Probing Internal Signals of Topic, Difficulty, and Prediction of Success in a 7B Math LLM

Authors:
Hamza Iqbal — *hiqbal@wustl.edu*
Hailey Montgomery — *h.e.montgomery@wustl.edu*

1. Goal & Motivation

We will evaluate a single open-weight 7B math-specialized LLM on competition-style problems and train linear probes on frozen activations to test three questions:
(O2) *Topic probe:* is the problem topic (algebra/number theory/combinatorics) linearly decodable after reading the question?
(O3) *Success probe:* can a probe predict eventual correctness from the model's state at 0% (question-only) and 50% (mid-solution replay)? (we may extend this to 25% and 75% progress points as well if time permits)
(O4) *Difficulty probe:* do activations encode human difficulty (MATH levels 1–5), and how does that align with actual accuracy?
These interpretability checks connect to evidence that simple linear directions can track truthfulness [1] and in-advance correctness [2], and extend them to math reasoning where models have shown 'cliffs' of performance within certain problem types.

Importance: If an LLM internally "knows" topic, senses hardness, or foresees failure mid-solution, we can design routing/guardrails (e.g., self-consistency, fallback to tools, human-in-the-loop) to reduce wrong-but-confident answers, most crucially within areas where failure and uncertainty is high.

2. Related Work

Linear probes & emergent linear structure. Linear classifier probes [5] reveal what information is linearly decodable at each layer. Recent work shows linear structure for truth vs. falsehood in LLM residual streams and provides causal evidence via interventions [1]. Closest to our success probe, question-only linear probes can predict whether the forthcoming answer will be correct, with signals peaking at mid layers, but **generalization is weaker on math, motivating our domain-specific study** [2].

Math benchmarks & inference strategies. The MATH benchmark by Hendrycks provides 12.5k competition problems with topic tags and difficulty levels (1–5), plus solutions [3]. Inference-time self-consistency (sample multiple chains, then vote) reliably boosts math accuracy and motivates our multi-sample evaluation [4].

Models & tooling. We will select an open 7B model tuned for math (e.g., DeepSeekMath‑7B, Mathstral‑7B) that cleanly exposes hidden states for probing [6–7], and use TransformerLens/HF hooks to capture activations [8]. Note: initial testing with determine our model of choice here.

3. Method

(Objective 1) Performance matrix by topic × difficulty (How well does the model perform at different topics of different difficulties?)

- **Dataset:** Stratified MATH subset (≈400–600 problems) balancing topics and difficulty (1–5). We will exclude geometry items, given issues with describing such problems to text only models.
- **Decoding:** Given resource constraints we propose using 5 samples/problem, but may adjust this if variance is high. Our prompt enforces a standard final answer format (e.g., \boxed {answer}) so an additional evaluation model or scheme is not necessary.
- **Outputs:** Accuracy vs. difficulty and topic, and identification of steep drops ("cliffs").

(Objective 2) Topic probe (Given the question and before any output, do the hidden states of the model tell us it knows what type of problem is being asked?)

- **Inputs:** Hidden states after encoding the question (no generated tokens).
- **Features:** Start with final token residual per layer (or mean‑pooled question tokens).
- **Layer sweep:** Evaluate every 4th layer first, then zoom into the best region.

(Objective 3) Success probe [0%,50%] (How well can the model before answering as well as halfway through answering, predict its correctness?)

- **Labeling:** Generate once to obtain an answer and ground‑truth correct/incorrect.
- **Checkpoints (replayed back after initial generation):**
  - **0%:** question only.
  - **50%:** question + first half of the *already generated* tokens.
- **Probe:** Logistic regression (balanced classes; split by problem). This tests whether internal states anticipate success and whether the signal strengthens mid‑solution [2].

(Objective 4) Difficulty probe (question-only, may exclude if time and resources are insufficient)

- Predict human difficulty (regression or 5-way classification) from question-only states; correlate with actual accuracy. Interesting cases: predicted “easy” yet wrong (overconfidence) vs. “hard” yet right (good, may provide predictive power that model is nearing its capability limit)

Considerations:

- **VRAM limits:** We’re aiming to run on a 16 GB card in FP16/BF16 (but may quantize if ending up with insufficient space). In addition, to reduce VRAM needs we will restrict context (≤1.5k tokens).
- **Data hygiene:** Train/val/test split by problem, we will balance classes within our selected subset of the MATH dataset, and we will store seeds and prompts for reproducibility

4. Proposed Timeline (each stage is 2 week block)

- **Stage I**: Setup and decide model and activation-capture path; implement answer parser; 20-problem mini run to ensure feasibility of different probes and constructing of the topic x difficulty table.

- **Stage II**: Run the stratified evaluation set with multi-sample decoding; produce accurate heatmaps and locate cliffs (steep dropoffs in performance)

- **Stage III**: Train/evaluate topic and difficulty, (if time permits) probes with layer sweeps; report best layers and generalization.

- **Stage IV**: Build (0%/50%) replay dataset; train the success probe; analyze where the signal peaks across layers and checkpoints.

- **Stage V**: Relate probe signals to observed cliffs (e.g., does the success probe drop precisely where accuracy falls?); finalize figures, write-up, and ablations.

5. Expected Outcomes & Risks

Outcomes. (i) Clear accuracy vs. difficulty/topic curves with at least one cliff; (ii) Topic probe above chance with a peak layer; (iii) Success probe improving from 0%→50% checkpoints (or a principled negative result on math, extending [2]); (iv) Optional difficulty probe correlations.

Risks & mitigations.

- **Mid-solution capture complexity:** Avoid streaming; use replay to fixed checkpoints.

- **Answer parsing fragility:** Enforce a single boxed answer format in the prompt; report both strict and lightly normalized metrics.
- **Diagram‑dependent geometry:** We are not planning on using geometry within our testing, as evaluating text-based models on this may lead to more of an issue with problem 'understanding' rather than a model's ability.

References

[1] **Marks, S., Tegmark, M.** *The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets.* arXiv:2310.06824, 2023. (arXiv)

[2] **Vicente Moreno Cencerrado, I., Padrés Masdemont, A., Gonzalvez Hawthorne, A., Africa, D. D., Pacchiardi, L.** *No Answer Needed: Predicting LLM Answer Accuracy from Question‑Only Linear Probes.* arXiv:2509.10625, 2025. (arXiv)

[3] **Hendrycks, D.,** et al. *Measuring Mathematical Problem Solving With the MATH Dataset.* arXiv:2103.03874; NeurIPS Datasets and Benchmarks 2021. (arXiv)

[4] **Wang, X.,** et al. *Self‑Consistency Improves Chain of Thought Reasoning in Language Models.* arXiv:2203.11171, 2022. (arXiv)

[5] **Alain, G., Bengio, Y.** *Understanding Intermediate Layers Using Linear Classifier Probes.* arXiv:1610.01644, 2016. (arXiv)

[6] **Shao, Z.,** et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning for Open‑Source Models.* arXiv:2402.03300, 2024. (arXiv)

[7] **Mistral AI.** *MathΣtral (Mathstral‑7B) announcement and model overview*, 2024. (Mistral AI)

[8] **Nanda, N., et al.** *TransformerLens: A Library for Mechanistic Interpretability of Generative LMs* (GitHub & Docs), 2023–2025. (GitHub)

[9] **Lewkowycz, A.,** et al. *Solving Quantitative Reasoning Problems with Language Models (Minerva).* NeurIPS 2022; arXiv:2206.14858. (arXiv)