Probing the Origins of Reasoning Performance: Representational Quality for Mathematical Problem-Solving in RL vs. SFT Fine-Tuned Models

Antyabha Rahman[1], **Akshaj Gurugubelli**[2], **Omar Ankit**[3], **Kevin Zhu**[2], **Aishwarya Balwani**[4]

[1]University of New South Wales, [2]Algoverse AI Research, [3]University of Waterloo, [4]St. Jude Children's Research Hospital

Abstract

Large reasoning models trained via reinforcement learning (RL) have been increasingly shown to outperform their supervised fine-tuned (SFT) counterparts on mathematical reasoning tasks; Yet the mechanistic basis for this advantage remains unclear. We therefore ask, *what internal representational differences enable RL models' superior performance?* Our work presents two converging lines of evidence: First, linear probes trained on layer-wise hidden states reveal that RL models tend to achieve higher accuracy in predicting answer correctness compared to SFT models, indicating more linearly separable and structured representations. Second, mean ablation studies show that RL models develop a hierarchical architecture where deeper layers become progressively more critical, whereas SFT models distribute importance uniformly across layers. Together, these findings demonstrate that RL training fundamentally restructures how models represent and process reasoning problems. Finally, we analyze token-count variability under repeated sampling across problems to assess adaptive compute allocation. While we observe higher variability in some RL-tuned models than in their SFT counterparts, we see strong consistency in others, suggesting that token allocation may depend more on the overall training pipeline than on RL versus SFT alone. We believe this token-allocation variability reveals the spread of plausible on-policy reasoning, highlighting which models exhibit stable policies versus those that are under-determined, potentially non-identifiable solution behaviour.

Code — https://oankit.github.io/-rl-sft-reasoning/

Introduction

Large Reasoning Models (LRMs) such as OpenAI's o1 and DeepSeek-R1 substantially outperform traditionally fine-tuned large language models (LLMs) on reasoning and logic problems across benchmarks (OpenAI et al. 2024; Shao et al. 2024). Understanding *why* this is the case though, requires moving beyond performance metrics to mechanistic explanations. While we know LRMs generate longer chains of thought and achieve higher accuracy, *how* they differ internally from base LLMs remains an open challenge.

Current research has approached this question from two mutually reinforcing but disconnected perspectives. **Mechanistic interpretability** has identified specific circuits for arithmetic operations (Sachan, Stolfo, and Sun 2025; Hanna, Liu, and Variengien 2023; Zhu, Dai, and Sui 2024) and shown that chain-of-thought increases activation sparsity (Chen, Plaat, and van Stein 2025), primarily in smaller models on elementary operations. **Behavioural studies** have revealed information-theoretic compression limits (Lee, Che, and Peng 2025), cross-variant sensitivity to problem phrasing (Mirzadeh et al. 2025), and RL training's potential for long reasoning (Yeo et al. 2025). However, *what internal representational differences* enable LRMs' superior performance remains unexplored.

We seek to bridge this gap through integrated behavioural-mechanistic analysis using three complementary methods: (1) *Linear probing* on layer-wise hidden states to predict answer correctness (Alain and Bengio 2018; Belinkov 2022), measuring *when* and *how strongly* representations emerge across model families. (2) *Mean ablation interventions* (Zhang and Nanda 2024; Meng et al. 2023) to identify which layers are critical for mathematical reasoning across training methodologies. (3) *Generation consistency analysis* via multiple samples per problem, extending variance analysis (Mirzadeh et al. 2025) to within-problem comparisons and empirically validating compression theory (Lee, Che, and Peng 2025).

Specifically, our contributions include: (1) Evidence that RL models develop stronger, earlier-emerging representations through scalable linear probing. (2) Discovery that RL training reshapes computational architecture, concentrating reasoning in deeper layers versus instruction-tuning's uniform distribution. (3) Empirical validation of token complexity theory, revealing that superior representations manifest as consistent token usage across difficulty levels, with current RL training showing unexploited potential for adaptive allocation. Our analysis reveals that **training methodology fundamentally reshapes computational architecture**: RL-trained models show earlier engagement and progressive concentration in deeper layers (r=0.47), whereas instruction-tuned models distribute reasoning uniformly (r=-0.11). This architectural difference, combined with earlier-emerging and stronger answer representations in RL models, provides initial mechanistic insight into performance differences, moving from descriptive benchmarking (OpenAI et al. 2024; DeepSeek-AI 2025) to mechanistic explanation.

Measuring Representation Quality via Probing

Understanding *when* and *how* correct answer information emerges across model layers can reveal fundamental differences between LRMs and SFT models. If LRMs develop "clearer" representations, we should be able to detect this mechanistically: problems with more linearly separable internal representations should exhibit higher discriminability between correct and incorrect answers. To test this, we train linear probes on layer-wise hidden states to predict final answer correctness, building on the interpretability literature using linear probes to study intermediate representations (Alain and Bengio 2018), particularly for mathematical reasoning (Zhu, Dai, and Sui 2024). We hypothesize that probe accuracy correlates with model accuracy, potentially explaining why LRMs outperform SFT models.

Synthetic Problem Generation

To investigate whether model failures stem from reasoning limitations or surface-form artifacts, we generated 1,000 synthetic mathematical problems using four fixed templates covering probability, fractions, and cost calculations. Following Mirzadeh et al. (2025)'s approach of generating synthetic GSM8K variants (Cobbe et al. 2021), our controlled generation isolates representational properties required for general reasoning from question-specific memorization effects. Template details and example problems are provided in Appendix .

