Out of Distribution, Out of Luck: Process Rewards Misguide Reasoning Models

Alexey Dontsov[1,2], Anton Korznikov[1,3], Andrey V. Galichin[1,3,4], Elena Tutubalina[1,2,5]
[1]AIRI, [2]HSE University [3]Skoltech, [4]MTUCI, [5]Sber AI
Moscow, Russia
Correspondence: dontsov@duck.com

Abstract

Process Reward Models (PRMs) have emerged as a promising approach for guiding large language models (LLMs) through multi-step reasoning by providing step-level feedback during inference. However, our evaluation across 7 LLMs reveals a failure mode: while PRMs improve performance for instruct mathematical models, they fail to enhance and sometimes degrade reasoning model performance. Through systematic analysis with linear probes, we identify distinct reward prediction patterns that differentiate reasoning from non-reasoning model outputs. To understand this mechanism, we train Sparse Autoencoders on the Qwen2.5-Math-PRM and analyze reasoning features. Our analysis reveals that 80% of these features respond to formatting artifacts (whitespace patterns, Unicode tokens, punctuation) rather than mathematical content. Reasoning model outputs exhibit distinct metacognitive patterns absent from standard mathematical solutions. This explains why they lead to unreliable reward estimation. Our findings expose a fundamental limitation in applying existing reward models to reasoning systems and provide mechanistic insights into this failure mode. We release our trained SAEs to facilitate future research into reward model interpretability.

1 Introduction

Reasoning remains a central challenge in artificial intelligence, requiring models to perform multi-step inference and deduction (Russell and Norvig, 2020; Bao et al., 2022; Li et al., 2024a). Recently, large language models have demonstrated strong performance on mathematical benchmarks (Hendrycks et al., 2021; Lightman et al., 2024; Shao et al., 2024; Galichin et al., 2026), with Process Reward Models (PRMs) emerging as a promising technique to further enhance performance (Zhang et al., 2025). PRMs provide step-level feedback during reasoning, enabling models to identify and correct errors early.

In parallel, a new generation of reasoning models has emerged (OpenAI et al., 2024; DeepSeek-AI, 2025), which employ extended inference-time computation to tackle challenging problems. These models develop internal verification mechanisms during training through reinforcement learning. This raises a natural question: can PRMs, which explicitly provide step-level guidance, further improve these reasoning models that already perform implicit verification?

We find the answer is no. While PRMs yield substantial improvements for standard mathematical models (up to 75% relative accuracy gain), they consistently fail to enhance reasoning model performance, with degradation of up to 14%, despite requiring additional computational resources. This result suggests an incompatibility between this PRM and reasoning model outputs.

We hypothesize this failure stems from the PRM relying on superficial stylistic features rather than semantic content: reasoning models generate trajectories with distinct formatting patterns absent from standard mathematical outputs. To test this, we employ two mechanistic interpretability approaches. First, we train linear probes at each layer of Qwen2.5-Math-PRM (Zhang et al., 2025) to predict reward values, observing that standard models show gradual improvement across layers while reasoning models exhibit high predictability from early layers. Second, we train Sparse Autoencoders (SAEs) (Huben et al., 2024) on all 28 layers, revealing that reasoning-preferential features predominantly respond to superficial formatting artifacts rather than mathematical content. [1]

Our findings expose fundamental limitations in applying existing reward models to reasoning systems and provide mechanistic insights into this failure mode.

[1]Code available at github.com/somvy/prm_interp

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 2: Short Papers, pages 421–435

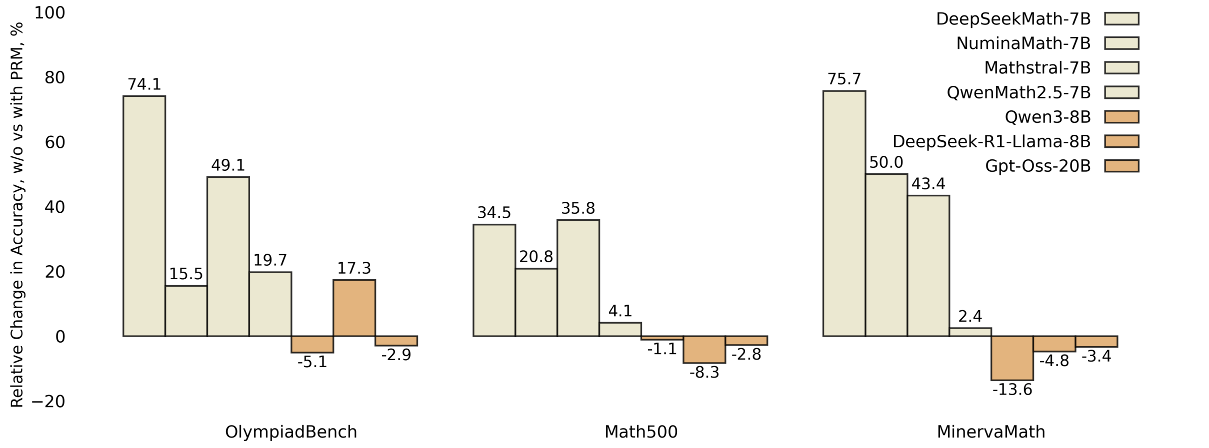

Figure 1: Relative accuracy improvement from PRM-guided inference. Reasoning models (shown in orange) exhibit performance degradation under PRM guidance. Standard mathematical models (shown in white) demonstrate substantial improvements, with some achieving gains exceeding 70%.

2 Related work

Chain-of-thought prompting (Wei et al., 2022) enabled step-by-step reasoning in language models, with recent advances incorporating data augmentation (Yu et al., 2024), hybrid formats (Yue et al., 2024), and reinforcement learning (Luo et al., 2023). PRMs provide step-level feedback, outperforming outcome-only supervision – Lightman et al. (2024) achieved 78.2% on MATH using their PRM800K dataset.

