

517 A Implementation Details

518 We implement all the experiments on 8 NVIDIA A100 GPUs. The whole computational resource
519 cost of this research is about 80 A100 GPU days, which is mainly spent on the answer generation.

520 For the answer generation, we use the vLLM [54] framework for acceleration. Following the
521 suggestions of DeepSeek ², we set the temperature as 0.6 to prevent endless repetitions, set the
522 maximum new generation length as 16,384, and set the rollout number as 8. We take the average
523 accuracy of all 8 rollouts as the accuracy of one question, and report the final accuracy by taking the
524 average accuracy of each question.

525 We use the following template for generation:

PROMPT:

Please reason step by step, and put your final answer within \boxed{ }.
This is the problem:
{problem}

Figure 11: Prompts used for answer generation.

526 Here, the {problem} will be replaced by a real question. After we finish generation, we extract the
527 answer inside the \boxed{ } for evaluation.

²<https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B>

B Reasoning Strength in LRMs is Pre-Planned

B.1 Experimental Details

We visualize the linear probing process in Figure 12. We first extract the activation of LRMs $h^{(l)}$ at the last token position. Then we train a linear regression for predicting the subsequent reasoning token number y , which is calculated through the tokenizer. For reducing the overfitting, we set the regularization term α in Equation 1 as 10.

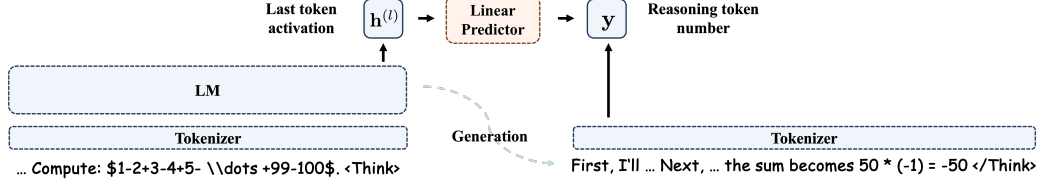


Figure 12: The procedure of linear probing.

B.2 Layer-wise Regression Results

We visualize the layer-wise regression results on R1-Distill-Qwen-14B and R1-Distill-Qwen-32B in Figure 13. On these two models, the linear regression exhibits the same pattern of an increasing trend as the model depth increases. Similarly, the correlation coefficient reaches over 0.8, indicating that the reasoning strength can be predicted before the model generation.

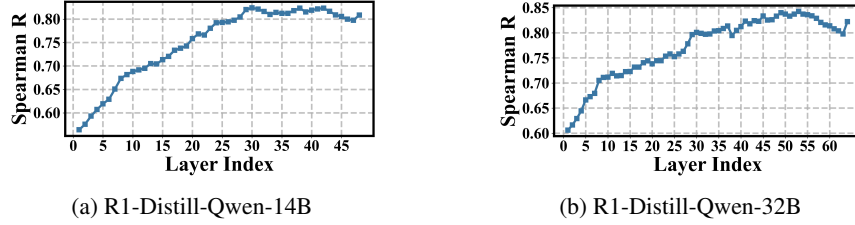


Figure 13: Layer-wise linear regression results

C LRMs Encode Reasoning Strength via a Pre-allocated Direction Vector

C.1 Existence of Pre-allocated Direction Vectors for Reasoning Strength Control

We visualize the cosine similarity matrix of the four extracted vectors from R1-Distill-Qwen-14B and R1-Distill-Qwen-32B in Figure 14. We have a similar observation that all these vectors exhibit extremely high cosine similarities near 1.0. This indicates that LRMs actually use a single direction vector for distinguishing questions of different difficulty levels.

