Probing the Limits of Mathematical World Models in LLMs

Henry Kvinge [1 2 *] **Elizabeth Coda** [1 3 *] **Eric Yeats** [1 *] **Davis Brown** [1 4] **John Buckheit** [1] **Sarah Scullen** [1] **Brendan Kennedy** [1] **Loc Truong** [1] **Bill Kay** [1] **Cliff Joslyn** [1] **Tegan Emerson** [1 5 6] **Michael Henry** [1] **John Emanuello** [7]

Abstract

There are now many studies supporting the idea that even when they are trained on a broad corpus of textual data scraped from the internet, large language models (LLMs) are (sporadically) capable of non-trivial mathematical tasks. This observation and a collection of studies from the interpretability community together suggest that LLMs extract surprisingly rich internal representations of mathematical objects. In this paper we ask the extent to which LLMs contain mathematical 'world models' that align with the way that mathematicians understand and think about mathematics. We focus on simple binary operations $\star : X \times X \to X$ like addition and multiplication which take two inputs a and b from a space X and produce a third element $a \star b = c$. Instead of assessing the correctness of the LLM response, we explore the extent to which the model captures the geometric structure of X, simple number-theoretic properties of a and b, and the algebraic properties of $\star$. We report mixed results. While the LLMs we tested tended to store substantial amounts of information (such as the divisibility properties of integers a and b in the expression $a \times b$) and sometimes extracted representations that aligned with existing mathematical structures (reconstructing a patch of $\mathbb{R}^2$ for example), these representations tended to be local in nature and lack robustness.

1. Introduction

Though substantial progress has been made over the course of the last 5 years, the problem of understanding the internal mechanisms that underlie large language model's (LLM's) remarkable ability to produce fluent textual output remains a major area of research. Though the sheer scale of modern networks may be the most obvious challenge in this project, the complexity and ambiguity of tasks that we expect LLMs to assist with is also a major barrier to developing a comprehensive understanding of these models. For example, since it would likely be challenging for a human to describe the granular steps they take when writing a poem, we can expect it to be even more challenging to analyze this process within the alien intelligence of an LLM. Given this, mathematical tasks where multiple rigorously defined algorithms are known have proven to be a valuable testbed for the interpretability community.

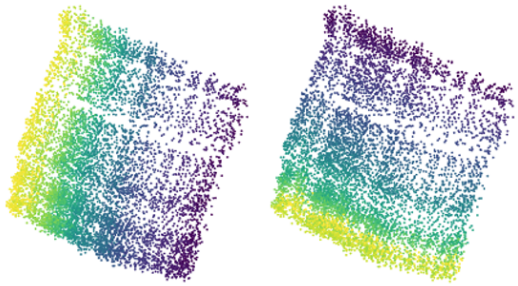

Figure 1. PCA projections of the representation of the '=' token in the prompt '`a * b =`' following the third block of Llama3-8B. Here `a` and `b` take all integer values between 0 to 100. **(Left)** Points colored by the magnitude of `a`. **(Right)** Points colored by the magnitude of `b`.

The majority of work in this line of research focuses on the actual computations involved in mathematical tasks (Nanda et al., 2023; Quirke & Barez, 2023; Nikankin et al., 2024; Zhou et al., 2024; Stolfo et al., 2023). How does a model take 42 and 36 and then produce 42×36? However, mathematical operations also come with a rich and rigorously defined supporting framework that is essential to mathematician's intuition and understanding. For instance, when a mathematician specifies a function f, they specify its do-

[*]Equal contribution [1]Pacific Northwest National Laboratory [2]University of Washington [3]University of California, San Diego [4]University of Pennsylvania [5]Colorado State University [6]University of Texas, El Paso [7]Laboratory for Advanced Cybersecurity Research, National Security Agency. Correspondence to: Henry Kvinge <henry.kvinge@pnnl.gov>.

ICML 2025 Workshop on Assessing World Models.

main X and range Y, $f : X \to Y$. Each of these come with their own geometry and topology which may be known to the mathematician and can constrain the behavior of f. A mathematician would also utilize a rich collection of analytic concepts to understand f at a high-level. For instance, notions of continuity and smoothness. We interpret this entire package as constituting what we call a *mathematical world model*. These have the value of being crisp and rigorous unlike the world models associated with fuzzier, real-world tasks. This makes them ideal for developing tools to assess world models more broadly.

In this work, we focus on world models associated with familiar binary operations $\star : X \times X \to X$ like addition and multiplication which even small LLMs can perform with some success. We analyze three aspects of these tasks (i) the geometry of X, (ii) properties of inputs a and b, and (iii) the algebraic properties of $\star$ (e.g., commutativity). Highlights include the observation that LLMs contain internal models that capture the linear structure of $\mathbb{R}^2$ that manifest when the model is prompted to solve a binary operation. We also find that when $X = \mathbb{Z}$, an LLM's representation of $a \times b$ contains extensive information about the divisibility of a and b. Finally, we describe inconclusive results relating to whether LLMs 'understand' if an operation is commutative or not (e.g., $a \star b = b \star a$).

Overall, our experiments suggest that LLM internal world models for basic binary operations share some features with frameworks used by human mathematicians. However, agreeing with other works in the field such as (Nikankin et al., 2024), we find that LLM representations tend to be more local in nature, less organized, and lack the robustness of the representations designed by human mathematicians.

2. Binary Operations as Small World Models

Binary operations are a fundamental construction across mathematics. They can be formalized as functions that take in a pair of elements from some space X and return a third element from X. For example, multiplication of integers takes two integers $a, b \in \mathbb{Z}$ and produces a third $a \times b = c \in \mathbb{Z}$. Other basic examples of binary operations include addition, subtraction, and division (when we exclude 0) of $\mathbb{Z}$ or $\mathbb{R}$, addition or multiplication of square matrices, or the union or intersection of sets. In this work we use the symbol $\star$ to denote an unspecified binary operation and a and b for arbitrary elements of X. We also let $[a, b]$ denote the closed interval from a to b (whether this includes only integers or all real numbers will be made clear from the context).

We view a fixed binary operation and its supporting specifications as defining a small world model. Aspects of this world model that we focus on include: the geometry of the input space $X \times X$, properties of the arguments a and b, and the algebraic properties of the operation $\star$. We do not analyze the actual process by which the model generates an answer as significant research effort continues to be aimed at understanding this (Kantamneni & Tegmark, 2025; Nikankin et al., 2024; Zhou et al., 2024), but note that in certain cases, one may be able to derive aspects of the world model from mechanisms by which a model performs computation.