Mechanistic interpretability reverse engineers networks into interpretable algorithms. The superposition hypothesis (Elhage et al., 2022) posits that networks encode more features than dimensions through sparse representations. *Sparse Autoencoders* (SAEs) extract interpretable features from superposition (Huben et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Korznikov et al., 2025), while *linear probes* assess what information representations encode (Alain and Bengio, 2017).

Despite their critical role in RLHF and inference-time scaling, reward model interpretation remains underdeveloped. Marks et al. (2024) first applied SAEs to reward models, identifying unique features in hidden spaces through contrastive analysis of base and RLHF-tuned models. Probing approaches offer complementary insights – Li et al. (2024b) showed that linear functions on embedding spaces can approximate KL-constrained reward maximization.

Building on these interpretability methods, we apply sparse autoencoders and linear probes to PRMs, extending mechanistic interpretability to the verification domain. By identifying which features PRMs use to evaluate reasoning steps, this work bridges interpretation methods and reward modeling – advancing our understanding of neural networks and the engineering of more reliable reasoning systems.

3 Application of PRM for reasoning models

We begin by investigating whether PRMs can be directly applied to improve reasoning models' performance. For this analysis, we use the state-of-the-art QWEN2.5-MATH-PRM (Zhang et al., 2025) as our reward model.

We evaluate 7 models across two categories: Reasoning models: QWEN3-8B (with thinking mode) (Yang et al., 2025), DEEPSEEK-R1-DISTIL-LLAMA-8B (DeepSeek-AI, 2025), and GPT-OSS-20B (OpenAI, 2025), and standard mathematical models: QWEN2.5-MATH-7B (Yang et al., 2024), NUMINAMATH-7B (Beeching et al., 2024), MATHSTRAL-7B (Albert Jiang, 2024), and DEEPSEEKMATH-7B (Shao et al., 2024). Our evaluation spans three benchmarks: MinervaMath (Lewkowycz et al., 2022), MATH500 (Lightman et al., 2024), and OlympiadBench (He et al., 2024).

We apply the reward model straightforwardly: at each reasoning step, we provide the generating model with the question and previous reasoning steps, and then generate 5 candidate continuations. We define a reasoning step as generation until a double newline, following standard practice (Zhang et al., 2025; Cao et al., 2025; Xiong et al., 2025). We score these candidates using the PRM, select the highest-scoring step, and append it to the reasoning trace. Algorithm 1 provides detailed implementation specifics.

Figure 1 presents our results. For standard mathematical models, PRM-guided inference yields substantial accuracy improvements (up to 70%). However, for reasoning models, PRM guidance provides no benefit and even degrades performance, despite requiring additional computational resources for reward modeling. More details in Appendix A.

To understand this failure, we investigate the underlying cause. We hypothesize that reasoning model traces activate distinct feature patterns in the PRM that were not encountered during its training. This shift likely stems from the RL training procedures used for these models, which induce distinct reasoning patterns compared to the data used to train the reward model. Importantly, DEEPSEEK-MATH, which also underwent RL training but without explicit thinking, does not exhibit performance degradation. This suggests that the presence of the thinking regime during training, rather than RL training itself, may be the critical factor causing this shift.

Algorithm 1 PRM Step Selection

Require: Question q, number of candidates K
Ensure: Final reasoning trace S
  Initialize $S \leftarrow [\,]$ (empty reasoning trace)
  for each step i **do**
    Generate candidates:
    $\{s_i^1, \ldots, s_i^K\} \leftarrow \text{Model}(q, S)$
    for $j = 1$ to K **do**
      Score candidates:
      $r_j \leftarrow \text{RewardModel}([q, S, s_i^j])$
    end for
    Add best candidate to the trace:
    $s_i^* \leftarrow \arg\max_j r_j$
    $S \leftarrow S \,||\, [s_i^*]$
  end for

4 Linear probes analysis

To investigate the underlying cause of this phenomenon, we analyze the internal representations of the reward model using linear probing. Our goal is to determine whether the reward model processes reasoning model outputs differently from standard mathematical reasoning traces.

Linear probes provide a direct method for examining whether model representations encode specific information. We construct a dataset of input-output pairs $[(x_1, y_1), ..., (x_n, y_n)]$, where x_i represents a reasoning step and y_i is the corresponding reward score. By extracting internal representations z from different layers and training linear models to predict the rewards y from these representations, we can measure how reward-relevant information develops across the model's depth. This approach has proven effective for understanding model internals in various domains (Belinkov, 2022; Li et al., 2024b).

For our analysis, we sample 12,000 reasoning steps from each model and obtain their PRM rewards. We extract the activations from each layer of the reward model and then split the data into a 5:1 train-test ratio. We standardize the activations to zero mean and unit variance, and train linear regression models to predict rewards. We evaluate probe performance using R^2 scores on the test set.

Figure 2 reveals two distinct behavioral patterns across different model types. For standard mathematical models, R^2 scores begin near zero in early layers and gradually increase, reaching peak performance only in the model's latter half. In contrast, reasoning models exhibit high scores ($R^2 > 0.6$) from the second layer, with continued improvement throughout the network depth.

This difference suggests that the PRM processes reasoning and non-reasoning outputs through different pathways. The high predictability from early layers when processing reasoning traces indicates that the reward model may be relying on superficial features rather than on the deep semantic analysis typically required for mathematical reasoning evaluation. This early saturation of predictive signal indicates that the PRM processes these outputs through shallow pattern matching, leading to poor reward estimation and explaining the lack of performance gains observed in the previous section.