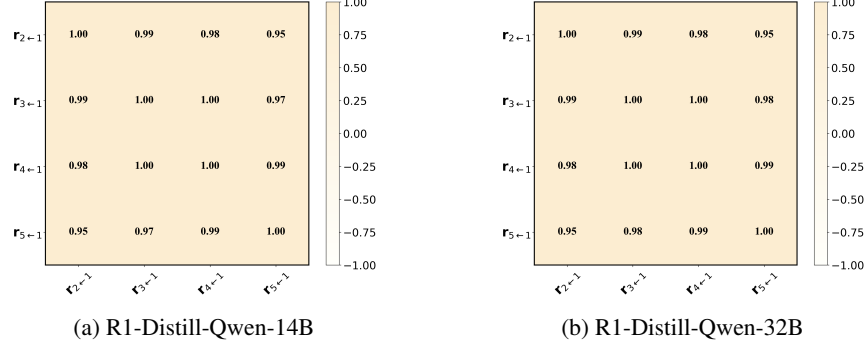


Figure 14: Cosine similarity between pre-allocated vectors across different difficulties. These vectors exhibit extremely high cosine similarities, indicating LRMs pre-allocate a single direction vector for distinguishing different question difficulties.

We visualize the layer-wise mean cosine similarities between four extracted vectors from R1-Distill-Qwen-14B and R1-Distill-Qwen-32B in Figure 15. We observe that these vectors exhibit consistently high cosine similarities, with an increasing trend as the layer depth increases. Finally, the mean cosine similarity reaches around 1.0, indicating these vectors become a single direction vector in the later layers.

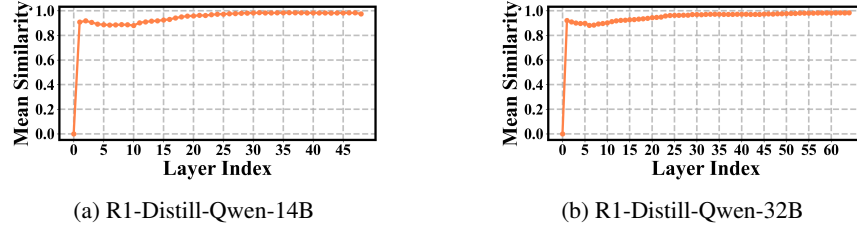


Figure 15: Layer-wise cosine similarities between four pre-allocated vectors

We visualize the L2 norm of these extracted four vectors from R1-Distill-Qwen-14B and R1-Distill-Qwen-32B in Figure 16. We observe that the norm of these vectors becomes bigger as the difficulty increases. Moreover, this trend is also similar to the increased reasoning token number as the difficulty increases, as shown in Figure 17. This indicates that LRMs use the magnitude of these direction vectors for handling different question difficulties.

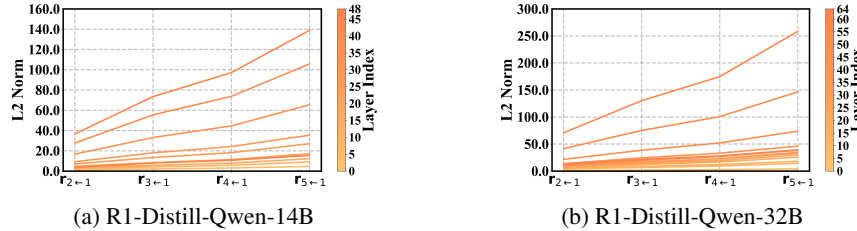
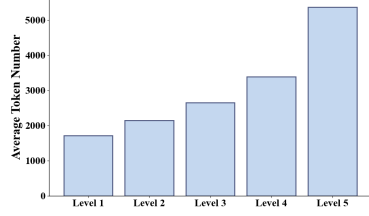
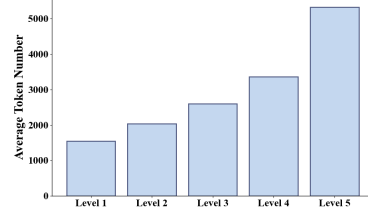


Figure 16: L2 norms of four pre-allocated vectors. The norm becomes bigger as the difficulty increases.



(a) R1-Distill-Qwen-14B

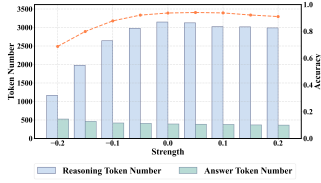


(b) R1-Distill-Qwen-32B

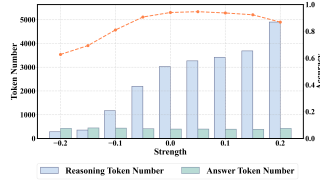
Figure 17: Average reasoning token number for different question difficulties.