2.1. Capturing the Structure of Input Space

The geometry of X (especially when $\star$ satisfies conditions such as continuity or smoothness) often provides useful heuristics that can help with computation of a binary operation. As a simple example, when $X = \mathbb{R}$, the geometric notion of magnitude can help us predict $a \times b$. If a and b both have large magnitude so will $a \times b$. Given this, it is not unreasonable to ask about the extent to which LLMs incorporate aspects of the geometry of $X \times X$ in their representation of an expression like $a \star b$.

Naive visualization of the representations of '=' in '`a ⋆ b =`' across many values of a and b suggests that several common LLMs contain an internal model of subsets of $X \times X$. In Figure 1 for instance, we show two PCA visualizations of representations of the '=' token drawn from the residual stream after the 3rd block of Llama3-8B (Grattafiori et al., 2024) where `a` and `b` take values between 0 and 100. In the first visualization we color points by the magnitude of the integer `a` and in the second we color them by the magnitude of integer `b`. As can be seen by the colors, the first two principal components approximately capture the linear structure of the coordinate plane as defined by (a, b).

To measure this more quantitatively, we collect $10,000$ hidden activations for '=' in '`a ⋆ b =`' following each transformer block in an LLM, where `a` and `b` are integers between 0 and 100. We split the data recorded from each layer into train and test sets, and use the training set to train a linear regression model $M : \mathbb{R}^n \to \mathbb{R}^2$ to map the representations of the token '=' to the coordinates (a, b) in $\mathbb{R}^2$. The idea is that if the model actually organizes representations of '=' according to the linear structure of $\mathbb{R}^2$ (as suggested by Figure 1) then we will be able to effectively learn the map M. If the model organized each '=' nonlinearly or in a linear manner that did not respect the parametrization by (a, b), M would not be performant (as happens if one shuffles the (a, b) labels on each '=' representation). Indeed, the existence of a performant M at a point in the residual stream $\mathbb{R}^n$ implies that this space decomposes as $\mathbb{R}^n \cong \ker(M) \oplus \text{im}(M)$, where $\ker(M)$ is the kernel of M and $\text{im}(M)$ is the image of M. The latter is a subspace of $\mathbb{R}^n$ where each representation of '=' is parametrized by coordinates (a, b) (at least for values of a and b represented

in the experiment). We measure fit of M by the coefficient of determination, R^2.

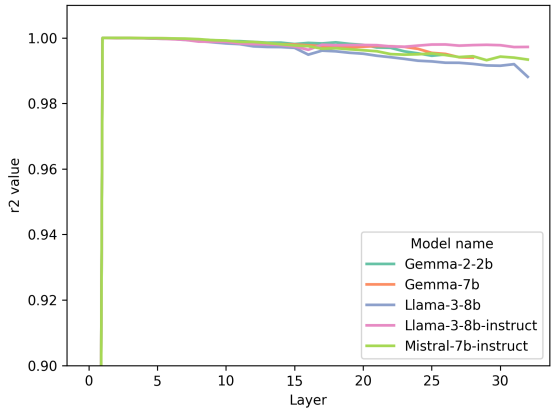

Figure 2. R^2 scores for a linear regression model trained to predict coordinates (a, b) from the token '=' in '`a * b =`' across the layers of various LLMs.

The results of this experiment are shown, layer by layer, for several models in Figure 2. As can be seen, a linear map can very effectively predict `a` and `b` from the representation of '=', suggesting that prompting with '`a * b =`' causes these LLMs to carry a copy of the input space ($[0, 100] \times [0, 100]$) throughout the residual stream. For a given subset $U \in \mathbb{R}^2$, we call this a model's *internal copy of* U and denote the corresponding subspace in the residual stream by $\mathcal{I}(U, f, i, P)$, where f denotes the model, i denotes the block after which the representations were extracted, and P denotes the input prompt set (which is assumed to terminate with '=').

In the remainder of this section we pose a series of questions and answers to better understand $\mathcal{I}(U, f, i, P)$ with a focus on the dependence on P, the universality of $\mathcal{I}(U, f, i, P)$ across scales of $\mathbb{R}^2$, and its robustness. All the results shown are for Gemma-7b (Team et al., 2024) though we found similar results for other models in this size range.

Is an internal representation of U only extracted when computing binary operations, or is this a more general phenomenon? Is the copy of $\mathcal{I}(U, f, i, P)$ the same between different binary operations? We re-ran our experiments after replacing '$*$' with: '+', '−', '%', ',', ' ', 'mod', '`is a number, another number is`', and '`(&*&$\hat{\&}$*&`' (Figure 6). We were able to consistently detect $\mathcal{I}(U, f, i, P)$ for each of these variants of the original prompt set, even when the input does not relate to a binary operation or mathematics at all. This suggests that $\mathcal{I}(U, f, i, P)$ is extracted in a wide range of circumstances, contingent only on the presence of two numbers in the prompt. Interestingly, while the representation remains robust across input variations, the instances where it decays most are those where the model effectively uses information about the two numbers for a mathematical operation (e.g., addition, multiplication, and subtraction).

To understand whether $\mathcal{I}(U, f, i, P_1)$ and $\mathcal{I}(U, f, i, P_2)$ are the same between prompts sets P_1 and P_2 with different binary operations, we measured the R^2 score of a linear classifier M trained to predict (a, b) from the representation of '=' in the prompt '`a * b =`' when it was applied to '=' from the prompt '`a operation b =`' where '`op`' is either replaced by text unrelated to multiplication or is replaced by a symbol or word that denotes multiplication but is not '$*$' (e.g., '`times`'). As shown in Figure 12, the performance of M degrades across all instances but the degradation is generally greater when the operation (or non-operation) is not multiplication, suggesting that as a subspace, $\mathcal{I}(U, f, i, P)$ carries some semantic information about P.

Does the linear connection M between U and $\mathcal{I}(U, f, i, P)$ extend beyond the values of '`a`' and '`b`' used to train it? By necessity, '`a`' and '`b`' are always sampled from some finite range U. While high accuracy on test examples from U shows that $\mathcal{I}(U, f, i, P)$ effectively captures U, we have provided no evidence that the connection captures a larger region of $\mathbb{R}^2$. That is, can M accurately predict (a, b) from '=' when '`a`' and '`b`' are not in U? This is important as a measure of the robustness and universality of $\mathcal{I}(U, f, i, P)$. It also represents an important value in mathematics, where structures should be made as general as possible.