5 SAEs analysis

5.1 Training

The linear probe analysis suggests that the PRM relies on superficial features when processing reasoning outputs, but does not reveal which specific features drive this behavior. To identify these features, we employ Sparse Autoencoders (SAEs) (Huben et al., 2024; Gao et al., 2025) to decompose the model's internal representations into interpretable

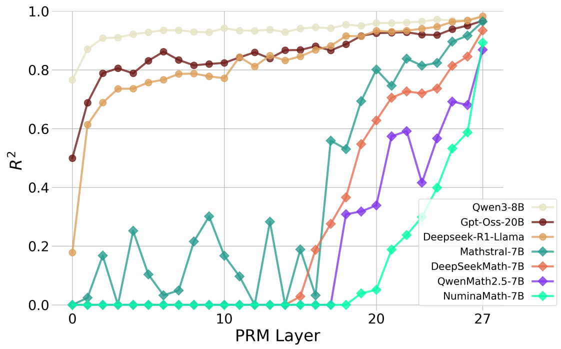

Figure 2: Linear probe performance for predicting PRM rewards across different reward model layers. The x-axis shows PRM layer depth, and the y-axis shows R^2 scores achieved by linear probes trained on activations from each layer. Two distinct patterns emerge: standard mathematical models (diamond markers) show gradual R^2 improvement starting from the model's midpoint, while reasoning models (round markers) achieve high score from early layers and continue improving throughout the network depth.

features. They are designed to reconstruct input activations while enforcing sparsity constraints on the latent representation. An SAE consists of an encoder E and a decoder D. The encoder maps input activations x to a sparse latent representation $y = E(x)$, and the decoder reconstructs the original activations $\hat{x} = D(y)$. The training objective minimizes reconstruction error while promoting sparsity in the latent space. The standard SAE loss function is:

$$L = \|x - \hat{x}\|^2 + \lambda \sum_i |y_i| \tag{1}$$

where λ controls the sparsity.

We employ BatchTopK (Bussmann et al., 2024) to enforce exact sparsity constraints. This technique selects only the top-K activations within each batch and zeros out the remainder, ensuring a fixed sparsity level across all inputs.

We train SAEs on all 28 layers of the Qwen2.5-Math-PRM model. Our training dataset comprises 800 million tokens from QwenMath2.5-Instruct and DeepSeek-R1 generations. We use a latent dimension of 32,768 features with a TopK parameter of 60 active features. The trained SAEs achieve explained variance scores ranging from 0.72 to 0.92 across layers, indicating successful reconstruction (Lieberum et al., 2024). All training details, hyperparameters, and metrics are in Appx. C.

We release the trained SAE weights to facilitate further research into reward model interpretability and mathematical reasoning evaluation.

5.2 SAE Feature Analysis

To identify features that preferentially activate on reasoning model outputs, we analyze SAE activations at layer 15 (54% depth), where semantic concepts typically emerge (Chen et al., 2024; Jin et al., 2025).

We construct comparison pairs by sampling across all combinations of 3 reasoning models , 4 non-reasoning models , and 3 datasets. Each pair compares traces from a reasoning model on one dataset against traces from a non-reasoning model on a potentially different dataset, yielding $3 \times 4 \times 3 \times 3 = 108$ total pairs, allowing cross-dataset comparisons. For each feature f and dataset pair, we compute mean per-token activation $\bar{a}_f$ and define a reasoning preference score:

$$\text{Score}_f = \frac{\bar{a}_f^{(\text{reasoning})}}{\bar{a}_f^{(\text{reasoning})} + \bar{a}_f^{(\text{non-reasoning})} + \epsilon}$$

where scores approaching 1 indicate a strong reasoning preference. Since 99% of the 32,768 SAE features show minimal discriminative power ($\text{Score}_f < 0.1$), we focus on features with $\text{Score}_f > 0.7$.

We randomly partition the 108 pairs into 72 discovery pairs and 36 validation pairs. Candidate features must achieve $\text{Score}_f > 0.7$ in at least 80% of discovery pairs and maintain a mean $\text{Score}_f > 0.7$ across all discovery pairs. We validate candidates using permutation testing with Benjamini-Hochberg false discovery rate correction ($q < 0.05$). Complete methodology is in Appendix D.

5.3 Feature Interpretation

Our analysis identified 50 features that activate on reasoning model traces with statistical significance. These features were validated via permutation testing with Benjamini-Hochberg FDR correction ($q < 0.05$), achieving 89% validation success rate with medium-to-large effect size (Cohen's $d = 0.62$, $p < 0.001$). These features reveal the mechanism driving the reward model's differential processing of reasoning vs non-reasoning outputs.

They fall into distinct categories based on activation patterns. Ten demonstrate interpretable mathematical content detection: three (Features 24993, 5370, 11615) show strong mathematical expression and terminology detection, where Feature 24993 activates strongly on mathematical expressions and equations (e.g., "$n = \underbrace{\ n\ }$", "$f(x) = \frac{3x}{x}$"), Feature 5370 responds to mathematical terminology such as "remainder", "slope", "perimeter," and "numerator", while Feature 11615 detects structural elements of mathematical problem-solving discourse, and seven others respond to mathematical discourse structure. Critically, the remaining 40 (80%) respond to superficial artifacts rather than mathematical content. These features respond to whitespace patterns, special characters, Unicode tokens, and other presentation-level distinctions. Validation across 36 held-out pairs confirms these patterns are statistically significant ($p < 0.001$ after FDR correction) and exhibit medium-to-large effect sizes. Examples of all activation patterns are in Appx. E.

These findings provide direct mechanistic evidence that the PRM relies on superficial stylistic signatures rather than mathematical content evaluation. Rather than evaluating mathematical reasoning, the PRM has primarily learned to detect stylistic signatures that distinguish reasoning model traces from its training distribution. The predominance of artifact-detecting features over content-analyzing features (80% vs 20%) explains why PRM-guided inference fails to improve reasoning model performance despite requiring additional computational resources.

6 Conclusion

We investigate applying a state-of-the-art PRM to reasoning models and identify a critical incompatibility. While it improves standard mathematical models, it fails to enhance reasoning model performance despite additional computational cost.

Through linear probe and SAE analysis, we identify the underlying mechanism. Linear probes reveal high reward predictability from early layers ($R^2 > 0.6$ at layer 2) for reasoning outputs versus gradual emergence for standard outputs, indicating shallow processing. SAE analysis shows that 80% of reasoning-preferential features (40 of 50 validated features) respond to superficial formatting artifacts – whitespace patterns, Unicode tokens, and punctuation – rather than mathematical content. This demonstrates that this PRM has learned to detect stylistic signatures rather than evaluate mathematical validity.