555 C.2 Pre-allocated Vectors Causally Affect the Reasoning Strengths

556 We apply the activation steering at each layer and each position of LRMs. We provide more results
 557 when steering with the pre-allocation vector $\mathbf{r}^{(l)}$, in Figure 18, Figure 19, and Figure 20. We have
 558 similar observations to those in Section 4.2. When steering with negative λ , we observe a consistent
 559 decreasing trend in the reasoning token number and the decreased performance. When steering
 560 with positive λ , we observe a consistent increasing trend in the reasoning token number. However,
 561 despite appropriate positive λ can improve the performance, this is not consistent as the λ increases.
 562 We attribute this to the capability upper bound of these LRMs. Moreover, the steering only affects
 563 the reasoning token number, while maintaining the answer token number largely unchanged. This
 564 indicates that the pre-allocated direction vector is mainly responsible for the reasoning token number.

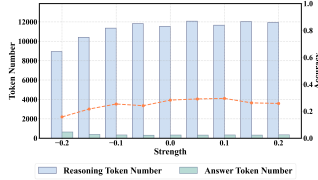


(a) R1-Distill-Qwen-14B

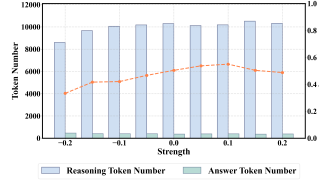


(b) R1-Distill-Qwen-32B

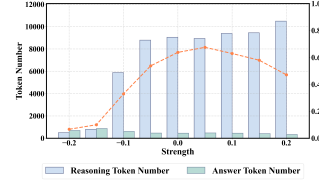
Figure 18: The causal effect on the reasoning token number and corresponding performance under different steering strength λ on the dataset MATH500.



(a) R1-Distill-Qwen-1.5B

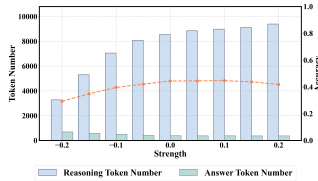


(b) R1-Distill-Qwen-7B

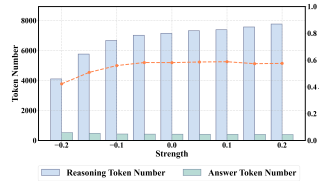


(c) QwQ-32B

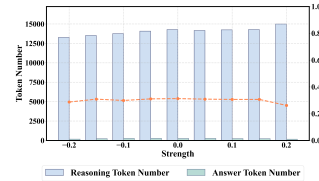
Figure 19: The causal effect on the reasoning token number and corresponding performance under different steering strength λ on the dataset AIME.



(a) R1-Distill-Qwen-1.5B



(b) R1-Distill-Qwen-7B



(c) QwQ-32B

Figure 20: The causal effect on the reasoning token number and corresponding performance under different steering strength λ on the dataset OlympiadBench.

565 C.3 Pre-allocation Vectors Yield Positive Reasoning Token Number Prediction

566 We provide more results in predicting the reasoning token number directly using the pre-allocation
 567 vectors in Figure 21. As shown in this figure, in most cases, the pre-allocation vectors yield positive
 568 predictions, indicating the close correlation of such vectors with our obtained predictors. This
 569 suggests that LRMs are indeed using such pre-allocated vectors for planning their reasoning strength,
 570 and the predication also largely relies on these vectors.

