How Do Transformers "Do" Math? A Study of Meaningful Intermediates for Linear Regression

Subhash Kantamneni
Massachusetts Institute of Technology
subhashk@mit.edu

Daniel Kim
Massachusetts Institute of Technology
dyk0518@mit.edu

Matthew McManus
Massachusetts Institute of Technology
mattmcm@mit.edu

Alex Quach
Massachusetts Institute of Technology
aquach@mit.edu

Abstract

We investigate if transformers perform mathematical computations by representing meaningful intermediate values. Focusing on linear regression, we probe models for encodings of the slope. Our experiments show that larger models robustly encode this intermediate in usable forms. We find a strong correlation between intermediate encoding and task learning. Using reverse probing and representational interventions, we provide causal evidence of these representations in computations. Our work advances the interpretability of transformer models and promotes more transparent machine learning systems.

1 Introduction

Transformer language models have demonstrated remarkable performance across a wide range of tasks, from natural language understanding to mathematical reasoning. However, the underlying mechanisms by which transformers operate remain largely opaque, limiting our ability to interpret their behavior and build trust in their outputs. In particular, there is a lack of clarity around how transformers perform apparently sophisticated computations, such as solving mathematical equations.

In this work, we aim to shed light on the inner workings of transformers by investigating the role of *intermediates*—internal quantities computed by the model that are instrumental to the final output but not directly observable. Formally, we hypothesize that if a method g is being used by a transformer for calculation, its associated intermediate I is encoded in the transformer. We iterate and improve on this hypothesis in the setting of linear regression to provide a framework on intermediates useful to other researchers.

We structure our work by asking fundamental questions about intermediates, namely

- **What** is an intermediate?
- **How** can intermediates be encoded and how can we robustly probe for them?
- **When**, or under what circumstances, are intermediates encoded?

All of these questions develop an understanding of intermediates that builds up to the **Key Question: How can we use intermediates to demonstrate that a transformer is using a method in its computations?** By systematically addressing these questions, we develop a framework for identifying and interpreting the intermediates in transformer computations. We show that larger models are more likely to represent intermediates in human-interpretable forms, and that a model's ability to "learn" a task is closely tied to its encoding of the relevant intermediates. Crucially, we provide causal evidence that models are using encoded intermediates in their computations by demonstrating that interventions on latent representations can predictably modify model outputs.

With an answer to this general question, we can use our understanding of intermediates in the linear regression case to eventually better understand how transformers model more complex phenomena.

2 Related Work

2.1 Transformers

Transformer models which use attention mechanisms (Vaswani et al., 2017) have been shown to achieve State-of-the-Art performance in a variety of fields. By assigning different weights to different parts of the input, the attention mechanism allows the model to weigh the relevance of each piece of information, thereby enhancing its ability to capture intricate patterns and dependencies within the data. Notably, recent breakthroughs in large language models (LLMs), exemplified by GPT-3 (Brown et al., 2020) rely on transformer architectures. Beyond their conventional application in nat-

ural language processing (NLP), transformers have demonstrated efficacy across diverse technical domains, including physics and mathematics. For instance, transformer models have exhibited success in particle physics simulations (Pastor-Serrano and Perkó, 2022). This widespread success prompts our study's inquiry: "How do transformers acquire and encode complex technical concepts?"

2.2 Mechanistic Interpretability

Mechanistic interpretability (MI) is a burgeoning field that aims to understand "how" models do various tasks. For example, (Nanda et al., 2023) found that transformers implement modular addition using the discrete Fourier transformer by reverse engineering the weights of the model. But models can often be more complex than initially meets the eye - (Zhong et al., 2023) found that (Nanda et al., 2023)'s description of the modular addition task was incomplete, and the model employs a "Pizza" algorithm in addition to the original "Clock" algorithm. We take a more coarse-grained approach to investigate how transformers model linear regression, but remain cautious in drawing conclusions from incomplete evidence.

2.3 Interpretability Probes

(Alain and Bengio, 2018) introduce linear probes as a means to understanding the dynamics of intermediate layers, by predicting the target labels from these layers. If the predictor is accurate, then one can argue that the representation captures the corresponding concept. We use linear probes to argue that hidden state representations capture versions of the slope term in linear regression. While (Laina et al., 2022) introduces reverse linear probes to map label vectors to representation vectors to show the semanticity of data representations. In our work, we use this technique in our simple linear regression problem to quantify the proportion of the embedding representation explained by the various polynomial forms of the slope term.

