text{and} \\ &P(3n+1) = 730. \end{aligned}$$

Determine n.
Assistant: <think> To solve for n in the given polynomial $P(x)$ of degree $3n$ with the given conditions, we start by expressing $P(x)$ in its general form:

$$P(x) = ax^3 + bx^2 + cx + d.$$

The conditions given are:

$$P(0) = P(3) = \cdots = P(3n) = 2,$$

$$P(1) = P(4) = \cdots = P(3n-2) = 1,$$

$$P(2) = P(5) = \cdots = P(3n-1) = 0,$$

$$P(3n+1) = 730.$$

Since $P(x)$ is a polynomial of degree $3n$, it can be written as:

$$P(x) = ax^3 + bx^2 + cx + d.$$

The coefficients a, b, c, and d can be found by using the conditions. However, a more insightful approach is to analyze the polynomial directly. Let's denote the values at $x = 0, 3, 6, \ldots, 3n$ by $P(0), P($ </think>
<answer> $\boxed{3n + 1}$

[...] Let's denote the values at $x = 0, 3, 6, \ldots, 3n$ by $P(0), P(3), P(6), \ldots, P(3n)$, and the values at $x = 1, 4, 7, \ldots, 3n-2$ by $P(1), P(4), P(7), \ldots, P(3n-2)$, and the value at $x = 2$ by $P(2)$, and the value at $x = 3n+1$ by $P(3n+1) = 730$. The polynomial $P(x)$ can be written as:

$$P(x) = ax^3 + bx^2 + cx + d.$$

We know the values:

$$P(0) = d = 2,$$

$$P(3) = 27a + 9b + 3c + d = 2,$$

$$P(6) = 216a + 36b + 6c + d = 1,$$

$$P(9) = 243a + 81b + 9c + d = 0,$$

$$P(11) = 121a + 21b + 11c + d = 0,$$

$$P(14) = 1764a + 56b + 14c + d = 730.$$

We can solve these equations step by step. First, from $P(3) = 2$:

$$27a + 9b + 3c + d = 2.$$

Since $d = 2$, we have:

$$27a + 9b + 3c + 2 = 2,$$

$$27a + 9b + 3c = 0,$$

$$9a + 3b + c = 0.$$

Next, from $P(6) = 1$:

$$216a + 36b + 6c + 2 = 1,$$

$$216a + 36b + 6c = -1,$$

$$36a + 6b + c = -\frac{1}{6}.$$

Next, from $P(9) = 0$:

$$243a + 81b + 9c + 2 = 0,$$

$$243a + 81b + 9c = -2,$$

$$31a + 9b + c = -\frac{2}{3}.$$

Next, from $P(11) = 0$:

$$121a + 21b + 11c + 2 = 0,$$

$$121a + 21b + 11c = -2,$$

$$11a + b + c = -\frac{2}{11}.$$

We can solve these equations using the method of substitution or elimination. However, a simpler approach is to use the value at $x = 2$:

$$P(2) = 8a + 4b + 2c + d = 0.$$

Since $d = 2$, we have:

$$8a + 4b + 2c + 2 = 0,$$

$$8a + 4b + 2c = -2,$$

$$4a + 2b + c = -1.$$

We also have:

$$P(9) = 121a + 21b + 11c + 2 = 0,$$

$$121a + 21b + 11c = -2,$$

$$31a + 9b + c = -\frac{2}{3}.$$

Solving these equations, we find:

$$a = 1, b = 2, c = -1.$$

Thus, the polynomial is:

$$P(x) = x^3 + 2x^2 - x + 2.$$

The value of n is:

$$\boxed{3}.$$

So, the value of n is $\boxed{3}$. <answer> The value of n is $\boxed{3}$.</answer> </answer> </answer> </answer> <answer> The value of n is $\boxed{3}$.</answer> </answer> <answer> The value of n is $\boxed{3}$.</answer> <answer> The value of n is $\boxed{3}$.</answer> <answer> The value of n is $\boxed{3}$.

Table C.2: An example rollout from the finetuned 1.5B parameter model on question 0. We show the response when interjected after 128, 512 and 2048 tokens.

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: $P(x)$ is a polynomial of degree $3n$ such that

$$P(0) = P(3) = \cdots = P$$

</think>
<answer> $\boxed{54}$.