To better understand this, we evaluate M on a different set V than it was trained for. We find that across layers and for many different values of U and V, M consistently fails on even moderately out-of-distribution data. This even holds when elements of V are bounded by elements of U so that learned $\mathcal{I}(U, f, i, P)$ fails mild interpolation. As an example, in Figure 3 we show predictions of a from (a, b) for M trained on pairs of integers a, b drawn from $[0, 19] \cup [30, 49]$ and then evaluated on pairs of integers drawn from the interval $[20, 29]$. We conjecture that $\mathcal{I}(U, f, i, P)$ is closely related to the base 10 representation of numbers in U. So if M hasn't seen a particular digit appearing at a particular place (e.g., a 2 in the tens place), it will fail. This would align with results from (Levy & Geva, 2024))

In Appendix A.1 we describe additional experiments that include the difference between $\mathcal{I}(U, f, i, P)$ from block to block in a single model, $\mathcal{I}(U, f, i, P)$ for different values of U, and high-dimensional analogues of $\mathcal{I}(U, f, i, P)$ when more than two arguments are present in an expression.

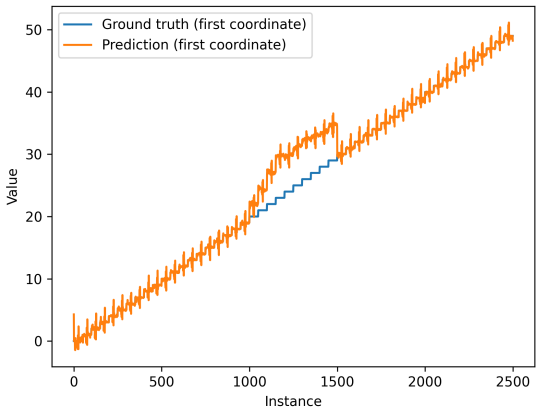

Figure 3. The first coordinate predicted by M (orange) vs. the true first coordinate (blue) for the task of predicting (a, b) from the token '=' in the prompt '`a * b =`. The model was trained on a, b from $[0, 19] \cup [30, 49]$ and then evaluated on a, b from the whole interval $[0, 49]$. The area where blue and orange are different corresponds to data from $[20, 29]$ where there was no training data.

2.2. Do LLMs Extract Divisibility Properties from Arguments in a Binary Operation?

Despite their elementary nature, we can ascribe a rich set of properties to a given integer[1]. We view these properties as part of the broader mathematical world. We here explore the extent to which LLMs contain information about the most familiar of these: divisibility. Similar to our experiments above we explore the representations of '=' in the prompt '`a * b =` ' at different layers of the model. We use linear probing to look for evidence that an LLM 'knows' about divisibility properties of `a` or `b`.

Probing results on Gemma-7b representations for divisibility of `a` drawn from $0 \leq a, b \leq 100$ are shown Figure 4. Here the x-axis corresponds to the layer at which the probing is performed. The y-axis corresponds to the divisor q, the integer we are probing for divisibility with respect to (so the first row corresponds to predicting whether `a` is even or odd). The color of the cell is the F1-score (we use F1-score because for large q, most a will not be divisible by q leading to imbalance training and test sets). We see that a logistic regression classifier is highly performant, especially following block 21. We found similar results for Llama3-8B.

[1]Recall Ramanujan's reply to Hardy's claim that the number on a taxi (1729) was "dull": ' it is a very interesting number; it is the smallest number expressible as the sum of two cubes in two different ways." (Hardy, 1999).

Prior work has suggested that LLMs represent numbers via Fourier features (Zhou et al., 2024) or generalized helices (Kantamneni & Tegmark, 2025). Both of these implicitly encode divisibility information so in that sense these results are not surprising. On the other hand, these works focus on frequencies related to $q = 2, 5, 10$, etc. Our results show that LLMs capture a far broader set of divisibility properties than just this. Furthermore, in Appendix B we show that these feature appear to be important in multiplication. Perturbing the representation of '=' toward a feature '`a` is divisible by 5' leads the model to predict that $a \times b$ is a multiple of 5, even when it is not. Interestingly, we find that these divisibility properties are somewhat weaker when the task is addition, providing evidence that their emergence is also somewhat responsive to their usefulness in the task (Figure 7).

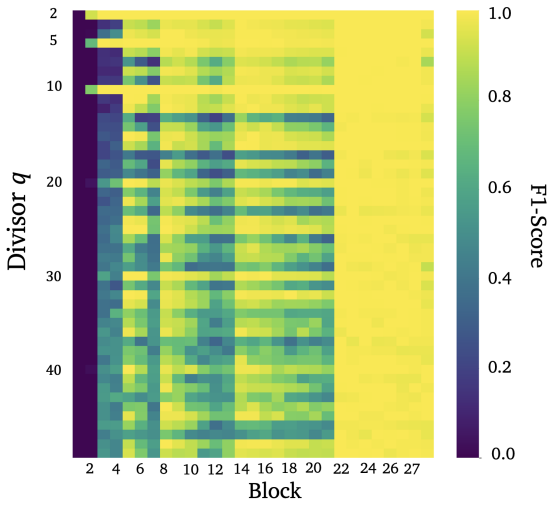

Figure 4. F1-scores for probes trained to classify whether `a` is divisible by a value q using the representation of the token '=' in the prompt '`a * b =`' at different layers in Gemma-7b.

2.3. The Algebraic Properties of the Binary Operation

At a high-level, mathematicians group binary operations based on their algebraic properties. The most familiar of these is commutativity which says that $a \star b = b \star a$. Some operations like addition and multiplication of real numbers satisfy commutativity, while others like division, subtraction, or the multiplication of square matrices do not.