2.4 Linear Regression

Other works have demonstrated the merits of using linear regression as a task problem. (Garg et al., 2023) show the in-context ability of transformers to in-context learn linear functions. While (Akyürek et al., 2023) use linear regressions as a problem for showing transformer-based in-context learners implement SGD implicitly. And (Vacareanu et al., 2024) demonstrate how pre-trained large language models can do linear and non-linear regression. We use linear regression as a problem to probe and analyze intermediates.

3 Experimental Setup

In this section, we introduce different possible ways the transformer models can encode the intermediates. Then, we show the setup of our study that helps us investigate which form of intermediate encoding is being used in the examined transformer models.

3.1 What is an intermediate?

We define an intermediate as a quantity that a transformer uses to complete a computation but is not directly inputted/outputted to/by the transformer. More formally, if the input to the transformer is X and its output Y, we can model the transformer's computation as $Y = g(X, I)$, where g is the method used and I is the intermediate of that method. For example, if we believe the transformer is computing the linear regression task using $Y = wX$, then $g(X, I) = g(X, w) = wX$.

3.2 Probing for intermediates

We want to understand what form of the intermediate, $f(I)$, is encoded in the network's hidden states, HSs. For example, while it may be obvious to humans to compute $y = wx$, perhaps transformers prefer $\exp(\log(w) + \log(x))$ or $\sqrt{w^2x^2}$. We want to develop a robust probing methodology that captures these diverse possibilities. We identify three ways an intermediate I can be represented: linearly, nonlinearly, and not at all.

Linearly encoded We say I is linearly encoded if there is a linear network that takes $I = Linear(HS)$ for a particular hidden state in the network. We determine the strength of a linear encoding by evaluating how much of the variance in I can be determined by the HS, e.g. the R^2 of the probe.

Nonlinearly encoded To probe for an arbitrary $f(I)$, we define a novel **Taylor probe**, which finds coefficients a_i such that $f(I) = a_1 I + a_2 I^2 + ... + a_n I^n$, and $f(I) = Linear(HS)$. To actually implement this probing style, we use Canonical Correlation Analysis probes (cca, 2007), which given some multivariate data X and Y, find directions within X and Y that are maximally correlated. Here, $X = [I, I^2, I^3, ..., I^n]$, and $Y = HS$. If I is of bounded magnitude and n is sufficiently

large, we are able to probe the transformer for any function $f(I)$. We evaluate the strength of this nonlinear encoding using R^2.

Not encoded If I fails to be linearly or nonlinearly encoded, we say that it is not encoded within the network. Note that this does not imply that the information required to represent I is not at all stored in the network, just that $f(I)$ is not a linear function of the model's hidden states at a single depth position (e.g. it's possible $f(I) = MLP(HS)$). But since transformers are a sequence of linear functions interspersed with nonlinearities (Elhage et al., 2022), it's unlikely that the transformer is using $f(I)$ in its calculations if it's not a linear feature of hidden states, making this outcome less interesting to us.

3.3 Training

Setup In our linear regression setup, we generate X and w between $[-0.75, 0.75]$, where X has size $(5000, 65)$ and w with size $(5000,)$. We generate Y using $Y = wX$, and train a small transformer with $L, H = 1, 16$ to predict y_{n+1} given $[x_1, y_1, ..., x_n, y_n, x_{n+1}]$. We want to apply our probing techniques to better understand what types of models generate intermediates. Under the described setting of linear regression, we train transformers of size $L = [1, 2, 3, 4, 5]$ and $H = [2, 4, 8, 16, 32]$. We train for 20,000 epochs using Adam (Kingma and Ba, 2017) and with a mean squared error (MSE) loss.

4 Results

4.1 Larger models have stronger encodings of intermediates

We find that smaller models often don't have w encoded, while larger models encode w linearly, as evidenced by Fig. 1. We formalize this further by defining $\max(\bar{R^2})$ as the maximum average R^2 value across context lengths, identifying the depth at which an intermediate representation is optimally encoded. In Fig. 2, we observe a clear phase transition in encoding across model size and also find that $\max(\bar{R^2})$ does not significantly improve if we extend the degree of the Taylor probes to $n > 2$. Thus, in the case of linear regression, we find that models represent w linearly, quadratically, or not at all.