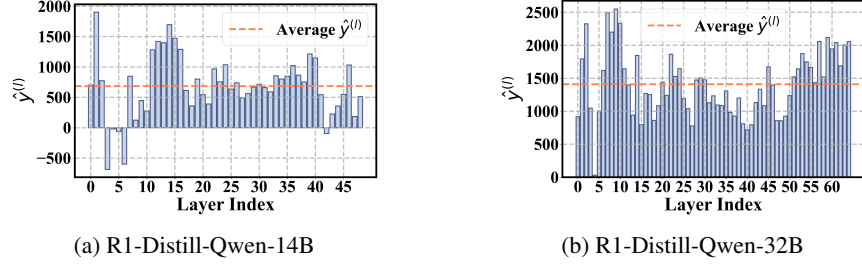


Figure 21: The predicted reasoning number $\hat{y}^{(l)}$ yielded by the pre-allocation vector $\mathbf{r}^{(l)}$. Pre-allocation vector yields positive predictions in most cases.

571 C.4 Pre-allocated Vectors Control Reasoning Strengths by Modifying Logits of `</think>`

572 We provide more results about how these pre-allocated direction vectors control the reasoning strength
 573 by modifying the logits of the end-of-reasoning token `</think>`.

574 We conduct the same activation steering as we do in Section 4.2, varying the steering strength λ
 575 from -0.2 to 0.2. Then, we directly extract the logits of each token at the last token position (*i.e.*,
 576 the start-of-reasoning `<think>`). We visualize the results in Figure 22. We can observe that, as the
 577 steering strength λ increases from -0.2 to 0.2, the logits of the end-of-reasoning token `</think>`
 578 decrease. This indicates that LRMs are less likely to generate such tokens, thereby leading to more
 579 reasoning tokens. Moreover, as shown in Figure 22b, this steering mainly has more impact on the
 580 logits of `</think>` than randomly selected tokens and the EOS token `<endoftext>`. Here, the
 581 random token logits denote the average logits of 500 randomly selected tokens. This indicates that
 582 the steering mainly focuses on adjusting the reasoning strength by manipulating the `</think>`.

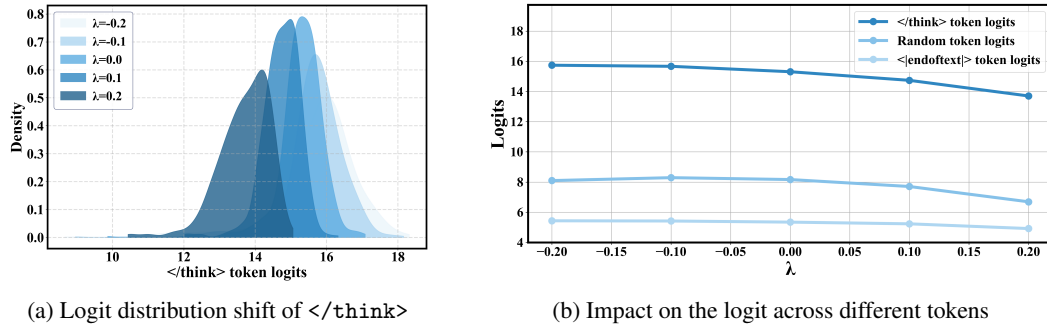


Figure 22: The effect on the end thinking token `</think>` when steering with different strengths on R1-Distill-Qwen-7B. (8a) The pre-allocated vectors control the reasoning strength by causally affecting the logits of end-of-reasoning token `</think>`. (8a) The impact of logits on `</think>` is significantly higher than other tokens.

583 **D Potentials of Our Findings**

584 **D.1 Overthink Detection before Model Generation**

585 In this section, we discuss whether our findings can help us detect the potential overthinking phe-
586 nomenon even before the generation. To study this, we sample 100 questions from the AlpacaEval
587 [50] dataset as the vanilla questions. We then generate one overthink question for each vanilla
588 question, with the overthink attack [48], which proves to be effective in inducing overthink while
589 maintaining the accuracy on vanilla questions. In this way, we can test whether our predictor can
590 detect the overthink phenomenon in advance, by checking whether the predicted token number on
591 overthink questions is much more than that on vanilla questions. Results in Section 5.1 show that our
592 predictor shows potential in overthink detection.

593 **D.2 Efficient Inference**

594 **D.2.1 Details**

595 For evaluation efficiency, we sample 100 questions from MMLU [51] and transform them into
596 single-choice questions for LRMs to answer. We adopt the same setting as in lm-evaluation-harness³
597 for evaluation.