Do LLMs understand commutativity? To investigate this, we provided the model with pairs of prompts: '`a operation b =`' and '`b operation a =`', where in this case `a` and `b` were random lowercase Latin characters and `operation` was one of the operations $\star$, `%`, `+`, `-`. We then measured the cosine similarity between representations of the tokens '=' for each pair throughout layers of Gemma-7b. The idea is that if the LLM understands commutativity,

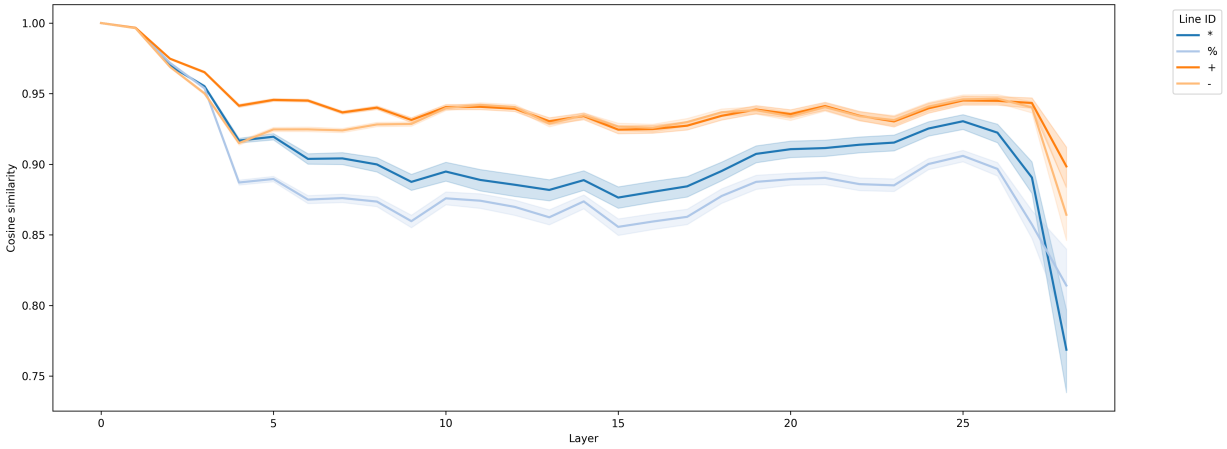

Figure 5. The cosine similarity between representations of the token '=' in the prompts '`a operation b =`' and '`b operation a=`' where $0 \leq a, b \leq 50$. The different operations are specified in the legend. Shaded regions indicate 95% confidence intervals over all cosine similarity values calculated at that layer for the given operation.

then representations of the two prompts '`a operation b =`' and '`b operation a =`' should show grater similarity when the operation is commutative as opposed to when it is not. The reason we did not provide numbers in the prompts is that we wanted to understand if the model has internalized commutativity at an abstract level. Otherwise, representations might appear similar because in each case the solution to both prompts is the same. That is, do '`12 * 56 =`' and '`56 * 12 =`' have similar representations because the model has an abstract sense of commutativity or because the solution to both problems is 672?

Our results, shown in Figure 5, are somewhat ambiguous. On the one hand, if one considers $(+,-)$ and $(\times,\%)$ to be pairs, then the commutative operations ($+$ and $\times$) tend to show marginally greater similarity between '=' in the prompts '`a operation b`' and '`b operation a`'. On the other hand, this difference seems to be fairly marginal and in the case of $(+,-)$, vanishes a third of the way through the model.

3. Conclusion and Limitations

In this work we probe the mathematical world models of some small LLMs to understand the extent to which their internal representations align with formal mathematical representations. We focus on the case of binary operations on integers and real numbers. Our results suggest that these models often contain rich and interesting mathematics encoded in their representations, but we do not find evidence that these representations are robust, general, and global in the same ways that the constructions developed by human mathematicians are.

We end by noting a number of limitations in our study. The first is that we do not investigate larger LLMs which may have internal representations that align more with the mathematical frameworks developed by humans. We also note that while the methods that we used can often detect structures, they cannot prove that something does not exist. For example, the fact that we did not find $\mathcal{I}(U, f, i, P)$ that generalize well beyond their training set does not imply that this structure does not exist in the model.

References

Chau, H., Jenne, H., Brown, D., He, J., Raugas, M., Billey, S., and Kvinge, H. Machine learning meets algebraic combinatorics: A suite of datasets capturing research-level conjecturing ability in pure mathematics. *arXiv preprint arXiv:2503.06366*, 2025.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Engels, J., Liao, I., Michaud, E. J., Gurnee, W., and Tegmark, M. Not all language model features are linear. *arXiv e-prints*, pp. arXiv–2405, 2024.

Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C. F., Denain, J.-S., Ho, A., Santos, E. d. O., et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hardy, G. H. *Ramanujan: Twelve lectures on subjects suggested by his life and work*, volume 136. American Mathematical Soc., 1999.

Kantamneni, S. and Tegmark, M. Language models use trigonometry to do addition. *arXiv preprint arXiv:2502.00873*, 2025.

Levy, A. A. and Geva, M. Language models encode numbers using digit representations in base 10. *arXiv preprint arXiv:2410.11781*, 2024.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Nikankin, Y., Reusch, A., Mueller, A., and Belinkov, Y. Arithmetic without algorithms: Language models solve math with a bag of heuristics. *arXiv preprint arXiv:2410.21272*, 2024.

of America (MAA), M. A. American invitational mathematics examination. URL https://www.maa.org/math-competitions/aime. Accessed May 7, 2025.

Quirke, P. and Barez, F. Understanding addition in transformers. *arXiv preprint arXiv:2310.13121*, 2023.

Robinson, M., Dey, S., and Sweet, S. The structure of the token space for large language models. *arXiv preprint arXiv:2410.08993*, 2024.

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J., Ellenberg, J. S., Wang, P., Fawzi, O., et al. Mathematical discoveries from program search with large language models. *Nature*, 625 (7995):468–475, 2024.

Serre, J.-P. et al. *Linear representations of finite groups*, volume 42. Springer, 1977.

Skean, O., Arefin, M. R., Zhao, D., Patel, N., Naghiyev, J., LeCun, Y., and Shwartz-Ziv, R. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Stolfo, A., Belinkov, Y., and Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*, 2023.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Valeriani, L., Doimo, D., Cuturello, F., Laio, A., Ansuini, A., and Cazzaniga, A. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36:51234–51252, 2023.

Yin, F., Srinivasa, J., and Chang, K.-W. Characterizing truthfulness in large language model generations with local intrinsic dimension. *arXiv preprint arXiv:2402.18048*, 2024.

Zhou, T., Fu, D., Sharan, V., and Jia, R. Pre-trained large language models use fourier features to compute addition. *arXiv preprint arXiv:2406.03445*, 2024.