It's somewhat intuitive that larger models have w directly encoded - they have the capacity to be inefficient with their storage of information, compared to smaller networks that likely have to be more efficient with compression and storage of information. However, we do see that some small models achieve an encoding of w, like the quadratic encoding for the $L = 2$, $H = 8$ model in Fig. 2. We attribute this to the "lottery ticket hypothesis" - larger models have more "lottery tickets" in their increased capacity to find a "winning" representation of w (Frankle and Carbin, 2018). Interestingly, the intuitive understanding that larger models have w better encoded leads us to the counterintuitive conclusion that larger models are actually *more* interpretable for our purposes.

4.2 Encoding quality is tied to model performance

We find that the improvements in model prediction as a function of context length, often deemed in-context learning, are correlated to improvements in the encoding of w. In Fig. 3, we see that for better-performing models, the trajectories of the encoding of w and model MSE are increasingly similar. If our models were indeed using the identified representations of w in their computations, this would make sense - our models need to calculate $y_{n+1} = wx_{n+1}$, but x_{n+1} is always known with perfect precision because of skip connections in the model. Thus, in theory, all the model needs to do to calculate y is understood w well, which explains the findings of Fig. 3.

4.3 Transformers causally use intermediates in its computations

So far we've discovered that models encode w, either linearly or nonlinearly, and found relationships between model size, performance, and encoding strength. But how can we ensure that the model is using w in its computations? For an intermediate to be meaningful, it must be interpretable and used by the model for computations. Transformers have been shown to encode intermediates that are not used in their computations (Ravichander et al., 2021), and we want to ensure w is not just encoded in some small, insignificant part of the residual stream and is instead used by the model.

Reverse Probing We set up probes going from $[w, w^2] => HS$, as opposed to $HS => f(w)$, and show the degree that hidden states can be explained by w in Fig. 4. Often, we find that w can explain a large amount of variance in a single self-attention, implying that these self-attentions are being dedicated to representing w. We take this

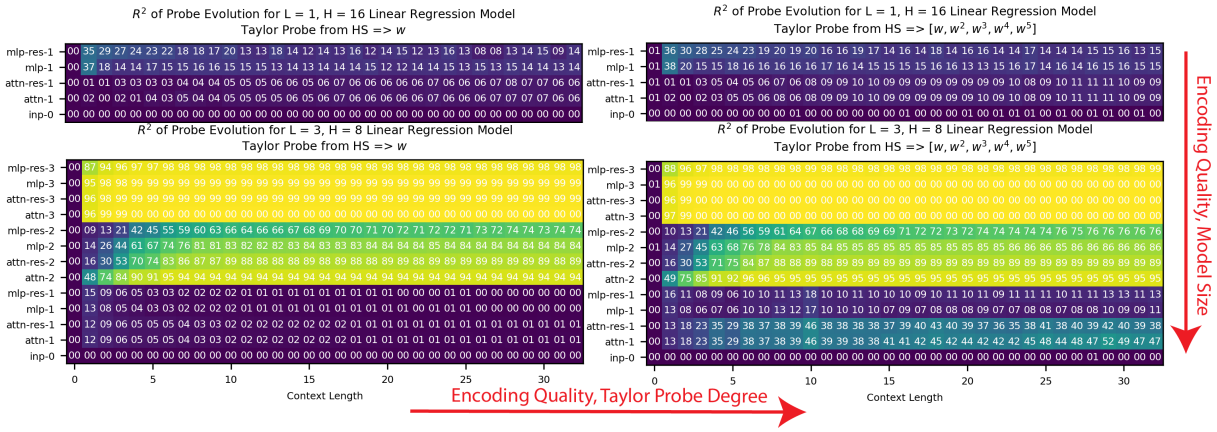

Figure 1: We plot the R^2 of Taylor probes for the intermediate w within models trained on the task $Y = wX$. We see that larger models have w encoded, often linearly, with little gain as we move to higher degree Taylor probes, while small models do not have w encoded. The encoding of w generally improves then plateaus. This provides initial correlational evidence that some models use w in their calculations.