Rationale: Synthetic generation offers three advantages: (1) eliminates data contamination, (2) enables algorithmic verification with known ground-truth parameters, and (3) scales to large sample sizes for robust statistical analysis. Prior work demonstrates that synthetic benchmarks effectively reveal model reasoning capabilities (Mirzadeh et al. 2025) while avoiding artifacts of human-authored datasets.

Completion Generation and Labeling For each model, we generate a single completion for every problem using sampling (temperature $T \in [0.6, 0.7]$, *top p* $= 0.95$). Final answers are extracted from within `\boxed{}` delimiters. Each completion is labeled as correct if the extracted answer matches the ground truth (allowing a tolerance of ± 1 for rounding), and incorrect otherwise. Problems where no valid answer can be extracted are omitted from further analysis.

Balancing procedure: To ensure fair comparison across models with different accuracy distributions, we identify the intersection of problems answered by all models, then sample equal numbers of correct and incorrect examples per model. This produces balanced training sets where each model contributes identical sample sizes with equal class distribution, removing confounding variables. Data is split 70/15/15 into train/validation/test sets, stratified by label.

Activation Extraction Activation extraction captures the model's internal state at the precise moment it has completed reasoning but not yet committed to an answer.

Probe position: We extract hidden states at the token immediately preceding `\boxed{}`. This position is chosen for two reasons: (1) all answers follow this delimiter, ensuring the model has completed reasoning before answer articulation, and (2) the delimiter tokenizes consistently across examples, unlike final answers which may span variable token lengths. We tokenize the full generated text and locate `\boxed{` in the token sequence.

Batched extraction with position preservation: We process completions in batches of 16–32. To maintain consistent token positions across variable-length sequences, we apply *right padding* (padding appended after sequences) rather than left padding, ensuring extraction positions remain unchanged relative to sequence start. Each sequence receives an attention mask with 1s for real tokens and 0s for padding.

Layer-wise representations: For each batch, we perform a single forward pass with `output_hidden_states=True` and extract activations from all L transformer layers. We collect `hidden_states[1:L+1]`—the outputs of transformer blocks 1 through L—omitting `hidden_states[0]` (input embeddings) and architecture-specific post-normalization states that may reduce probe effectiveness. This yields tensors of shape $[L \times N \times D]$, where N is batch size and D is hidden dimension (4096).

Probe Training For each layer $\ell \in \{1, \ldots, L\}$, we train logistic regression probe $f_\ell : \mathbb{R}^D \to \{0, 1\}$ to predict binary answer correctness from the D-dimensional hidden state $\mathbf{h}_\ell$. The probe is defined as:

$$f_\ell(\mathbf{h}_\ell) = \sigma(\mathbf{w}_\ell^\top \mathbf{h}_\ell + b_\ell) \tag{1}$$

where $\mathbf{w}_\ell \in \mathbb{R}^D$ is the weight vector, $b_\ell \in \mathbb{R}$ is the bias term, and $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. Following best practices (Belinkov 2022), we use 5-fold cross-validation to select regularization strength C from $\{0.001, 0.01, 0.1, 1.0, 10.0\}$, with `class_weight='balanced'` to handle residual class imbalance.

We report test set accuracy as the primary metric. Our methodology assumes the linear representation hypothesis (Zhu, Dai, and Sui 2024): if correct answer information exists in a layer's representations, a simple linear classifier should reliably extract it. Higher probe accuracy indicates more robust linear separability between correct and incorrect answer representations.

Results: Layer-Wise Probe Accuracy

Motivation. We hypothesize that the improved mathematical reasoning ability of reasoning-capable models on mathematical benchmarks stems from developing *clearer* internal representations of correctness—i.e., representations that are more linearly separable and consistently structured across samples. To test this, we train linear probes to classify correct vs. incorrect solutions at each layer, using probe accuracy as a proxy for representation clarity.

Reasoning Models Exhibit Representation Clarity

Figure 1 reveals substantial differences in how correctness information is encoded across model types. Models trained with reinforcement learning from verifiable rewards—DeepSeek-Math-7B-RL and Olmo-3-Think—achieve

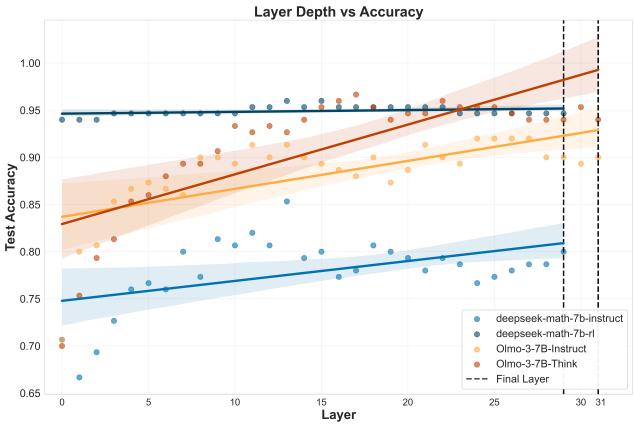


Figure 1: Layer-wise probe accuracy for predicting answer correctness across model families. Reasoning models (DeepSeek-Math-7B-RL, Olmo-3-Think) achieve higher probe accuracy (83–98%) and earlier emergence compared to base and instruction-tuned models (DeepSeek-Math-Instruct, Olmo-3-Instruct) (75–90%). Notable late-layer regression appears in final layers for all models.

markedly higher probe accuracy (83–98%) compared to instruction-tuned models (75–90%). This 8 percentage point gap, combined with noticeably reduced variance between individual samples (tighter scatter in Figure 1), suggests that RL and chain-of-thought training could lead to representations with linear separability between correct and incorrect answer states (Zhang et al. 2025; Park, Choe, and Veitch 2024).