Our findings highlight critical limitations in applying existing evaluation frameworks to reasoning models and provide mechanistic insights into reward model behavior.

7 Limitations

While our work provides mechanistic evidence for incompatibilities between the analyzed PRM (Qwen2.5-Math-PRM) and reasoning models, several limitations warrant discussion. Our analysis focuses on a single state-of-the-art PRM and three reasoning models. Examining additional reward models would strengthen claims about generality. We lack access to the full PRM training distribution, so our claims about distribution mismatch are based on observed behavioral patterns (early-layer reward prediction, superficial feature activation) rather than direct distribution comparisons. Our interpretation of the 50 identified SAE features, while validated through statistical testing and manual inspection of activation patterns, remains subject to the inherent subjectivity of feature interpretation in this emerging research area. We do not extensively explore potential solutions such as retraining PRMs on reasoning outputs or alternative step-selection algorithms, which represent important future directions. Despite these limitations, our work establishes a methodology for investigating reward model behavior using interpretability tools and provides evidence for systematic failure modes that we hope will enable the community to develop more robust evaluation frameworks for reasoning systems.

8 Ethics Statement

This work analyzes existing reward models and reasoning systems without collecting personal data or training new models on sensitive information. No foreseeable negative societal impacts arise from this research, as it focuses on technical analysis of model internals.

AI Assistants: We used Claude Code to help with code and Claude Sonnet for writing assistance. All AI-generated content was carefully reviewed and verified by the authors.

Acknowledgements The work of Elena Tutubalina was supported within the framework of the HSE University Basic Research Program. We acknowledge the computational resources of the HPC facilities at HSE University. We thank Mikhail Seleznev for his valuable feedback on the paper.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

Alexis Tacnet Alok Kothari Antoine Roux Arthur Mensch Audrey Herblin-Stoop Augustin Garreau Austin Birky Bam4d Baptiste Bout Baudouin de Monicault Blanche Savary Carole Rambaud Caroline Feldman Devendra Singh Chaplot Diego de las Casas Eleonore Arcelin Emma Bou Hanna Etienne Metzger Gaspard Blanchet Gianna Lengyel Guillaume Bour Guillaume Lample Harizo Rajaona Henri Roussez Hichem Sattouf Ian Mack Jean-Malo Delignon Jessica Chudnovsky Justus Murke Kartik Khandelwal Lawrence Stewart Louis Martin Louis Ternon Lucile Saulnier Lélio Renard Lavaud Margaret Jennings Marie Pellat Marie Torelli Marie-Anne Lachaux Marjorie Janiewicz Mickaël Seznec Nicolas Schuhl Niklas Muhs Olivier de Garrigues Patrick von Platen Paul Jacob Pauline Buche Pavan Kumar Reddy Perry Savas Pierre Stock Romain Sauvestre Sagar Vaze Sandeep Subramanian Saurabh Garg Sophia Yang Szymon Antoniak Teven Le Scao Thibault Schueller Thibaut Lavril Thomas Wang Théophile Gervet Timothée Lacroix Valera Nemychnikova Wendy Shang William El Sayed William Marshall Albert Jiang, Alexandre Sablayrolles. 2024. Mathstral.

Qiming Bao, Alex Hariri, Yining Gao, Jin Koh, Mimi Mokhtar, Xiongfei Wan, Robert McHardy, Fatemeh Saffari, and Adam Wyner. 2022. Multi-step deductive reasoning over natural language: An empirical study on out-of-distribution generalisation. *arXiv preprint arXiv:2207.14000*.

Edward Beeching, Shengyi Costa Huang, Albert Jiang, Jia Li, Benjamin Lipkin, Zihan Qina, Kashif Rasul, Ziju Shen, Roman Soletskyi, and Lewis Tunstall. 2024. Numinamath 7b cot. `https://huggingface.co/AI-MO/NuminaMath-7B-CoT`.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. *Preprint*, arXiv:2412.06410.

Lang Cao, Yingtian Zou, Chao Peng, Renhong Chen, Wu Ning, and Yitong Li. 2025. Step guided reasoning: Improving mathematical reasoning using guidance generation and step reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21101–21118, Suzhou, China. Association for Computational Linguistics.

Nuo Chen, Ning Wu, Shining Liang, Ming Gong, Linjun Shou, Dongmei Zhang, and Jia Li. 2024. Is bigger and deeper always better? probing llama across scales and layers. *Preprint*, arXiv:2312.04333.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Liao, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Andrey Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets. 2026. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *Preprint*, arXiv:2402.14008.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? *Preprint*, arXiv:2404.07066.

Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Rogov, Elena Tutubalina, and Ivan Oseledets. 2025. Ortsae: Orthogonal sparse autoencoders uncover atomic features. *Preprint*, arXiv:2509.22033.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.

Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Jialian Li and 1 others. 2024a. Boosting deductive reasoning with step signals in rlhf. *arXiv preprint arXiv:2410.09528*.

Kenneth Li, Samy Jelassi, Hugh Zhang, Sham M Kakade, Martin Wattenberg, and David Brandfonbrener. 2024b. Q-probe: A lightweight approach to reward maximization for language models. In *International Conference on Machine Learning*, pages 27955–27968. PMLR.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *Preprint*, arXiv:2408.05147.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Luke Marks, Amir Abdullah, Clement Neo, Rauno Arike, David Krueger, Philip Torr, and Fazl Barez. 2024. Interpreting learned feedback patterns in large language models. *Advances in Neural Information Processing Systems*, 37:36541–36566.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Stuart J Russell and Peter Norvig. 2020. *Artificial Intelligence: A Modern Approach*, 4th edition. Pearson.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, and 1 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. 2025. Mapping the minds of LLMs: A graph-based analysis of reasoning LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17751–17763, Suzhou, China. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning.

A Full evaluation results

For LLM inference, we use recommended sampling parameters (different for each model) and the vLLM (Kwon et al., 2023) engine. For each pair of LLM and PRM the evaluation took in average 30 hours on single Nvidia A100 80 Gb. In total, we spend $30 \times 7 = 210$ GPU-hours on evaluation. Results in Table 2.