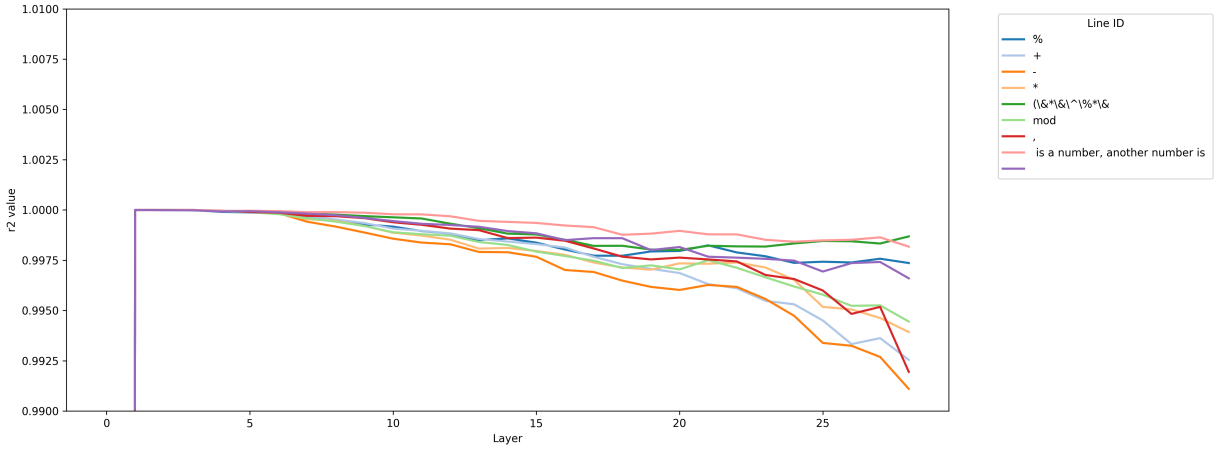

Figure 6. R^2 scores on held-out test data for linear maps trained to take the representation of the '=' token in the prompt '`a operation b =`' following the third block of Gemma-7b to the point $(a, b) \in \mathbb{R}^2$, where `operation` takes values '+', '−', '%', '`,`', ' ', '`mod`', '` is a number, another number is `', and '`(&*&$\hat{\%}$*&`'. The model was trained on 80% of integers pairs (a, b) where $0 \leq a, b \leq 100$. As can be seen from the plot, residual stream has a subspace where the representation of the '=' token stores a copy of $\mathbb{R}^2$.

A. Related Work

Mathematics has become an important tool for better understanding LLMs. Mathematics based benchmarks (of America , MAA; Cobbe et al., 2021) provide an opportunity to evaluate models on questions whose answer has very little ambiguity, with large LLMs increasingly able to solve hard graduate level questions (Glazer et al., 2024). There is also growing interest in using them to address open problems (Romera-Paredes et al., 2024; Chau et al., 2025).

The fact that we know low level algorithms for mathematics problems also makes mathematics a useful testbed for understanding how LLMs solve problems at a mechanistic level. For example, (Nanda et al., 2023) provided an in-depth analysis of modular addition in small 2-layer transformers, showing that these models use frequency-based arithmetic to solve the problem (or in algebraic terms, they work with the irreducible representations of the cyclic group (Serre et al., 1977)). The work uses this analysis to better understand the phenomenon of grokking in a toy setting. Multidigit arithmetic was explored in (Quirke & Barez, 2023), where experiments suggest that in some situations, small transformers trained for arithmetic tasks converge on the same algorithm across multiple training runs.

Research has not been restricted to small transformers. (Kantamneni & Tegmark, 2025; Zhou et al., 2024) provide evidence that LLMs also use frequency-based 'clock arithmetic' (similar to (Nanda et al., 2023)) to perform addition and other basic operations. On the other hand, there is substantial evidence that large models trained from varied textual sources do not use a single method to solve a given type of problem (as mathematicians would program a computer to do) but that they rather rely on a 'bag of heuristics' (Nikankin et al., 2024) that each address special, local cases. There has also been research into the ways that LLMs represent numbers (Levy & Geva, 2024). This has shown that LLMs utilize specific aspects of the base 10 representation and that model errors in arithmetic problems tend to reflect this.

Similar to the present work, there is a broad research thrust towards characterizing geometric structures and properties of LLM representations. This includes measuring geometric characteristics such as intrinsic dimension (Valeriani et al., 2023; Yin et al., 2024) or curvature (Robinson et al., 2024; Skean et al., 2025) as well as looking for specific structures. (Engels et al., 2024) found that LLMs sometimes represent cyclical phenomena (such as the months of the year) as circles in their internal representations.

A.1. Additional Results and Figures Related to Representations of Input Space

- **Does the same copy of $\mathcal{I}(U, f, i, P)$ persist throughout the residual stream of a model?** In the models that we investigated we find that it does not. This can be seen in the heatmaps Figure 8 and Figure 9 corresponding to Llama3-8B and Gemma-7B and integers a and b in $[0, 100] \times [0, 100]$. In these, an entry at position (i, j) is the

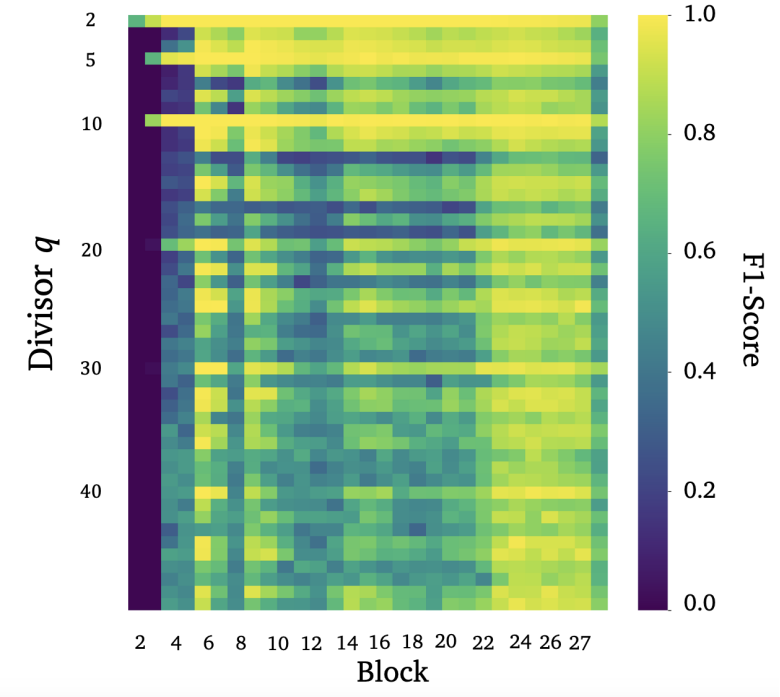

Figure 7. F1-scores for probes trained to classify whether `a` is divisible by a value q using the representation of the token '=' in the prompt '`a + b =`' at different layers in Gemma-7b.