³<https://github.com/EleutherAI/lm-evaluation-harness>

Question: Compute: $1 - 2 + 3 - 4 + 5 - \dots + 99 - 100$.

w/o Steering: <think>Okay, so I have this problem to compute the sum: $1 - 2 + 3 - 4 + 5 - 6 + \dots + 99 - 100$. Hmm, that looks like an alternating series where the signs alternate between positive and negative. Let me try to figure this out step by step. First, let me write out the series to understand the pattern better: $1 - 2 + 3 - 4 + 5 - 6 + \dots + 99 - 100$. I notice that this series alternates between adding an odd number and subtracting the next even number. So, each pair of terms is like $(1 - 2)$, $(3 - 4)$, $(5 - 6)$, and so on. Maybe I can group them into these pairs and see what each pair adds up to. Let me compute a few pairs: $1 - 2 = -1$, $3 - 4 = -1$, $5 - 6 = -1$, $7 - 8 = -1$, ... It seems like each pair of terms adds up to -1 . That's interesting. So, if I can figure out how many such pairs there are, I can multiply that number by -1 to get the total sum. Since the series starts at 1 and ends at 100, that's a total of 100 numbers. If I'm pairing them two by two, the number of pairs should be 100 divided by 2, which is 50. So, there are 50 pairs.

...

That is, for each term, if n is odd, we add it, and if n is even, we subtract it. So, the general term is $(-1)^{(n+1)} * n$. Maybe I can use a formula for such alternating series. I remember that the sum of an alternating series can sometimes be calculated using arithmetic series formulas or by grouping terms. Wait, another approach: Maybe I can separate the series into two separate sums: the sum of the odd terms and the sum of the even terms. Let's see: The series is $1 + 3 + 5 + \dots + 99$ minus $(2 + 4 + 6 + \dots + 100)$. So, if I can compute both of these sums separately, I can subtract them to get the result. First, the sum of the odd numbers from 1 to 99. How many terms are there? Since we're going up to 99, which is the 50th odd number (since $99 = 2*50 - 1$). So, the number of terms is 50.

...

Wait, another way: Let's think about the series as telescoping. But I don't think it's telescoping because there's no cancellation beyond the immediate pairs. Each term cancels only part of the next term, but in this case, each pair is independent. Alternatively, maybe writing the series as: $(1 - 2) + (3 - 4) + \dots + (99 - 100)$. Each parenthesis is -1 , and there are 50 of them. So, $50*(-1) = -50$. Yep, same result. I think I've approached this problem in multiple ways: pairing terms, separating into odd and even sums, using the formula for alternating series, and each time I get -50 . So, it seems pretty solid. Just to recap all the methods: 1. Pairing: Each pair $(1-2)$, $(3-4)$, etc., is -1 , 50 pairs total, sum is -50 . 2. Separating into odds and evens: Sum of odds is 2500, sum of evens is 2550, difference is -50 . 3. Using the formula for the sum of an alternating series of consecutive integers with even terms: $-n/2$, which gives -50 . All methods lead to the same conclusion, so I think it's safe to say the answer is -50 . ****Final Answer**** The value of the series is $\boxed{-50}$. (2253 tokens)

w Steering: <think>First, I observe that the sequence alternates between addition and subtraction of consecutive integers. To simplify the calculation, I can group the terms in pairs: $(1 - 2)$, $(3 - 4)$, $(5 - 6)$, and so on, up to $(99 - 100)$. Each pair results in -1 . Since there are 100 numbers in total, there are 50 such pairs. Multiplying the number of pairs by each result gives 50 times -1 , which equals -50 .</think> (113 tokens)

Figure 23: Case study on R1-Distill-Qwen-32B. The model generates the correct answer both with (i.e., *w*) and without (i.e., *w/o*) steering, but steering significantly reduces the reasoning token number.