B Generation examples

Here we present examples of generations of different models. We highlight the traces of the reasoning models with blue color - DeepSeekR1-Llama at Table 10, Qwen3-8B at Table 9 and Gpt-Oss-20B at Table 8, and of the non-reasoning with green - NuminaMath-7B at the Table 4, DeepSeekMath-7B - Table 5, Mathstral7B - Table 6 and QwenMath2.5-7B - Table 7.

C SAE training details

We train a suite of SAEs on all of the 28 layers of the QwenMath2.5-PRM-7B. The training is performed on 800 million tokens from the combined QwenMath2.5-Instruct and DeepSeek-R1 generations with a context lenght of 1024. We use the AdamW optimizer with learning rate of 7×10^{-5}, $\beta_1 = 0.9, \beta_2 = 0.999$ and the batch size of 16384. We use a latent dimension of 32768 and the TopK parameter of 60. The achieved R^2 (also known as explained variance) metrics are reported in the Table 3. Training was performed on 3 Nvidia A100 GPUs for approximately 100 hours, requiring a total of 300 GPU-hours. We release the weights of these SAEs under the MIT License to open source for advancing the interpretability of the reward models.

D Feature Scoring Methodology

To identify features that preferentially activate on reasoning model outputs, we analyze SAE activations at layer 15 (54% depth), where semantic concepts typically emerge (Chen et al., 2024; Jin et al., 2025).

Experimental Design. We construct comparison pairs by sampling across all combinations of reasoning models (3), non-reasoning models (4), and datasets (3). Each pair compares traces from a reasoning model on one dataset against traces from a non-reasoning model on a potentially different dataset, yielding $3 \times 4 \times 3 \times 3 = 108$ total pairs. This cross-dataset sampling ensures our findings generalize beyond specific dataset characteristics. We randomly partition these into 72 discovery pairs for candidate identification and 36 validation pairs for statistical testing.

Feature Scoring. For each feature f and trace type $D \in \{D_{\text{reasoning}}, D_{\text{non-reasoning}}\}$, we compute mean activation per token across all generated tokens:

$$\bar{a}_f^{(D)} = \frac{1}{|T_D|} \sum_{t \in T_D} a_f(t)$$

where $a_f(t)$ represents feature f's activation at token position t, and T_D denotes all token positions in dataset D. This per-token normalization ensures longer reasoning traces do not artificially inflate activation scores.

We define a reasoning preference score:

$$\text{Score}_f = \frac{\bar{a}_f^{(\text{reasoning})}}{\bar{a}_f^{(\text{reasoning})} + \bar{a}_f^{(\text{non-reasoning})} + \epsilon}$$

where $\epsilon = 10^{-12}$ prevents division by zero when both activations are negligible. Scores approaching 1 indicate strong reasoning preference; scores near 0.5 suggest equal activation across trace types.

Feature Selection. Initial analysis reveals that 99% of the 32,768 SAE features exhibit minimal discriminative power ($\text{Score}_f < 0.1$). We therefore focus on highly discriminative features using an empirically determined threshold $\text{Score}_f > 0.7$, which effectively selects the top 50 most reasoning-preferential features.

Candidate features must satisfy two discovery criteria: (1) $\text{Score}_f > 0.7$ in at least 80% of the 72 discovery pairs, ensuring cross-pair consistency, and (2) mean $\text{Score}_f > 0.7$ across all discovery pairs, ensuring substantial effect magnitude.

Statistical Validation. For each candidate feature, we apply permutation testing on the 36 held-out validation pairs. For each validation pair, we randomly permute the reasoning/non-reasoning labels 1,000 times and compute empirical p-values by comparing observed scores against the null distribution. We apply Benjamini-Hochberg false discovery rate correction ($q < 0.05$) to control for multiple testing across features. Features are retained as robust reasoning indicators only if they maintain $\text{Score}_f > 0.7$ on validation data and achieve statistical significance after correction.

E SAE Feature Activation Patterns

Table 1 presents representative activation patterns for the three interpretable features identified in our analysis. These features demonstrate the reward model's ability to detect legitimate mathematical content including expressions, terminology, and discourse structure.

The remaining 40 features exhibit activation patterns on superficial formatting artifacts rather than mathematical content. Statistical validation across 36 held-out pairs confirms these patterns are statistically significant ($p < 0.001$ after Benjamini-Hochberg FDR correction). Representative examples include:

- Features activating on Unicode characters and special symbols (mathematical symbols, Greek letters, special operators)
- Features responding to specific whitespace patterns and indentation (nested reasoning blocks, paragraph structure)
- Features detecting punctuation sequences and bracket arrangements (parenthetical asides, ellipses for reasoning continuity)
- Features triggered by tokenization artifacts unique to reasoning model outputs (capitalization patterns in metacognitive discourse, formatting from model templates)

These artifact-detecting features suggest that the reward model's reasoning-preferential behavior stems from surface-level pattern recognition rather than semantic evaluation of mathematical reasoning quality. The prevalence of such features (80% of reasoning-preferential features) provides strong evidence that the PRM relies on superficial pattern recognition when processing reasoning model outputs, leading to unreliable reward estimation.

Feature	Activation Patterns
5370	is needed here because the **concentration** is given in atoms per -4)), the **remainder** (R(x)) Therefore, I think the **slope** is -3/2 working in integers, the **remainder** is actually 0? Wait, perhaps the **perimeter** of the buffer is the **, and** ($\sqrt{a^2 + ...}$ $Compute$ **numerator** :(-6 - 4$\sqrt{2}$) **(-**$1)^2 = 3), (m$ $\mathbf{v}^2/c^2 = 0.0$
11615	in which each term after the first is obtained from **the** The smallest distance between **the** origin and a point on We can add **the** first and third equations to unit squares. At **the** midpoints of some of sit in a row with **the** two boys sitting next to **for** all y. Since $f(y) \geq 9$ in logarithmic form. Taking **the** logarithm (base think of **the** following: If Sergey asks for subsets :****-** We need (n) **!** =6, which matches. So, in that
24993	defined as n **=** $\underbrace{n}$ If $f(x)$**=**$\frac{3x}{x}$ $(x + 1)$ **=** x^2- 1^2 **=**1 $\frac{12}{2}$ **=**6 $5, b$ **=**$16/5$ $+bc + bd$ **=**2023 $(x^2$ **=**$y^{-3/4}$ $0^2)$ **=**10 Therefore So, in n **=**2, the PQR**=** $\frac{\sqrt{3}}{}$ **fraction** of turns. The gardener can increase at most **,** it's 90°. Therefore, triangle **are** in A_2. So, **set**. However, if I can have multiple terms divisible

Table 1: Activation patterns of the found features. We highlight the strongest feature activation with bold text.