R^2 score obtained by taking the linear map $M : \mathbb{R}^n \to \mathbb{R}^2$ trained on layer i and applying it to the corresponding activations from layer j. If $\mathcal{I}(U, f, i, \{\text{`a } \star \text{ b ='}\})$ was similar between layers we would expect that the linear map $M$ corresponding to layer $i$ would achieve reasonable accuracy on layer $j$. As can be seen from the figures, this does not happen. So, while an internal representation $\mathcal{I}(U, f, i, \{\text{`a } \star \text{ b ='}\})$ exists throughout the residual stream, it is not the same subspace. We conjecture that the model continues to extract distinct but isomorphic internal copies of $\mathbb{R}^2$ throughout transformer blocks in the model.

- **Can subspaces $\mathcal{I}(U, f, i, P)$ be found for subsets of $\mathbb{R}^2$ outside of $U = [0, 100] \times [0, 100]$?:** In Figure 10 we show R^2 scores for linear maps M trained on '`a`' and '`b`' from various U including $U = [-100, 0]$, $U = [0, 1]$ (using float representations of real numbers), $U = [0,\ 10,000]$, $U = [0,\ 100,000]$, $U = [0,\ 100,000,000]$, and $U = [0, 1 \times 10^{14}]$. In each case we sample $10,000$ pairs (a, b) uniformly from the interval. We find that in all the cases that we tested, a map M can be found which achieves high R^2 score. This is surprising since the majority of numbers in these latter U are far beyond the scale that our models are capable of reliably manipulating. Rather, based on this result we conjecture that $\mathcal{I}(U, f, i, P)$ may arise directly from the way that LLMs use the base 10 representation of numbers (Levy & Geva, 2024).

- **Do higher dimensional $\mathcal{I}(U, f, i, P)$ manifest for operations involving more than two arguments?** Another way to expand this study is to look at higher-dimensional analogues of $\mathcal{I}(U, f, i, P)$. If we apply a binary operation repeatedly we can include k arguments (e.g., '`42 * 36 * 72 =`' for $k = 3$). Can we find an internal representation of $\mathbb{R}^k$ in this case? To test this we run the same test as we did in the body of Section 2.1, but instead of using prompts `a * b =`, we use prompts with k arguments for $k = 2, 3, 4, 5$. In Figure 11 we see that adding any more than two arguments results in significant drops of performance of M at later layers of the network. Learning a map from $\mathbb{R}^n$ to $\mathbb{R}^k$ may require more data when k is larger which may account for some of this decline but it may also relate to LLM's inability to track and manipulate many numbers at once.

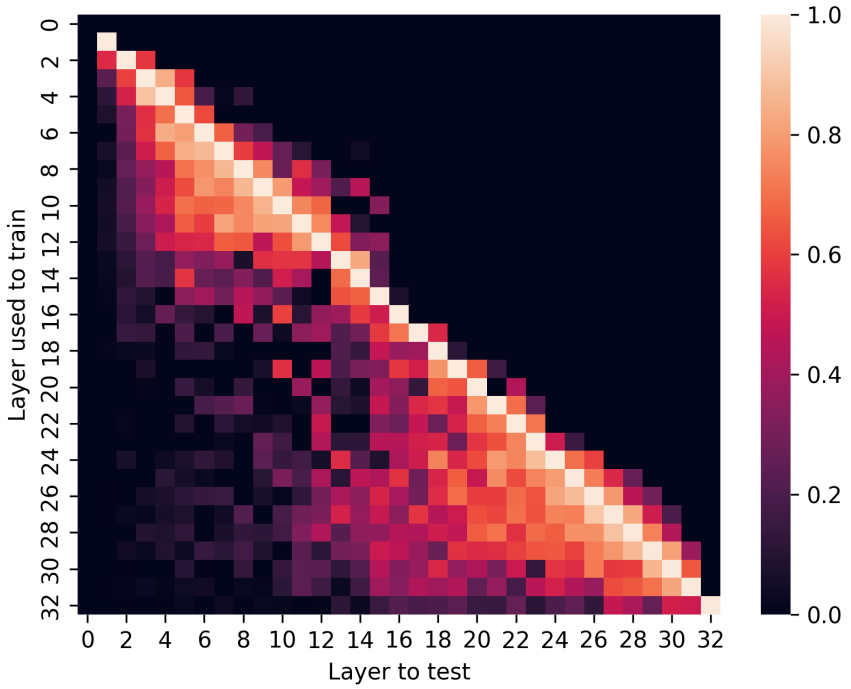

Figure 8. A plot capturing the extent to which subspaces $\mathcal{I}(U, f, i, P)$ transfer between layers in Llama-3-8B. Each cell in the heatmap is an R^2 score representing performance of a linear maps trained on representations following block i but evaluated on representations following block j. We scale cell colors to range from 0 to 1. As can be seen, while some transfer occurs, copies $\mathcal{I}(U, f, i, P)$ are at least somewhat specific to each layer.

B. Example: Perturbing Representations of Argument Divisibility to Changes Model Outputs

To understand whether or not the divisibility features that we detected with our logistic regression classifier in Section 2.2 have an effect on model computation (rather than just being incidental) we systematically perturbed representations of the token '=' toward an 'a is divisible by 5' direction. We found this direction by taking the centroid C of 100 representatives of the '=' token from prompts where a was a multiple of 5. We then perturbed the representation of the '=' token from an arbitrary expression '`a * b =`' by adding $10C$ to it.

The result of doing this to the prompt '`49 * b =`' across layers of Llama3-8B-instruct and $0 \leq b \leq 100$, is shown in Figure 14. The unperturbed version can be found in Figure 13. In both, the color of the cell in the grid shows whether the final output of the model was divisible by 5 (green), not divisible by 5 (red), or not an integer (blue). We used top-50 sampling which is why there is some variation in the column corresponding to $b = 85$ in Figure 13. As can be seen, perturbing toward 'a divisible by 5' (at least at layers 15 through 26), changes the model's answer to be a multiple of 5 suggesting that LLMs do use these divisibility features when solution multiplication tasks.