Immediate emergence in reasoning models. Reasoning-capable models exhibit remarkably high probe accuracy from the very first layer, with test accuracies of 70% at layer 0 compared to 65% for Instruct models. Both Olmo-3-Think and DeepSeek-Math-7B-RL maintain $\sim$94–95% accuracy throughout layers 15–29. This immediate availability of correctness information contrasts sharply with instruction-tuned models, where probe accuracy gradually improves from $\sim$65% to $\sim$66–80% and 84–90% for DeepSeek-Math-Instruct and Olmo-3-Instruct respectively over 28 layers. Defining an **emergence layer** ℓ_{emerge} as the first layer achieving $> 80\%$ test accuracy, we find $\ell_{\text{emerge}} = 0$ and 2 for DeepSeek-Math-RL and Olmo-3-Think respectively, and $\ell_{\text{emerge}} = 6$ and 1 for DeepSeek-Math-Instruct and Olmo-3-Instruct respectively. The threshold for emergence layer must be dynamically defined based on accuracy across all models, as some thresholds are crossed by all models from layer 0. Despite similar emergence layers for the Olmo-3 family, the Think model achieves higher overall accuracy.

Pre-training and training objectives matter. The Olmo-3-Instruct model, fine-tuned from the base model using the Dolci Instruct SFT dataset, outperforms DeepSeek-Math-7B-Instruct. This demonstrates that a more robust and diverse dataset for pre-training and SFT can still provide significant performance increases, which is supported by our probe findings where the accuracy for Olmo-3-Instruct is significantly higher (Mosbach et al. 2020; Zhou and Srikumar 2022). DeepSeek-Math-7B-RL and Olmo-3-Think substantially outperform their instruction-tuned counterparts, demonstrating that reasoning-specific training objectives—not merely scale or general fine-tuning—are key to developing clear correctness representations.

Representation Clarity as a Mechanism for Better Mathematical Reasoning

Our results provide a potential mechanistic explanation for why RL-trained and chain-of-thought models outperform instruction-tuned models on mathematical benchmarks: they develop fundamentally clearer internal representations of correctness. The high probe accuracy (75–95%) indicates that these models encode correctness as robust, linearly-separable features (Park, Choe, and Veitch 2024; Jiang et al. 2024). This clarity likely enables more reliable access to correctness signals during autoregressive generation, leading to more consistent correct outputs.

Caveats. Our probing methodology has an important limitation: it requires the model to produce a sufficiently balanced distribution of correct and incorrect answers. When the correct-to-incorrect ratio deviates substantially from 50:50, the probe risks learning to predict the majority class rather than genuinely detecting representational differences. This dependency means our findings are most reliable for models operating near their capability boundaries, where both outcomes occur with reasonable frequency.

Layer-Wise Mean Ablations

We aim to investigate the criticality each layer has upon the mathematical reasoning capabilities of DeepSeek-Math models through systematic activation patching. Our methodology employs mean ablation interventions to replace layer activations with their corresponding mean values computed from a reference dataset (GSM8K training data). This approach follows the established activation patching protocols introduced by Zhang and Nanda (2023).

Experimental Setup

We evaluate the `DeepSeek-Math-7B-Instruct` and `DeepSeek-Math-7B-RL` models on 20 GSM8K problems per model. For each layer $\ell \in \{0, 1, \ldots, L-1\}$, we replace the activation h_ℓ with its corresponding reference mean activation μ_ℓ and measure the resulting degradation in accuracy.

Evaluation Metric

Accuracy Drop (AD): This metric quantifies the change in model accuracy relative to the baseline performance. For each layer ℓ, we compute: $\text{AD}_\ell = \text{Acc}_{\text{base}} - \text{Acc}_\ell^{\text{abl}}$, where Acc_{base} is the baseline accuracy (without ablation), and $\text{Acc}_\ell^{\text{abl}}$ is the accuracy measured when the activations at layer ℓ are replaced by μ_ℓ. Larger values of AD_ℓ indicate higher importance of that layer in mathematical reasoning.

Pearson Correlation Coefficient (r): This statistic measures the linear correlation between layer depth and accuracy drop. A positive r indicates that deeper layers are more

critical to performance, while a value near zero implies that importance is distributed evenly across the network.

Implementation

We extract final answers using pattern matching on the `\boxed{...}` notation. All generations use fixed decoding parameters (temperature = 0.1, top_p = 0.9). Prompts enforce structured, step-by-step reasoning to ensure that mathematical problem-solving processes are made explicit.