Model	**OlympiadBench**		**Math500**		**MinervaMath**	
	Pass@1/2/4	**PRM**	**Pass@1/2/4**	**PRM**	**Pass@1/2/4**	**PRM**
QwenMath2.5-7B	0.30/0.36/0.42	0.364	0.66/0.70/0.72	0.688	0.24/0.29/0.33	0.251
DeepSeek-R1-Llama-8B	0.24/0.28/0.32	0.415	0.54/0.60/0.65	0.494	0.15/0.19/0.24	0.140
Qwen3-8B	0.51/0.54/0.56	0.484	0.74/0.75/0.75	0.728	0.34/0.37/0.39	0.298
Gpt-Oss-20B	0.47/0.52/0.55	0.461	0.72/0.76/0.78	0.700	0.15/0.19/0.21	0.143
Mathstral-7B	0.17/0.24/0.31	0.246	0.40/0.50/0.58	0.546	0.11/0.17/0.22	0.162
NuminaMath-7B	0.19/0.26/0.33	0.216	0.43/0.52/0.60	0.522	0.10/0.14/0.20	0.147
DeepSeekMath-7B	0.11/0.16/0.22	0.188	0.36/0.44/0.53	0.480	0.11/0.16/0.20	0.195

Table 2: Model Performance Comparison Across Benchmarks

Layer number	Explained Variance
0	0.79
1	0.89
2	0.91
3	0.91
4	0.89
5	0.87
6	0.86
7	0.85
8	0.82
9	0.82
10	0.82
11	0.80
12	0.79
13	0.77
14	0.76
15	0.73
16	0.72
17	0.72
18	0.72
19	0.70
20	0.72
21	0.75
22	0.75
23	0.75
24	0.75
25	0.78
26	0.85
27	0.91

Table 3: The explained variances of SAEs, trained on the QwenMath2.5-PRM.

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation:

1. Recognize that the series $1-2+3-4+5-\cdots+99-100$ can be rearranged into pairs of terms, each pair being the sum of two consecutive numbers, and then add the last number, 100.

2. Group the terms in pairs:

$$(1-2)+(3-4)+(5-6)+\cdots+(99-100)$$

3. Each pair sums to -1:

$$1-2=-1, \quad 3-4=-1, \quad \ldots, \quad 99-100=-1$$

4. Since there are 50 pairs (from 1 to 100, there are 100 numbers, and each pair consists of 2 numbers), the sum of all pairs is:

$$-1\times 50=-50$$

5. Add the last number, 100, to the sum of the pairs:

$$-50+100=50$$

6. Thus, the sum of the series $1-2+3-4+5-\cdots+99-100$ is $\boxed{50}$.

Table 4: An example of solution of the task from Math500 generated by NuminaMath-7B

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation: The given expression is an arithmetic series with a common difference of -1 or 1. We can split this series into two separate series: one for the odd numbers and one for the even numbers.
The odd numbers in the series are $1,3,5,\ldots,99$. This is an arithmetic series with first term $a_1=1$, common difference $d=2$, and last term $a_{50}=99$. We can find the sum of this series using the formula for the sum of an arithmetic series, which is $S_n=\frac{n}{2}(a_1+a_n)$, where n is the number of terms.
The number of terms in this series is $n=\frac{99-1}{2}+1=50$. So, the sum of the odd numbers is $S_{50}=\frac{50}{2}(1+99)=25\times100=2500$.
The even numbers in the series are $2,4,6,\ldots,100$. This is also an arithmetic series, but with first term $a_1=2$, common difference $d=2$, and last term $a_{50}=100$. The number of terms in this series is also $n=50$. So, the sum of the even numbers is $S_{50}=\frac{50}{2}(2+100)=25\times102=2550$.
Finally, we can find the sum of the entire series by subtracting the sum of the even numbers from the sum of the odd numbers: $2500-2550=-50$.
So, the answer is $\boxed{-50}$.

Table 5: An example of solution of the task from Math500 generated by DeepSeekMath-7B

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation:
Sure, let's break this down step by step. This is an alternating series of integers from 1 to 100. We can group the terms in pairs to simplify the calculation:
$(1-2)+(3-4)+(5-6)+\ldots+(99-100)$
Each pair of terms in the parentheses simplifies to -1. There are 50 pairs in total (since we're pairing 100 numbers), so we have:
$-1+(-1)+(-1)+\ldots+(-1)$ (50 times)
This simplifies to: $-1*50=-50$ So, the final answer is $\boxed{-50}$.

Table 6: An example of solution of the task from Math500 generated by Mathstral-7B

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation:
To compute the sum $1-2+3-4+5-\cdots+99-100$, we can observe that the series alternates between positive and negative terms. Let's break it down step by step. First, we can group the terms in pairs:

$$(1-2)+(3-4)+(5-6)+\cdots+(99-100)$$

Each pair sums to:

$$\begin{gathered}1-2=-1\\3-4=-1\\5-6=-1\\\vdots\\99-100=-1\end{gathered}$$

Next, we need to determine how many such pairs there are. Since the series starts at 1 and ends at 100, and each pair consists of 2 terms, we can find the number of pairs by dividing the total number of terms by 2:

$$\frac{100}{2}=50$$

Now, we multiply the number of pairs by the sum of each pair:

$$50\times(-1)=-50$$

Therefore, the sum of the series $1-2+3-4+5-\cdots+99-100$ is:

$$\boxed{-50}$$

.