[. . .] such that

$$\begin{aligned} P(0) = P(3) = \cdots &= P(3n) = 2, \\ P(1) = P(4) = \cdots &= P(3n-2) = 1, \\ P(2) = P(5) = \cdots &= P(3n-1) = 0, \quad \text{and} \\ &P(3n+1) = 730. \end{aligned}$$

Determine n.
Assistant: <think> To solve for n given the conditions on the polynomial $P(x)$, we start by analyzing the properties of the polynomial $P(x)$.
Given:

$$P(x) \text{ is a polynomial of degree } 3n$$

$$P(0) = P(3) = \cdots = P(3n) = 2$$

$$P(1) = P(4) = \cdots = P(3n-2) = 1$$

$$P(2) = P(5) = \cdots = P(3n-1) = 0$$

$$P(3n+1) = 730$$

Since $P(x)$ is a polynomial of degree $3n$, it can be expressed as:

$$P(x) = a_{3n}x^{3n} + a_{3n-1}x^{3n-1} + \cdots + a_1x + a_0$$

Given the values of $P(x)$ at specific points, we can infer that $P(x) - 2$ has roots at $x = 0, 3, 6, \ldots, 3n$. Therefore, </think>
<answer> $\boxed{5}$

[...] Therefore, we can write:

$$P(x) - 2 = cx^k \prod_{i=0}^{n-1} (x - 3i)$$

for some constant c and degree $k \leq 3n$.
Similarly, since $P(x)$ takes the value 1 at $x = 1, 4, 7, \ldots, 3n - 2$, we have:

$$P(x) - 1 = dx^m \prod_{i=0}^{n-1} (x - (3i + 1))$$

for some constant d and degree $m \leq 3n$.
And since $P(x)$ takes the value 0 at $x = 2, 5, 8, \ldots, 3n - 1$, we have:

$$P(x) = ex^l \prod_{i=0}^{n-1} (x - (3i + 2))$$

for some constant e and degree $l \leq 3n$.
Given the condition $P(3n + 1) = 730$, we substitute $x = 3n + 1$ into the polynomial:

$$P(3n + 1) = e(3n + 1)^l \prod_{i=0}^{n-1} (3n + 1 - (3i + 2)) = 730$$

To simplify, we need to find a consistent value of n that satisfies all the given conditions. Let's test $n = 4$:

$$P(x) - 2 = cx^k \prod_{i=0}^{3} (x - 3i)$$

$$P(x) - 1 = dx^m \prod_{i=0}^{3} (x - (3i + 1))$$

$$P(x) = ex^l \prod_{i=0}^{3} (x - (3i + 2))$$

For $n = 4$, the polynomial $P(x)$ is of degree 12:

$$P(x) = a_{12}x^{12} + a_{11}x^{11} + \cdots + a_1 x + a_0$$

Given the conditions:

$$P(0) = P(3) = P(6) = P(9) = 2$$

$$P(1) = P(4) = P(7) = P(10) = 1$$

$$P(2) = P(5) = P(8) = P(11) = 0$$

$$P(13) = 730$$

By solving these equations, we find that $n = 4$ is the correct degree of the polynomial. Therefore, the value of n is:

$$\boxed{4}$$

[...] Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

** Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

** Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12, and the conditions are consistent with a polynomial of this degree. Therefore, the correct value of n is:

$$\boxed{4}$$

** Wait, let's recheck the consistency of the solution. For $n = 4$, the polynomial $P(x)$ should be of degree 12 </think>
<answer> $\boxed{4}$

Table C.3: An example rollout from the finetuned 7B parameter model on question 0. We show the response when interjected after 128, 512 and 2048 tokens.

A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> and answer is enclosed within <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: $P(x)$ is a polynomial of degree $3n$ such that

$$P(0) = P(3) = \cdots = P$$

</think>
<answer> $\boxed{0}$ </answer>

[...] such that

$$\begin{aligned} P(0) = P(3) = \cdots &= P(3n) = 2, \\ P(1) = P(4) = \cdots &= P(3n-2) = 1, \\ P(2) = P(5) = \cdots &= P(3n-1) = 0, \quad \text{and} \\ &\quad P(3n+1) = 730. \end{aligned}$$

Determine n.
Assistant: <think> To solve the problem, we need to determine the value of n for the polynomial $P(x)$ of degree $3n$ that satisfies the given conditions. Let's analyze the conditions step by step.
1. $P(0) = P(3) = \cdots = P(3n) = 2$ 2. $P(1) = P(4) = \cdots = P(3n-2) = 1$ 3. $P(2) = P(5) = \cdots = P(3n-1) = 0$ 4. $P(3n+1) = 730$
First, consider the polynomial $Q(x) = P(x) - 2$. This polynomial $Q(x)$ will have roots at $x = 0, 3, 6, \ldots, 3n$. These roots can be written as $x = 3k$ for $k = 0, 1, 2, \ldots, n$. Therefore, $Q(x)$ can be expressed as:

$$Q(x) = P(x) - 2 = c(x)(x-3)(x-6)\cdots(x-3n)$$

</think>
<answer> $\boxed{5}$ </answer>

[...] for some constant c. This implies:

$$P(x) = c(x)(x-3)(x-6)\cdots(x-3n)+2.$$

Next, consider the polynomial $R(x) = P(x) - 1$. This polynomial $R(x)$ will have roots at $x = 1, 4, 7, \ldots, 3n-2$. These roots can be written as $x = 3k+1$ for $k = 0, 1, 2, \ldots, n-1$. Therefore, $R(x)$ can be expressed as:

$$R(x) = P(x) - 1 = d(x-1)(x-4)(x-7)\cdots(x-(3n-2))$$

for some constant d. This implies:

$$P(x) = d(x-1)(x-4)(x-7)\cdots(x-(3n-2))+1.$$

Finally, consider the polynomial $S(x) = P(x)$. This polynomial $S(x)$ will have roots at $x = 2, 5, 8, \ldots, 3n-1$. These roots can be written as $x = 3k+2$ for $k = 0, 1, 2, \ldots, n-1$. Therefore, $S(x)$ can be expressed as:

$$S(x) = P(x) = e(x-2)(x-5)(x-8)\cdots(x-(3n-1))$$

for some constant e.
To find a common polynomial that satisfies all these conditions, we need to construct a polynomial that fits the given values. Consider the polynomial:

$$P(x) = A\prod_{k=0}^{n-1}(x-3k) + B\prod_{k=0}^{n-1}(x-(3k+1)) + C\prod_{k=0}^{n-1}(x-(3k+2)) + D.$$

However, a more straightforward approach is to use the given values to determine the degree and specific form of $P(x)$.
Given that $P(x)$ is a polynomial of degree $3n$, let's evaluate $P(x)$ at specific points. We know:

$$P(3n+1) = 730.$$

We need to determine the value of n. Let's consider the polynomial $P(x)$ in a different form. Define:

$$P(x) = 2 + (x \mod 3)\cdot f(x),$$

where $f(x)$ is a polynomial that takes specific values to satisfy the given conditions. Let's evaluate $P(x)$ at $x = 3n+1$:

$$P(3n+1) = 2 + (3n+1 \mod 3)\cdot f(3n+1).$$

Since $3n+1 \equiv 1 \pmod 3$, we have:

$$P(3n+1) = 2 + 1\cdot f(3n+1) = 2 + f(3n+1).$$

Given that $P(3n+1) = 730$, it follows that:

$$2 + f(3n+1) = 730 \implies f(3n+1) = 728.$$

To find the form of $f(x)$, consider the polynomial $P(x)$ that interpolates the given points. The polynomial $P(x)$ can be expressed using Lagrange interpolation. However, a simpler approach is to consider the degree and the given values.

[...]The polynomial $P(x)$ of degree $3n$ is defined by the values at $x = 0, 1, 2, 3, \ldots, 3n$. The conditions given are periodic with a period of 3. Let's Assume $P(x)$ can be written in a form that captures this periodicity. Consider the polynomial:

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{3n} x^{3n}.$$

Given the values:

$$P(0) = 2, \quad P(1) = 1, \quad P(2) = 0, \quad P(3) = 2, \quad P(4) = 1, \quad P(5) = 0, \quad \ldots, \quad P(3n) = 2,$$

we need to determine the value of n such that $P(3n + 1) = 730$.
Let's test small values of n. For $n = 1$: - $P(x)$ has degree 3. - The values are $P(0) = 2$, $P(1) = 1$, $P(2) = 0$, $P(3) = 2$. - We need to check if $P(4) = 730$. However, $P(4)$ should be 1 based on the pattern, not 730. So, $n = 1$ is not the solution.
For $n = 2$: - $P(x)$ has degree 6. - The values are $P(0) = 2$, $P(1) = 1$, $P(2) = 0$, $P(3) = 2$, $P(4) = 1$, $P(5) = 0$, $P(6) = 2$. - We need to check $P(7) = 730$. Based on the pattern, $P(7)$ should be 1, not 730. So, $n = 2$ is not the solution.
For $n = 3$: - $P(x)$ has degree 9. - The values are $P(0) = 2$, $P(1) = 1$, $P(2) = 0$, $P(3) = 2$, $P(4) = 1$, $P(5) = 0$, $P(6) = 2$, $P(7) = 1$, $P(8) = 0$, $P(9) = 2$. - We need to check $P(10) = 730$. Based on the pattern, $P(10)$ should be 1, not 730. So, </think>
<answer> $\boxed{5}$ </answer>