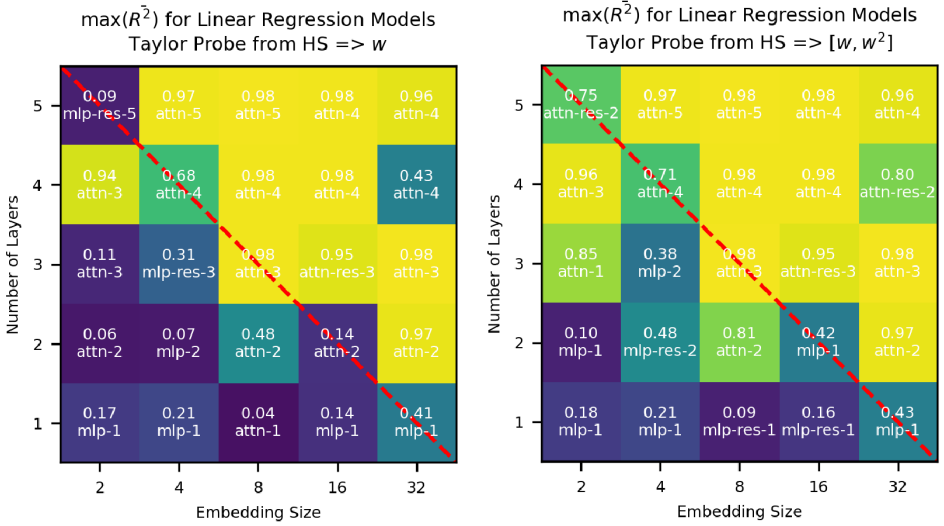

Figure 2: We calculate the mean of the R^2 of probes for $f(w)$ across all layers of the transformer and annotate each model with its highest mean score, $\max(\bar{R^2})$. When $f(w)$ is linear (left) and quadratic (right), we observe a striking phase transition of encoding based on model size, demarked by the red dashed line. If w is encoded, it is mostly encoded linearly, with the $(L, H) = (5, 2), (4, 32), (2, 8)$ models showing signs of a quadratic representation of w. We do not see any meaningful gain in encoding when extending the Taylor probe to degree $n > 2$. For models where $f(w)$ is well represented, it often happens in an attention layer. This is possibly because the attention layer aggregates all past estimates of $f(w)$ into an updated estimate.

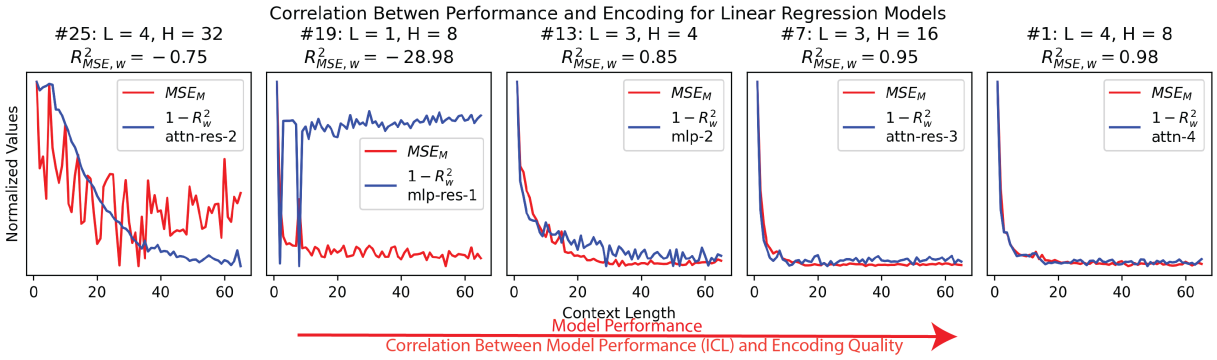

Figure 3: We test the correlation between model performance and the encoding of w on 5 of our 25 models of evenly spaced performance quality. In red, we plot normalized values for $1 - R^2$ of the probe for $HS => [w, w^2]$ at the best position (specified in Fig. 2), and in blue, the mean squared error (MSE_M) for these models. We find that the ability of the best-performing models to in context learn is highly correlated with their encoding of w ($R^2(MSE, w)$ trends to 1). If the model was truly using w, this would make sense, as its ability to predict the data should be strongly dependent on its "understanding" of w. This is stronger correlational evidence that some models use w in their calculations.

as evidence that w is being used by the model - otherwise, it's unclear why such a valuable part of the model would be dedicated to storing w.