Table 7: An example of solution of the task from Math500 generated by QwenMath2.5-Instruct

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation: analysis

We need compute alternating sum from 1 to 100: $\sum_{k=1}^{100}(-1)^{k+1}k$. That is $1-2+3-4+...+99-100$. Let's compute pairwise: $(1-2)=-1,(3-4)=-1,...,(99-100)=-1$.
There are 100 terms, 50 pairs.
Each pair gives -1. So total = $50*(-1)=-50$.
Alternatively compute formula: $\sum_{k=1}^{100}(-1)^{k+1}k=-50$.
So answer: -50. Provide in boxed

$$
\begin{aligned}
&1-2+3-4+\cdots+99-100\\
&=(1-2)+(3-4)+\cdots+(99-100)\\
&=(-1)+(-1)+\cdots+(-1)\quad\text{(50 terms)}\\
&=50(-1)=-50.
\end{aligned}
$$

$\boxed{-50}$

Table 8: An example of solution of the task from Math500 generated by Gpt-Oss-20B

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation: <think> Okay, so I need to compute this series: 1 - 2 + 3 - 4 + 5 - ... + 99 - 100. Let me think about how to approach this. First, let me write down the series to visualize it better. It starts with 1, then subtracts 2, adds 3, subtracts 4, and so on, alternating between adding and subtracting each subsequent integer. The last term is -100. So the pattern is +1 -2 +3 -4 +5 -6 ... +99 -100. Hmm, since the signs are alternating, maybe I can group the terms in pairs? Let me try that. If I group them as (1 - 2) + (3 - 4) + (5 - 6) + ... + (99 - 100). Each pair seems to follow the same pattern. Let me check the first few pairs: 1 - 2 = -1 3 - 4 = -1 5 - 6 = -1 ... 99 - 100 = -1 So each pair of terms adds up to -1. Now, how many such pairs are there in the entire series? The series starts at 1 and goes up to 100. Since each pair consists of two numbers, the total number of pairs would be 100 divided by 2, which is 50. So there are 50 pairs. If each pair is -1, then the total sum would be 50 multiplied by -1. Let me compute that: 50 * (-1) = -50. Wait, is that correct? Let me verify with a smaller example. Let's take the first 4 terms: 1 - 2 + 3 - 4. According to my method, grouping them as (1 - 2) + (3 - 4) = (-1) + (-1) = -2. If I compute directly: 1 - 2 + 3 - 4 = (1 - 2) + (3 - 4) = -1 + (-1) = -2. That matches. Another example: first 2 terms: 1 - 2 = -1. According to the formula, number of pairs is 1, so 1*(-1) = -1. Correct. How about first 6 terms: 1 - 2 + 3 - 4 + 5 - 6. Grouping as (1 - 2) + (3 - 4) + (5 - 6) = (-1) + (-1) + (-1) = -3. Which is 3 pairs, so 3*(-1) = -3. Correct again. So seems like my approach is working. Therefore, applying this to the original problem with 100 terms, which is 50 pairs, each summing to -1, total sum is -50. But let me think again. Is there another way to approach this problem? Maybe using formulas for arithmetic series? The original series is an alternating series. Let me recall that for an alternating series, sometimes you can separate the positive and negative terms.

...

skipped ≈ 800 tokens

...

Alternatively, think of the entire series as: $\sum_{k=1}^{100}(-1)^{k+1}k$ But maybe using generating functions or other methods? Probably overkill. Alternatively, consider pairing terms as I did first. But all methods lead to -50. Therefore, I think the answer is definitely -50. Wait, let me check once again with actual calculation for a slightly bigger number. Let me take n=10. Compute 1 -2 +3 -4 +5 -6 +7 -8 +9 -10. Grouping as (1-2)+(3-4)+(5-6)+(7-8)+(9-10) = (-1)*5 = -5. Which is -10/2 = -5. Correct. Alternatively, compute directly: 1 -2 = -1; -1 +3 = 2; 2 -4 = -2; -2 +5 = 3; 3 -6 = -3; -3 +7 = 4; 4 -8 = -4; -4 +9 = 5; 5 -10 = -5. Yep, same result. So all checks out. Therefore, I can be confident that the answer is -50. **Final Answer** The value of the series is $\boxed{-50}$. </think>

To compute the value of the alternating series: $1-2+3-4+5-6+\cdots+99-100$ we can approach this by recognizing the structure of the sequence and grouping terms in a way that simplifies the computation. ---

Step 1: Grouping Terms We notice that the series alternates between positive and negative integers. That is: $(1-2)+(3-4)+(5-6)+\cdots+(99-100)$ Each pair of terms follows the same pattern: $1-2=-1$, $3-4=-1$, $5-6=-1$, ... So, every pair contributes a value of -1. ---

Step 2: Counting the Pairs. The series consists of 100 terms, and since each pair consists of two terms, the total number of pairs is: $\frac{100}{2}=50$ Thus, the total sum is: $50\times(-1)=-50$ ---

Step 3: Confirming with an Alternative Approach. We can also consider the series as the sum of two separate sequences: - **Positive terms (odd numbers):** $1+3+5+\cdots+99$ - **Negative terms (even numbers):** $-2-4-6-\cdots-100$ The sum of the first n odd numbers is given by: Sum of odd numbers $=n^2$ There are 50 odd numbers from 1 to 99, so: Sum of positive terms $=50^2=2500$ The sum of the first n even numbers is: Sum of even numbers $=n(n+1)$ There are 50 even numbers from 2 to 100, so: Sum of even numbers $=50\times 51=2550$ Then the total sum is: Total sum $=2500-2550=-50$ ---

Step 4: General Formula. For a general alternating sum of the form: $\sum_{k=1}^{n}(-1)^{k+1}k$ - If n is even, the sum is $-\frac{n}{2}$ - If n is odd, the sum is $\frac{n+1}{2}$ Here, $n=100$ (even), so:

$$\text{Sum}=-\frac{100}{2}=-50$$

Final Answer $\boxed{-50}$

Table 9: An example of solution of the task from Math500 generated by Qwen3-8B

Problem: Compute $1-2+3-4+5-\cdots+99-100$.