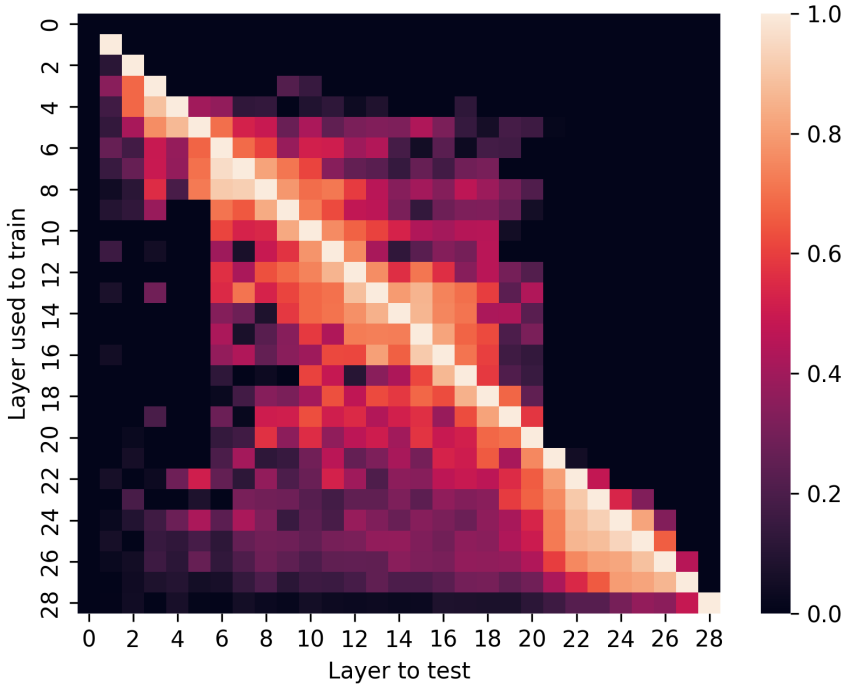

Figure 9. A plot capturing the extent to which subspaces $\mathcal{I}(U, f, i, P)$ transfer between layers in Gemma-7b. Each cell in the heatmap is an R^2 score representing performance of a linear maps trained on representations following block i but evaluated on representations following block j. We scale cell colors to range from 0 to 1. As can be seen, while some transfer occurs, copies $\mathcal{I}(U, f, i, P)$ are at least somewhat specific to each layer.

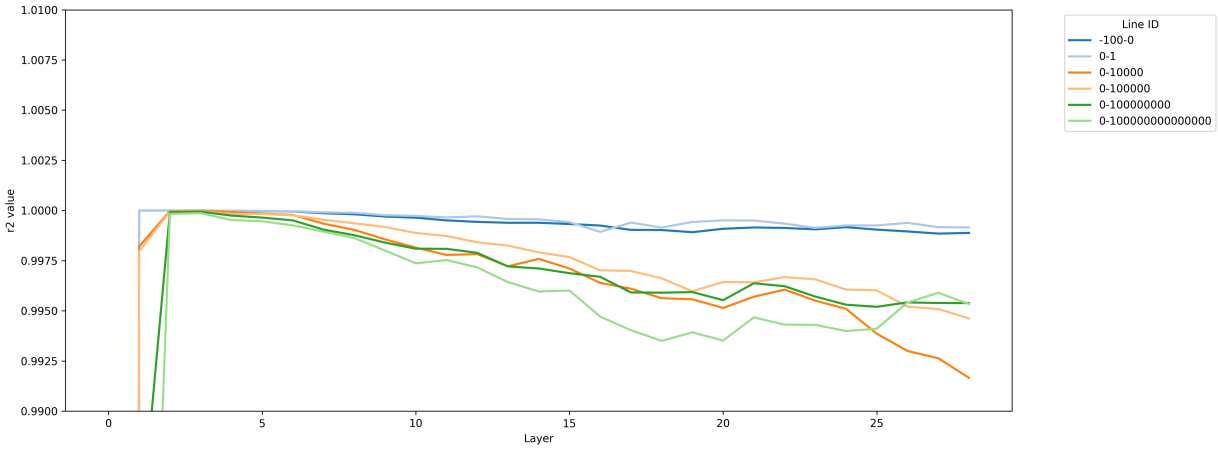

Figure 10. R^2 scores for a linear regression model trained to predict coordinates (a, b) from the token '=' in '`a * b =`' where a and b are taken from different ranges (identified in the legend). For instance, $0 - 10,000$ indicates that a and b are sampled uniformly from the closed interval $[0, 10,000]$.

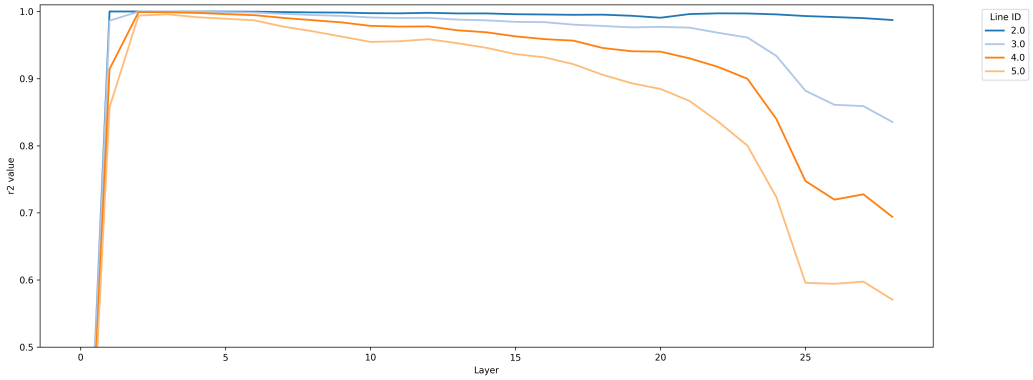

Figure 11. R^2 scores on held-out test data for linear maps trained to take the representation of the '=' token in the prompt '`a_1 operation a_2 ... a_k-1 operation a_k =`' to the point $(a_1, \ldots, a_k) \in \mathbb{R}^k$. Different lines correspond to different values of k.