Results and Analysis

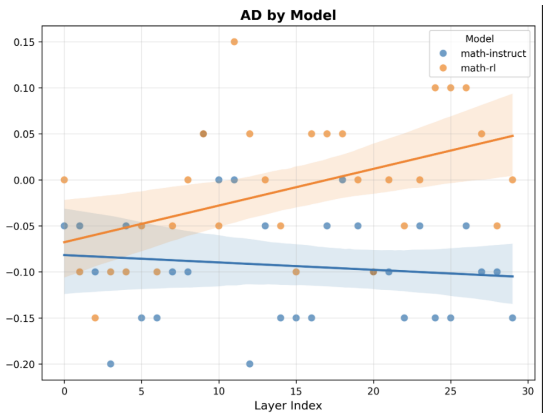


Figure 2: Accuracy Drop (AD) across layers for DeepSeek-Math-7B-Instruct and DeepSeek-Math-7B-RL.

Layer Criticality Patterns

As shown in Figure 2, the two DeepSeek variants exhibit distinct computational architectures. `DeepSeek-Math-7B-RL` (baseline acc. 70%) exhibits a significant positive correlation between layer depth and intervention impact ($r = 0.47$, $p < 0.01$), with AD ranging from -0.15 to $+0.15$. This indicates that deeper layers become increasingly critical for mathematical reasoning. In contrast, `DeepSeek-Math-7B-Instruct` (baseline accuracy: 65%) demonstrates a weak negative correlation ($r = -0.11$, $p = 0.55$), with AD ranging from -0.20 to $+0.05$, suggesting relatively flat layer importance with slight emphasis on early layers.

Layer Criticality Interpretation

These contrasting patterns reflect distinct computational strategies shaped by training objectives. The RL-trained model shows higher early-layer impact combined with progressive deepening, indicating hierarchical reasoning architecture with concentration in layers 9-18 and 22-26. Conversely, the instruction-tuned model displays a distributed reasoning profile, suggesting that supervision across full reasoning trajectories encourages balanced layer utilization and introduces redundancy that enhances robustness to perturbations.

Convergence and Divergence Points

Both models exhibit similar vulnerability across layers 0–10 (AD $\approx$ -0.15 to 0.00), indicating shared foundational mechanisms likely responsible for arithmetic operations and core reasoning primitives. Beyond layer 15, however, their trajectories diverge sharply, demonstrating that training methodology fundamentally reshapes higher-order mathematical reasoning. This divergence has implications for performance optimization and failure mode identification.

Token Variability in Mathematical Problem-Solving

We investigate whether the representational differences observed in our previous findings manifest in downstream behaviours such as token variability between RL and SFT models.

Experimental Setup

Models: We evaluate four models spanning two architectural families:

DeepSeekMath family: `DeepSeekMath-Instruct`, `DeepSeekMath-RL` (Shao et al. 2024)

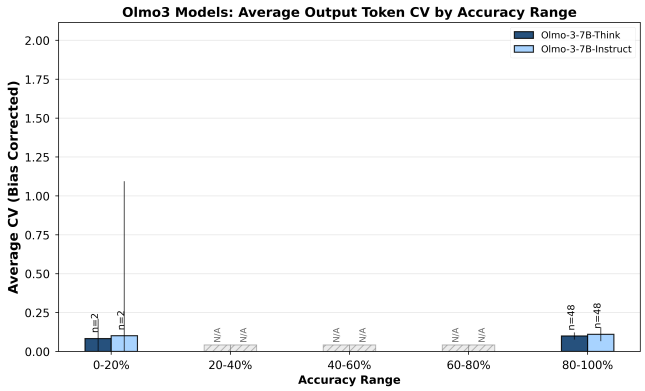
Olmo 3 family: `Olmo-3-Instruct`, `Olmo-3-Thinking` (Olmo et al. 2025).

Data and Methodology. We evaluate on 50 problems from GSM8K-Platinum (Vendrow et al. 2025), generating 50 independent responses per problem per model (15,000 responses per model). We measure answer correctness, input tokens, and output tokens (including reasoning tokens for LRMs). Full experimental details are provided in Appendix .

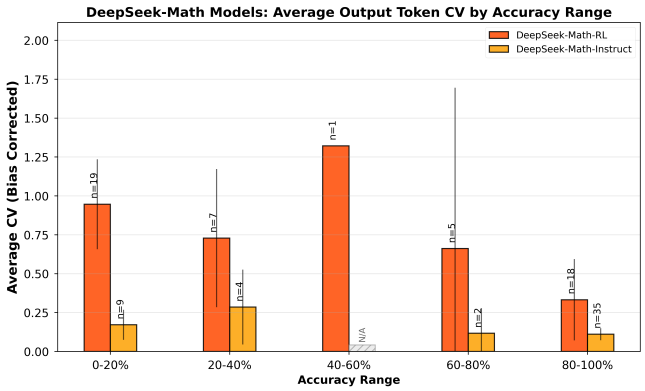
Evaluation Metrics. For each problem, we compute: (1) *answer consistency*, the proportion of runs producing correct answers; (2) *token coefficient of variation (CV)*, computed across the 50 responses per problem as $\text{CV} = \sigma_{\text{tokens}}/\mu_{\text{tokens}}$, where σ_{tokens} and μ_{tokens} are the standard deviation and mean of output token counts respectively. We use CV rather than raw standard deviation to enable fair comparison across model families with different baseline output lengths (LRMs typically generate 5–10$\times$ more tokens than SFT models). Since all responses contain at least hundreds of tokens, the mean is never near zero, avoiding the instability that CV exhibits when $\mu \to 0$; and (3) *median output tokens*, including reasoning tokens for LRMs.