Intervening We can generate even stronger evidence by using the reverse probes to intervene on the model and predictably change its output from $w => w'$. For example, in Fig. 5 we attempt to make $w' = 0.5$ for all series, and then measure the observed w from the models' outputs (e.g. $\hat{w}_i = \hat{y}_i / x_i$). For the best-performing models, the intervention worked, providing direct proof that the model uses its internal representation of w in computations. For models where we identified a quadratic representation of w, we see that $w = 0.5, -0.5$ are both represented in the observed intervention.

Putting it all together While it is intuitive that a transformer would simply calculate w and use it to compute $Y = wX$, we can generalize our understanding of intermediates from linear regression to create a framework to show that a transformer uses a method g with associated, unique intermediate I in its calculations:

1. If a model uses a method g, its hidden states should encode I (shown in Fig. 1).

2. If a model uses a method g, model performance should improve if I is better represented (shown in Fig. 3).

3. If and only if the model uses g, we expect some hidden state's variance to be almost fully explained by I (shown in Fig. 4). If g wasn't used, it doesn't make sense for a high degree of a hidden state's variance to be explained by I.

4. If and only if the model uses g, we can intervene on hidden states to change $I => I'$ and predictably change the model output from $g(X, I) => g(X, I')$ (shown in Fig. 5)

These are increasingly powerful methods of proof, with the first two being correlational and the last two being causal. Note that the ideal method of proof is finding the execution of g by cracking open the weights of a transformer and analyzing its computations directly. However, this is extremely time intensive, and using the described coarse-grained approach allows researchers to test many hypotheses for g quickly and effectively.

Thus, we have a general method to show that a model is using intermediate I associated with method g.

5 Future Work

While we have strong correlational and causal evidence that the transformer is using w in its computations to model linear regression, we believe that the true power of our work is using our framework of intermediates to understand how transformers model more complex tasks. For example, we can investigate how transformers model systems governed by linear ordinary differential equations. Humans have both analytical and numerical methods to model linear ordinary differential equations, and we could define unique intermediates for various methods to determine which method(s) the trans-

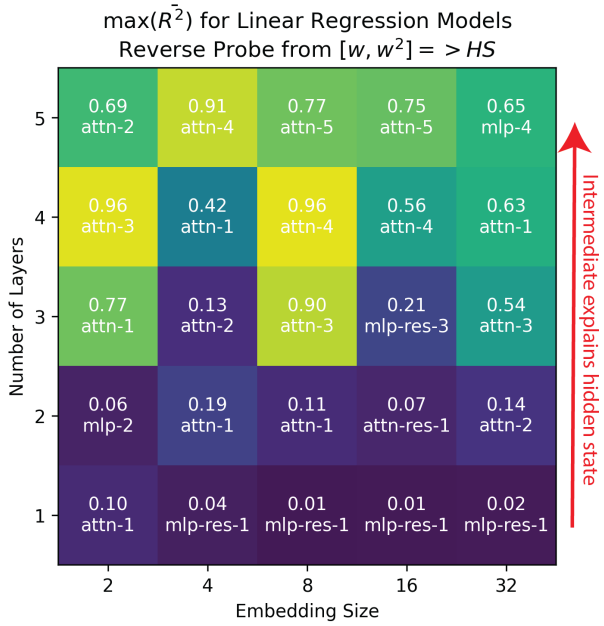

Figure 4: The first form of causal evidence we have for use of w use reverse probes. We plot $\max(\bar{R}^2)$ of the reverse probe from $[w, w^2] => HS$ across all models, and find that the intermediate can explain significant amounts of variance in model hidden states.