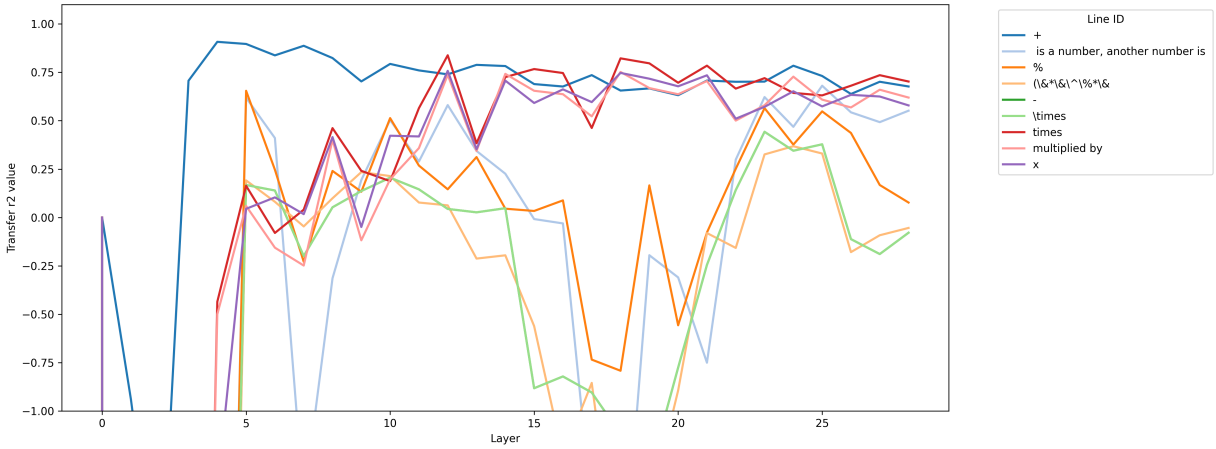

Figure 12. R^2 scores for a linear regression model trained to predict coordinates (a, b) from the token '=' in '`a * b =`' and then evaluated on '`a operation b =`' where '`operation`' varies. Note that some `operation` are synonymous with multiplication and some are not.

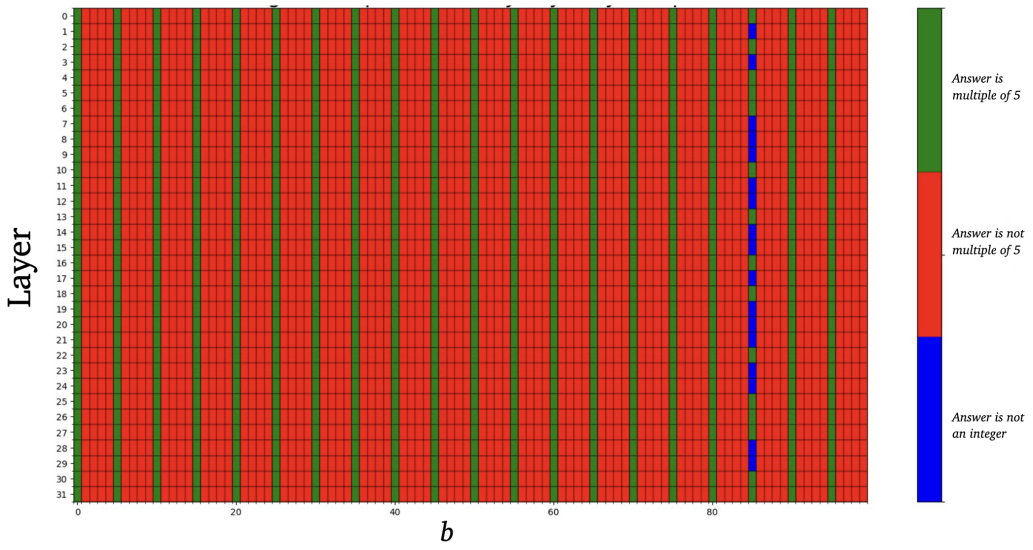

Figure 13. A figure showing whether a Llama3-8B-instruct prediction of '`49 * b =`' is a multiple of 5 (green), not a multiple of 5 (red), or not a number (blue) for $0 \leq b \leq 50$. Non-greedy sampling accounts for the variation in the column $b = 85$.

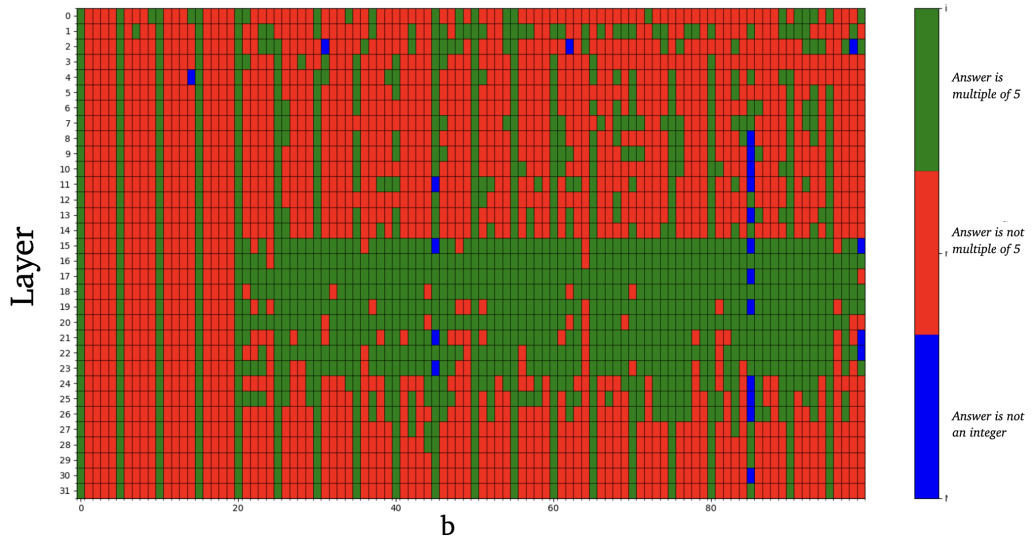

Figure 14. A figure showing whether a Llama3-8B-instruct prediction of '`49 * b =`' is a multiple of 5 (green), not a multiple of 5 (red), or not a number (blue) for $0 \leq b \leq 50$. In this case the representation of '=' has been perturbed toward the direction 'a is divisible by 5' at a given layer specified by the y-axis.