Results: Divergent Variability Patterns

Model families exhibit fundamentally different variability profiles. Figure 3b reveals that DeepSeek-Math-RL exhibits high variability across difficulty levels, peaking in the 40–60% accuracy region (CV $\approx$ 1.3) and remaining elevated even at capability boundaries (CV $\approx$ 0.95 at 0–20%, CV $\approx$ 0.35 at 80–100%). DeepSeek-Math-Instruct shows consistently lower variability across all bins. In contrast, Figure 3a shows both Olmo-3-Thinking and Olmo-3-Instruct maintain remarkably consistent low variability (CV $=$ 0.1–0.125) in the bins where data exists (0–20%



(a) Olmo 3 models



(b) DeepSeek-Math models

Figure 3: Token coefficient of variation by accuracy range across model families. (a) Olmo-3-Thinking and Olmo-3-Instruct maintain consistent low variability (CV $=$ 0.1–0.125) across all bins where data exists. (b) DeepSeek-Math models exhibit decreasing variability, with the highest variability observed in the hardest-difficulty region (0–20%) and decreasing to the lowest at 80–100%, demonstrating compression inefficiency at capability boundaries.

and 80–100%), suggesting qualitatively different generation strategies.

Empirical evidence for compression theory. Our findings partially support Lee, Che, and Peng (2025)'s token complexity framework. DeepSeek-Math-RL exhibits high variability at capability boundaries (CV $>$ 0.8), meaning identical problems elicit vastly different response lengths—precisely the calibration failure predicted by information-theoretic analysis. Even at high accuracy where models should reliably compress, non-trivial variability persists (CV $\approx$ 0.35), indicating systematic deviation from optimal compression. However, the Olmo-3 family demonstrates that consistent low variability is achievable across accuracy extremes.

RL training shows variable impact on adaptive allocation. Contrary to expectations, DeepSeek-Math-RL exhibits substantially *higher* variability than DeepSeek-Math-Instruct across all accuracy bins, suggesting that RL training in this case amplified rather than reduced output inconsistency. In contrast, both Olmo-3-Thinking and Olmo-3-Instruct maintain nearly identical low-variability profiles. This divergence suggests that the relationship between training methodology and token allocation consistency is model-dependent, likely influenced by the specific training pipeline and reward structure rather than RL versus SFT alone.

Conclusion

We investigated the mechanistic basis of reinforcement learning's success in mathematical reasoning through integrated behavioural-mechanistic analysis. Our findings reveal a coherent picture: RL-trained models develop superior representations that emerge earlier in the network; Specifically, in the DeepSeek model family, RL-trained models develop representations that are more linearly separable and emerge earlier while Olmo models show a similar pattern for representation quality although with less pronounced differences in emergence timing. Linear probing shows RL models achieve higher accuracy in predicting answer correctness, with representations emerging in earlier layers than in SFT models, while ablation studies confirm these representations are functionally critical. Token variability analysis reveals model-dependent patterns: while Olmo-3 models maintain consistent generation across difficulty levels regardless of training method, DeepSeek-Math-RL exhibits higher variability than its SFT counterpart—suggesting that the relationship between RL training and output consistency depends on the specific training pipeline.

Multiple promising avenues emerge from this work. Designing reward structures that explicitly incentivize adaptive token allocation could better exploit RL's potential. Our layer-wise analysis focused on answer correctness; extending probing to intermediate reasoning steps could reveal how multi-step solutions are constructed and validated. Scaling this analysis to larger models and diverse reasoning domains (code generation, scientific reasoning) would test whether our findings generalize beyond mathematical problem-solving. More broadly, developing methods to detect and measure representation quality during training could enable real-time assessment of model reliability – a critical need for deploying reasoning models in high-stakes domains.

References

Alain, G.; and Bengio, Y. 2018. Understanding intermediate layers using linear classifier probes. arXiv:1610.01644.

Belinkov, Y. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1): 207–219.

Chen, X.; Plaat, A.; and van Stein, N. 2025. How does Chain of Thought Think? Mechanistic Interpretability of Chain-of-Thought Reasoning with Sparse Autoencoding. arXiv:2507.22928.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Hanna, M.; Liu, O.; and Variengien, A. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. arXiv:2305.00586.

Jiang, Y.; Rajendran, G.; Ravikumar, P.; Aragam, B.; and Veitch, V. 2024. On the Origins of Linear Representations in Large Language Models. arXiv:2403.03867.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Lee, A.; Che, E.; and Peng, T. 2025. How Well do LLMs Compress Their Own Chain-of-Thought? A Token Complexity Approach. arXiv:2503.01141.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2023. Locating and Editing Factual Associations in GPT. arXiv:2202.05262.

Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2025. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. arXiv:2410.05229.