Generation: Okay, so I have this problem to solve: Compute the sum of the series 1 - 2 + 3 - 4 + 5 - 6 + ... + 99 - 100. Hmm, let me think about how to approach this.
First, I need to understand the pattern of the series. It starts at 1, then subtracts 2, adds 3, subtracts 4, and so on, alternating between addition and subtraction. Each pair of numbers seems to follow the pattern: positive odd number minus the next even number. So, it's like (1 - 2) + (3 - 4) + (5 - 6) + ... + (99 - 100).
Let me write this out for a few more terms to see if the pattern continues. The first pair is 1 - 2 = -1, the next is 3 - 4 = -1, then 5 - 6 = -1, and this should continue all the way up to 99 - 100. So, each of these pairs gives me -1.
Now, how many such pairs are there? Well, starting from 1 to 100, there are 100 numbers in total. Since each pair consists of two numbers, the number of pairs is 100 divided by 2, which is 50. So, there are 50 pairs, each adding up to -1.
Therefore, if each pair is -1 and there are 50 pairs, the total sum should be 50 multiplied by -1. Let me compute that: 50 * (-1) = -50. Wait, let me double-check that. Maybe I made a mistake in the number of pairs. So, starting from 1, each pair is (1,2), (3,4), (5,6), ..., (99,100). Each of these is indeed two numbers, so 100 numbers make 50 pairs. That seems right.
Alternatively, I can think of it as grouping the terms as (1 - 2) + (3 - 4) + ... + (99 - 100). Each of these is -1, and since there are 50 such terms, the sum is 50*(-1) = -50.
Is there another way to think about this? Maybe using the formula for arithmetic series? Let me recall that the sum of an arithmetic series is given by n/2*(first term + last term). But wait, in this case, the signs are alternating. So, is this series arithmetic?
Wait, no, it's not exactly arithmetic because the signs alternate. So, it's not a straightforward arithmetic series. It's an alternating series where the signs alternate between positive and negative. So, perhaps I can't apply the regular arithmetic series formula here.
Alternatively, maybe I can express this series as the sum of two separate series: the sum of the positive terms and the sum of the negative terms. Let me try that. The positive terms are 1, 3, 5, ..., 99. How many terms are there in this positive series? Since we're adding all odd numbers from 1 to 99, the number of terms is (99 - 1)/2 + 1 = 50 terms. So, the sum of positive terms is the sum of the first 50 odd numbers.
Wait, the sum of the first n odd numbers is known to be n^2. So, the sum of the first 50 odd numbers is $50^2 = 2500$. Okay, that seems right.
Now, the negative terms are -2, -4, -6, ..., -100. This is an arithmetic series as well, where each term is negative and even. The number of terms here is 50 because we're taking the even numbers from 2 to 100, which is 50 terms. The sum of an arithmetic series is (n/2)*(first term + last term). So, the sum of the negative terms is (50/2)*(-2 + (-100)) = 25*(-102) = -2550.
Alternatively, maybe I can think...
...
(skipped ≈ 1000 tokens)
...
Alternatively, maybe I can use the formula for the sum of an alternating series. Wait, but that's more for infinite series, and this is a finite one.
I recall that for the sum from k=1 to n of $(-1)^{k+1} * k$, the formula is $(-1)^{n+1} * (n+1)/2$.
Wait, is that correct? Let me check for small n. For n=1: sum=1. The formula would be $(-1)^2 * (2)/2 = 1 * 1 = 1$. Correct.
n=2: 1 -2= -1. Formula: $(-1)^3 * (3)/2 = -3/2$. That's not equal to -1. So, that can't be the formula.
Wait, maybe the formula is different. Let me think.
Wait, actually, the sum $S = 1-2+3-4+5-6+...+(-1)^{n+1} * n$.
There's a formula for this, but I think it's a bit more complicated. For even n, the sum is -n/2, and for odd n, it's (n+1)/2. Let me test that.
For n=1: S=1. Formula for odd n: (1+1)/2=1. Correct.
n=2: 1 -2= -1. Formula for even n: -2/2= -1. Correct.
n=3: 1 -2 +3=2. Formula for odd n: (3+1)/2=2. Correct.
n=4: 1 -2 +3 -4= -2. Formula for even n: -4/2= -2. Correct.
So, yes, the formula seems to be:
If n is even, S = -n/2.
If n is odd, S = (n+1)/2.
So, in this case, n=100, which is even. So, the sum should be -100/2= -50. Therefore, the sum is -50. That's another way to see it, using the formula for the alternating series of integers. So, that confirms the result again. Wait, so n=100, so S= -50.
So, several different methods lead to the same result, so I'm pretty confident that the answer is -50. **Final Answer** The sum is $\boxed{-50}$. </think>
To compute the sum of the series $1-2+3-4+5-6+\cdots+99-100$, we observe that the series can be grouped into pairs: $(1-2)+(3-4)+(5-6)+\cdots+(99-100)$ Each pair equals -1. There are 50 such pairs because there are 100 terms in total, and each pair consists of two terms. Therefore, the sum of the series is: $50\times(-1)=-50$ We can also verify this by considering the series as the sum of two separate series: the sum of the positive terms and the sum of the negative terms.
- The positive terms are $1,3,5,\ldots,99$, which are the first 50 odd numbers. The sum of the first n odd numbers is n^2, so the sum is $50^2=2500$. - The negative terms are $-2,-4,-6,\ldots,-100$, which form an arithmetic series. The sum of the first n even numbers is $n(n+1)$, so the sum is $50\times 51=2550$. Since these are negative, the sum is -2550.Adding these together: $2500+(-2550)=-50$ Using the formula for the sum of an alternating series of integers, we find that for $n=100$ (which is even), the sum is $-n/2=-50$.
Thus, the final answer is $\boxed{-50}$.

Table 10: An example of solution of the task from Math500 generated by DeepSeekR1-Distil-Llama-7B