Question: What is the correct answer to this question: Kant's humanity formulation of the categorical imperative makes it impermissible for us to, he argues, Choices: (A) sell one of our teeth. (B) engage in prostitution. (C) both A and B (D) Neither A nor B Only put the option letter in the box, e.g. boxedA. There is only one correct answer, .

w/o Steering: <think> Okay, so I need to figure out what Kant's humanity formulation says about selling one of our teeth or engaging in prostitution. Let me start by recalling what the humanity formulation is. From what I remember, Kant's categorical imperative has several formulations, and one of them is the humanity formulation. The key idea here is that we should treat humanity, whether in oneself or in others, always as an end and never merely as a means to an end. So, the question is asking which actions Kant would consider impermissible under this principle. The options are selling one's tooth, prostitution, both, or neither. First, let me break down each option. Selling a tooth: that's essentially selling part of your body. Now, in the context of Kant's ethics, using oneself as a means to an end might be problematic.

...
Wait, but what about the actual examples? I remember that Kant specifically mentions prostitution in his works, right? He considered it immoral because it treats the person as an object for sexual use, which is a means to an end. Similarly, selling body parts might be similar. But selling a tooth is a physical part, whereas prostitution is an action. However, in both cases, the person is using their body for monetary gain, possibly reducing themselves to a means. But maybe Kant would say that selling a tooth is permissible if it's for a good reason, like saving a life, but not sure. Wait, the formulation is about treating humanity as an end. So any action that uses a person (including oneself) merely as a means would be wrong. So selling your tooth would involve using your body as a means to get money, but does that mean you're treating yourself as a mere means? Maybe. Because you're using your body as a commodity, rather than valuing it as an end in itself. Alternatively, maybe Kant would distinguish between selling a tooth (a part) and selling services (like prostitution). But I think both involve commodifying the body. So perhaps both are impermissible. Thus, the answer would be (C) both A and B.

...
Alternatively, maybe he only opposes prostitution, but I think both. Let me think again. Another angle: the humanity formulation requires that in any action, you never use a person merely as a means. So if you sell your tooth, you are using yourself as a means (to get money), but you're also an end. However, if you are selling a tooth, maybe you are treating yourself as an end in some way. Wait, perhaps the distinction is whether you are using your body as a mere means. If you sell a tooth, you are using that part of your body as a tool, but does that mean you are treating your humanity as a means? The humanity is your rational nature. So maybe selling a tooth doesn't directly violate that, as it's a physical part. Alternatively, Kant might argue that the body is part of what makes a person, so using it as a commodity reduces oneself. So perhaps both are wrong. Since I'm a bit torn here, but I think the standard interpretation is that Kant would say both are impermissible. Hence, answer (C). </think>. (1610 tokens)

w Steering: <think> "Okay, so I need to figure out what the answer is here. The question is about Kant's humanity formulation of the categorical imperative and what he says is impermissible. The options are selling a tooth, prostitution, both, or neither. First, I remember that Kant's categorical imperative has different formulations.

...
So both selling a tooth (if it's your own) and prostitution would be impermissible because both involve treating humanity (either your own or someone else's) as a means. So the answer should be C, both A and B? But wait, maybe selling a tooth isn't considered the same as prostitution? Like, a tooth is part of your body but maybe it's not using humanity as an end? Hmm, but Kant might say that even your own body shouldn't be treated merely as a resource to exploit for gain. So selling a tooth could still be wrong because you're using your own body as a means to an end (making money) rather than respecting your own humanity as an end in itself. So both A and B are impermissible. Therefore, the answer should be C. " </think> (418 tokens)

Figure 24: Case study on QwQ-32B. The model generates the correct answer both with (*i.e.*, *w*) and without (*i.e.*, *w/o*) steering, but steering significantly reduces the reasoning token number.

599 **E Broader Impacts**

600 This paper aims to investigate whether LRMs pre-plan their reasoning strength within their activation
601 space, and how such planning is encoded with pre-allocated direction vectors. Our study contributes
602 to a deeper understanding of LRMs within the LLM research community.

603 However, our findings may also pose potential risks. For instance, malicious methods could exploit the
604 discovered property that reasoning length can be manipulated through the model’s internal activations
605 to implant backdoor attacks. Such attacks might trigger excessively long chains of thought under
606 specific conditions, thereby significantly slowing down model execution.