Mosbach, M.; Khokhlova, A.; Hedderich, M. A.; and Klakow, D. 2020. On the Interplay Between Fine-tuning and Sentence-Level Probing for Linguistic Knowledge in Pre-Trained Transformers. In *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Olmo, T.; :; Ettinger, A.; Bertsch, A.; Kuehl, B.; Graham, D.; Heineman, D.; Groeneveld, D.; Brahman, F.; Timbers, F.; Ivison, H.; Morrison, J.; Poznanski, J.; Lo, K.; Soldaini, L.; Jordan, M.; Chen, M.; Noukhovitch, M.; Lambert, N.; Walsh, P.; Dasigi, P.; Berry, R.; Malik, S.; Shah, S.; Geng, S.; Arora, S.; Gupta, S.; Anderson, T.; Xiao, T.; Murray, T.; Romero, T.; Graf, V.; Asai, A.; Bhagia, A.; Wettig, A.; Liu, A.; Rangapur, A.; Anastasiades, C.; Huang, C.; Schwenk, D.; Trivedi, H.; Magnusson, I.; Lochner, J.; Liu, J.; Miranda, L. J. V.; Sap, M.; Morgan, M.; Schmitz, M.; Guerquin, M.; Wilson, M.; Huff, R.; Bras, R. L.; Xin, R.; Shao, R.; Skjonsberg, S.; Shen, S. Z.; Li, S. S.; Wilde, T.; Pyatkin, V.; Merrill, W.; Chang, Y.; Gu, Y.; Zeng, Z.; Sabharwal, A.; Zettlemoyer, L.; Koh, P. W.; Farhadi, A.; Smith, N. A.; and Hajishirzi, H. 2025. Olmo 3. arXiv:2512.13961.

OpenAI; :; Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; Iftimie, A.; Karpenko, A.; Passos, A. T.; Neitz, A.; Prokofiev, A.; Wei, A.; Tam, A.; Bennett, A.; Kumar, A.; Saraiva, A.; Vallone, A.; Duberstein, A.; Kondrich, A.; Mishchenko, A.; Applebaum, A.; Jiang, A.; Nair, A.; Zoph, B.; Ghorbani, B.; Rossen, B.; Sokolowsky, B.; Barak, B.; McGrew, B.; Minaiev, B.; Hao, B.; Baker, B.; Houghton, B.; McKinzie, B.; Eastman, B.; Lugaresi, C.; Bassin, C.; Hudson, C.; Li, C. M.; de Bourcy, C.; Voss, C.; Shen, C.; Zhang, C.; Koch, C.; Orsinger, C.; Hesse, C.; Fischer, C.; Chan, C.; Roberts, D.; Kappler, D.; Levy, D.; Selsam, D.; Dohan, D.; Farhi, D.; Mely, D.; Robinson, D.; Tsipras, D.; Li, D.; Oprica, D.; Freeman, E.; Zhang, E.; Wong, E.; Proehl, E.; Cheung, E.; Mitchell, E.; Wallace, E.; Ritter, E.; Mays, E.; Wang, F.; Such, F. P.; Raso, F.; Leoni, F.; Tsimpourlas, F.; Song, F.; von Lohmann, F.; Sulit, F.; Salmon, G.; Parascandolo, G.; Chabot, G.; Zhao, G.; Brockman, G.; Leclerc, G.; Salman, H.; Bao, H.; Sheng, H.; Andrin, H.; Bagherinezhad, H.; Ren, H.; Lightman, H.; Chung, H. W.; Kivlichan, I.; O'Connell, I.; Osband, I.; Gilaberte, I. C.; Akkaya, I.; Kostrikov, I.; Sutskever, I.; Kofman, I.; Pachocki, J.; Lennon, J.; Wei, J.; Harb, J.; Twore, J.; Feng, J.; Yu, J.; Weng, J.; Tang, J.; Yu, J.; Candela, J. Q.; Palermo, J.; Parish, J.; Heidecke, J.; Hallman, J.; Rizzo, J.; Gordon, J.; Uesato, J.; Ward, J.; Huizinga, J.; Wang, J.; Chen, K.; Xiao, K.; Singhal, K.; Nguyen, K.; Cobbe, K.; Shi, K.; Wood, K.; Rimbach, K.; Gu-Lemberg, K.; Liu, K.; Lu, K.; Stone, K.; Yu, K.; Ahmad, L.; Yang, L.; Liu, L.; Maksin, L.; Ho, L.; Fedus, L.; Weng, L.; Li, L.; McCallum, L.; Held, L.; Kuhn, L.; Kondraciuk, L.; Kaiser, L.; Metz, L.; Boyd, M.; Trebacz, M.; Joglekar, M.; Chen, M.; Tintor, M.; Meyer, M.; Jones, M.; Kaufer, M.; Schwarzer, M.; Shah, M.; Yatbaz, M.; Guan, M. Y.; Xu, M.; Yan, M.; Glaese, M.; Chen, M.; Lampe, M.; Malek, M.; Wang, M.; Fradin, M.; McClay, M.; Pavlov, M.; Wang, M.; Wang, M.; Murati, M.; Bavarian, M.; Rohaninejad, M.; McAleese, N.; Chowdhury, N.; Chowdhury, N.; Ryder, N.; Tezak, N.; Brown, N.; Nachum, O.; Boiko, O.; Murk, O.; Watkins, O.; Chao, P.; Ashbourne, P.; Izmailov, P.; Zhokhov, P.; Dias, R.; Arora, R.; Lin, R.; Lopes, R. G.; Gaon, R.; Miyara, R.; Leike, R.; Hwang, R.; Garg, R.; Brown, R.; James, R.; Shu, R.; Cheu, R.; Greene, R.; Jain, S.; Altman, S.; Toizer, S.; Toyer, S.; Miserendino, S.; Agarwal, S.; Hernandez, S.; Baker, S.; McKinney, S.; Yan, S.; Zhao, S.; Hu, S.; Santurkar, S.; Chaudhuri, S. R.; Zhang, S.; Fu, S.; Papay, S.; Lin, S.; Balaji, S.; Sanjeev, S.; Sidor, S.; Broda, T.; Clark, A.; Wang, T.; Gordon, T.; Sanders, T.; Patwardhan, T.; Sottiaux, T.; Degry, T.; Dimson, T.; Zheng, T.; Garipov, T.; Stasi, T.; Bansal, T.; Creech, T.; Peterson, T.; Eloundou, T.; Qi, V.; Kosaraju, V.; Monaco, V.; Pong, V.; Fomenko, V.; Zheng, W.; Zhou, W.; McCabe, W.; Zaremba, W.; Dubois, Y.; Lu, Y.; Chen, Y.; Cha, Y.; Bai, Y.; He, Y.; Zhang, Y.; Wang, Y.; Shao, Z.; and Li, Z. 2024. OpenAI o1 System Card. arXiv:2412.16720.