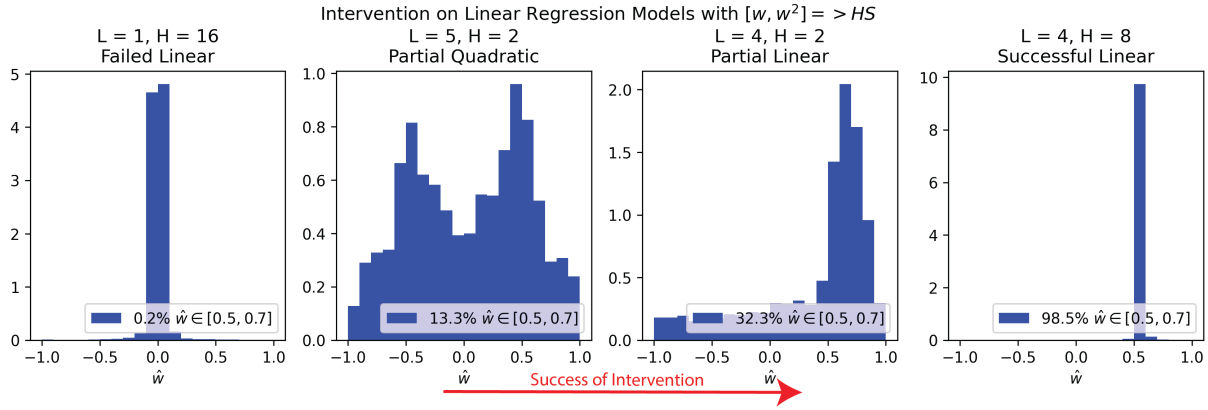

Figure 5: Right: We intervene using reverse probes to make all models output $w' = 0.5$. This intervention can either fail (16/25), be partially successful nonlinearly (2/25) or linearly (3/25), or be successful (4/25). From Fig. 2, we see that the $L = 5, H = 2$ model has a quadratic representation of w, which explains why we see a quadratic pattern in its intervention ($w = -0.5, 0.5$ are both represented). We would expect interventions on hidden states where more variance can be explained to be more successful, but both the $(L, H) = (4, 2), (4, 8)$ models have equal variance explained by intermediates but are considerably different in intervention success (right two panels). Intervening is the strongest causal proof that our model is using w in its computations.

formers are using. Due to time constraints, we leave this to future work.

6 Conclusion

In this work, we have developed a framework for understanding how transformer models perform mathematical computations by identifying and interpreting meaningful intermediate quantities. Focusing on the task of linear regression, we have shown that transformers robustly encode the key intermediate - the slope of the regression line - in their hidden states, often in linearly or quadratically accessible forms.

We found that a model's ability to learn the task is strongly tied to the quality of its intermediate representations, suggesting that transformers' success hinges on learning and manipulating interpretable computations. Using the techniques of reverse probing and representational interventions, we provided causal evidence that these encoded

intermediates are used directly in the model's calculations.

Our findings demonstrate that an "intermediate-centric" approach to interpretability can yield valuable insights into the inner workings of black-box transformer models. By identifying the right intermediates and probing for their representations, we can develop a clearer understanding of how transformers perform complex computations.

Looking forward, our framework can be extended to investigate a wide range of tasks beyond linear regression. By defining unique intermediates for different computational methods, we can uncover the specific algorithms and techniques employed by transformers in domains such as differential equations, optimization, and symbolic mathematics.

Ultimately, we believe that the systematic study of interpretable intermediates is a promising path toward developing more transparent, explainable, and trustworthy machine learning systems. As transformers continue to push the boundaries of artificial intelligence, a deeper understanding of their computational mechanisms will be crucial for ensuring their safe and responsible deployment in real-world applications.

Acknowledgments

- We are grateful for the advice of Ziming Liu and Max Tegmark.
- We would like to acknowledge the use of AI language models, specifically for assisting in the proofreading and grammatical improvement of the final draft of this paper.
- We used MIT Supercloud to run some experiments.

References

2007. *Canonical Correlation Analysis*, pages 321–330. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. *Preprint*, arXiv:2211.15661.

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes. *Preprint*, arXiv:1610.01644.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *arXiv preprint*.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2023. What can transformers learn in-context? a case study of simple function classes. *Preprint*, arXiv:2208.01066.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Iro Laina, Yuki M. Asano, and Andrea Vedaldi. 2022. Measuring the interpretability of unsupervised representations via quantized reverse probing. *Preprint*, arXiv:2209.03268.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Oscar Pastor-Serrano and Zoltán Perkó. 2022. Learning the physics of particle transport via transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12071–12079.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? *Preprint*, arXiv:2005.00719.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. *Preprint*, arXiv:2404.07544.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. 2023. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *arXiv preprint*.