Park, K.; Choe, Y. J.; and Veitch, V. 2024. The Linear Repre-

sentation Hypothesis and the Geometry of Large Language Models. arXiv:2311.03658.

Sachan, M.; Stolfo, A.; and Sun, Y. 2025. Probing for Arithmetic Errors in Language Models. arXiv:2507.12379.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Zhang, M.; Li, Y.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.

Vendrow, J.; Vendrow, E.; Beery, S.; and Madry, A. 2025. Do Large Language Model Benchmarks Test Reliability? arXiv:2502.03461.

Yeo, E.; Tong, Y.; Niu, M.; Neubig, G.; and Yue, X. 2025. Demystifying Long Chain-of-Thought Reasoning in LLMs. arXiv:2502.03373.

Zhang, A.; Chen, Y.; Pan, J.; Zhao, C.; Panda, A.; Li, J.; and He, H. 2025. Reasoning Models Know When They're Right: Probing Hidden States for Self-Verification. arXiv:2504.05419.

Zhang, F.; and Nanda, N. 2024. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. arXiv:2309.16042.

Zhou, Y.; and Srikumar, V. 2022. A Closer Look at How Fine-tuning Changes BERT. arXiv:2106.14282.

Zhu, F.; Dai, D.; and Sui, Z. 2024. Language Models Encode the Value of Numbers Linearly. arXiv:2401.03735.

Additional Token Variability Experimental Details

Dataset: 50 problems randomly sampled from the GSM8K-Platinum test split (Vendrow et al. 2025) (seed=42).

Generation parameters:

- *DeepSeekMath-Instruct*: Temperature $T = 0.6$, Max tokens = 4096
- *DeepSeekMath-RL*: Temperature $T = 0.6$, Max tokens = 4096
- *Olmo-3-Instruct*: Temperature $T = 0.6$, Top-p $p = 0.95$, Max tokens = 32768
- *Olmo-3-Think*: Temperature $T = 0.6$, Top-p $p = 0.95$, Max tokens = 32768

Model Inference: We use vLLM (Kwon et al. 2023) on a single GH200 GPU for efficient execution.

Prompt Template:

```
System: Please reason step by step, and
put your final answer within \boxed{}.
User: {question}
```

Additional Details for Linear Probing

Problem templates: Each template instantiates a word problem with randomized numerical parameters while maintaining fixed logical structure:

1. *Conditional probability*: Calculate probability of turning in homework given sequential conditional events (substitute teacher, class extension, personal extension). Requires probability multiplication and complementary probability computation. Answers range from 8–50%.
2. *Student demographics*: Given total students, age threshold, and gender ratios stratified by age group, compute total female students. Requires division, fraction multiplication, and subtraction. Answers range from 220–3,986 students.
3. *Sequential growth*: Given initial water flow and multiplicative/additive growth rules over days, compute final quantity. Requires tracking state across time steps with doubling and addition. Answers range from 7,057–25,513 gallons.
4. *Counting with unit conversion*: Track brownies (in dozens) received and consumed across multiple events, then convert to individual items. Requires dozen-to-unit conversion, fraction addition/subtraction, and summation. Answers range from 1–231 brownies.
5. *Cost calculation*: Given base price and dependent pricing rules (e.g., "leather seats cost one-third of the king cab upgrade"), compute total cost. Requires chained fraction operations and summation. Answers range from $\$34,490$–$\$66,846$.

Each problem includes the instruction: "Please reason step by step, and put your final answer within `\boxed{answer}` as an integer." Answers range from 100 to 10,000, ensuring consistent numerical magnitude. The following are the graphs for the question each of the question types

B.7 Per-Template Probe Accuracy Results

The following figures show layer-wise probe accuracy for each of the five synthetic problem templates. Each figure displays test set accuracy across all transformer layers for the four models evaluated.

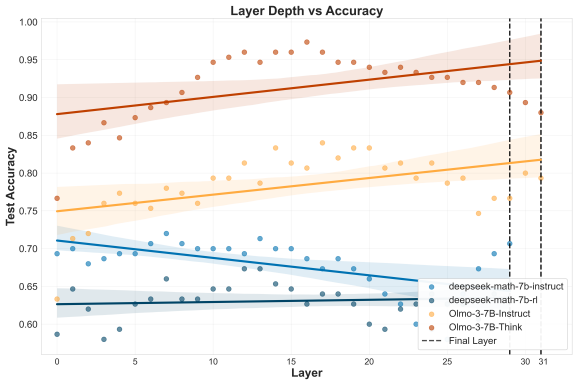


Figure 4: **Probing classification accuracy vs. transformer layer.** *Cost Calculation Problem*: DeepSeek-Math-7B-RL, DeepSeek-Math-7B-Instruct, Olmo-3-Think, Olmo-3-Instruct.

Sample Questions from each of the categories:

1. *Conditional probability*: `Yasmine is trying to decide whether they really need to do their homework. There's a 70% chance that tomorrow they'll`

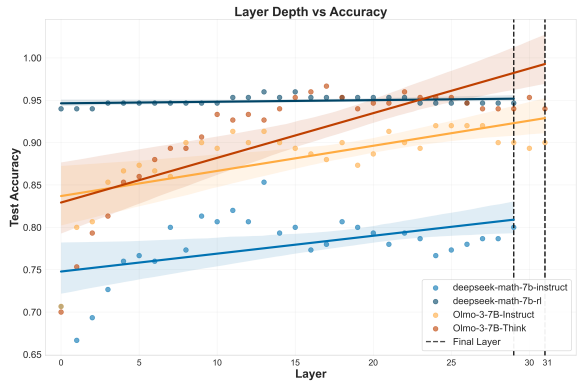


Figure 5: **Probing classification accuracy vs. transformer layer.** *Student Demographics Problem*: DeepSeek-Math-7B-RL, DeepSeek-Math-7B-Instruct, Olmo-3-Think, Olmo-3-Instruct

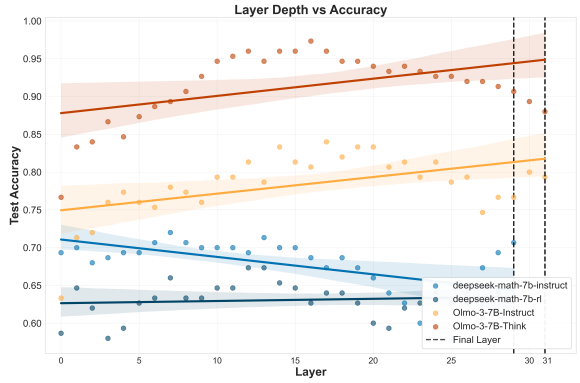


Figure 6: **Probing classification accuracy vs. transformer layer.** *Sequential Growth Problem*: DeepSeek-Math-7B-RL, DeepSeek-Math-7B-Instruct, Olmo-3-Think, Olmo-3-Instruct

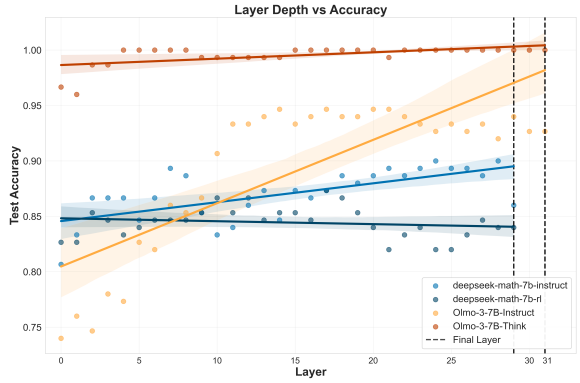


Figure 7: **Probing classification accuracy vs. transformer layer.** *Conditional Probability Problem*: DeepSeek-Math-7B-RL, DeepSeek-Math-7B-Instruct, Olmo-3-Think, Olmo-3-Instruct

have a substitute teacher who won't collect the homework. Even if the normal teacher comes in, there's a 60% chance she'll give everyone an extension. Even if the whole class doesn't get an extension, there's a 25% chance Yasmine can convince the teacher their dog ate their assignment and get a personal extension. What is the percentage chance that Yasmine will actually have to turn in their homework tomorrow? *(Answer: 9%)*

2. *Student demographics*: Brook Hills High School currently enrolls 4,374 students. Half of these students are over 18 years old, and one-fifth of the students over 18 years old are male. The remaining half of the students are under 18 years old, and 2/5 of the students under 18 are male. In total, how many female students are enrolled at this school? *(Answer: 3,062 students)*
3. *Sequential growth*: The amount of water passing through a river at one point in time is 5,904 gallons. After a day of heavy rain, the amount of water passing through the river doubles at the same point. If the volume of water passing through the river at that point increases by 7,202 gallons on the third day, calculate the total amount of water passing through the river at that point. *(Answer: 19,010 gallons)*
4. *Counting with unit conversion*: Quentin wanted brownies for her birthday. She made a batch for herself; nine dozen Nut Brownies. At her office, they threw her a party and sent her home with 9/10 dozen brownies. When she arrived home, her friends were there to throw her a surprise party and had 4 dozen brownies waiting. During the party, 2 2/10 dozen brownies were eaten. How many individual brownies did Quentin have left over from the entire day? *(Answer: 140 brownies)*
5. *Cost calculation*: Bill is ordering a new truck. He has decided to purchase a two-ton truck with several added features: a king cab upgrade, a towing package, leather seats, running boards, and the upgraded exterior light package. The base price of the truck is $42,572, and the other features are at extra cost. The king cab is an extra $6,890,

leather seats are one-third the cost of the king cab upgrade, running boards are $500 less than the leather seats, and the upgraded exterior light package is $1,724. What is the total cost of Bill's new truck, in dollars? *(Answer: $55,278)*

Reproducibility: Code and data is provided in the github link