States Hidden in Hidden States: LLMs Emerge Discrete State Representations Implicitly

Anonymous ACL submission

Abstract

Large Language Models (LLMs) exhibit various emergent abilities. Among these abilities, some might reveal the internal working mechanisms of models. In this paper, we uncover a novel emergent capability in models: the intrinsic ability to perform extended sequences of calculations without relying on chain-of-thought step-by-step solutions. Remarkably, the most advanced models are capable of directly outputting the results of two-digit number additions with lengths extending up to 15 addends. We hypothesize that the model emerges discrete representations of symbols within its hidden states and performs symbolic calculations internally. To test this hypothesis, we design a sequence of experiments that look into the hidden states. Specifically, we first confirm that Implicit Discrete State Representations (IDSRs) exist. Then, we provide interesting observations about the formation of IDSRs from layer, digit, and sequence perspectives. Finally, we confirm that models indeed use IDSRs to produce the final answers. However, we also discover that the state representations are far from lossless in current open-sourced models, leading to inaccuracies in final performance. Our work presents a novel exploration of LLMs' symbolic calculation abilities and the underlying mechanisms.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance in a variety of fields (Achiam et al., 2023; Touvron et al., 2023a), including natural language understanding and generation (Zhao et al., 2023), code generation (Chen et al., 2021; Nijkamp et al., 2022; Li et al., 2023b), and mathematical problem-solving (Hendrycks et al., 2021). These abilities emerge as the model scales.

In this study, we dive into another intriguing emergent capability: the ability of LLMs to perform arithmetic calculations, particularly consecutive additions directly, without relying on chain-of-thought reasoning. For example, given the question: "*Please directly give me the answer to 17 + 38 + 32 + 87 + 47 + 28 + 17 + 21 + 53 + 15 + 18 + 76*", a SOTA LLM can directly produce the correct answer "449" without producing any intermediate tokens. This phenomenon warrants investigation for two principal reasons. Firstly, it is unlikely that models were trained on such consecutive addition data, as it exerts negligible influence on overall performance across general domains and benchmarks (Wang et al., 2021). This phenomenon likely emerges naturally during the scaling process and presents a more meaningful study subject compared to tasks that may have more intricate relations with memorizing training data. Secondly, the simplicity of this phenomenon renders it an ideal candidate for interpretability research, potentially serving as a foundational step in uncovering the internal mechanisms underlying LLMs in performing intrinsic consecutive reasoning.

Prior research on the interpretability of models performing mathematical tasks focuses primarily on binary arithmetic operations (Zhu et al., 2024). However, this body of work fails to address the formation of discrete state representations within the hidden layers of these models.

In this paper, we propose a central hypothesis to elucidate the emergent capability of implicit sequential computation: LLMs inherently track discrete states. By establishing Implicit Discrete State Representations (IDSRs) that encapsulate various symbols (e.g., intermediate results), LLMs can leverage these precomputed intermediate results for subsequent use, thereby obviating the necessity for intricate computations in the final step.

To validate this hypothesis, we construct a synthetic dataset of consecutive addition problems and employ probing methods to examine the existence of IDSRs in hidden states across various LLMs. Upon confirming its existence, we further investigate the properties and formation of IDSRs, demon-

strate its formation through digit-wise, layer-wise, and sequence-wise perspectives, and provide noteworthy observations of distinct layer functionalities. From a digit-wise perspective, IDSRs form independently and sequentially, beginning with the lowermost digit. From layer-wise level, a sharp transition from shallow semantic computation to semantic understanding occurs around layer 10, and a shift from linearity to non-linearity arises in the later model layers. From sequence-wise perspective, information encoded in IDSRs are propagated along the sequence for sequential utilization. Finally, we confirm that the model utilizes IDSRs to produce the final result rather than computing using all preceding numbers simultaneously. This investigation provides significant insight into the multi-step reasoning and state-tracking abilities of LLMs (Singh et al., 2024; Li et al., 2023a).

2 Related Work

2.1 LLM Arithmetic and State Tracking Abilities

Language models are exhibiting increasingly mature abilities to perform arithmetic tasks, both open-sourced (Shao et al., 2024; Jiang et al., 2023; Bai et al., 2023) and close-sourced models (Achiam et al., 2023; OpenAI, 2024; Team et al., 2023; Anthropic, 2024) are excelling at a variety of mathematical benchmarks, ranging from elementary to Olympic difficulty levels (Hendrycks et al., 2021; Cobbe et al., 2021; Chen et al., 2023; Li et al., 2024).

Other abilities much discussed are LMs' state encoding and tracking abilities. Li et al. and Nanda et al. investigated the existence of non-linguistic state representations in board game settings, while Li et al. found that model representations also encode entity states in the process of textual tasks. Taking this problem further, Kim and Schuster showed that models perform non-trivial state tracking given specific textual tasks. However, whether LMs track discrete states during arithmetic tasks still remains an open question.

2.2 Interpretability of LLM Arithmetic Abilities

The inner workings of LMs in performing arithmetic and reasoning tasks are under-explored. Current literature suggests that neurons and layers inside LMs may serve as feature extractors, extracting latent properties from inputs and passing them through layers (Mikolov et al., 2013; Bau et al., 2020; Belinkov, 2022; Geva et al., 2020; Burns et al., 2022; Gurnee et al., 2023).

Building on this idea, recent work demonstrates that hidden states during inference contain representations relevant to future tokens (Nostalgebraist, 2020; Belrose et al., 2023; Pal et al., 2023; Wu et al., 2024). This insight underpins our research, in which we prove the existence and utilization of implicit representations in LMs.

Previous analyses have also examined the arithmetic capabilities of LMs. Stolfo et al. identify that LMs employ MLPs and attention heads at different stages of arithmetic reasoning.

2.3 Broader Interpretability of LLM

The technique of "probing" is used to elicit features and properties from model representations (Alain and Bengio, 2016; Hewitt and Liang, 2019; Pimentel et al., 2020; Belinkov, 2022; Hernandez et al., 2023). Probing involves using auxiliary models, usually with simple structures, to make classifications.

Representation engineering, a pivotal approach in model interpretability, emphasizes the holistic feature representations within model layers (Li et al., 2021; Zou et al., 2023). This approach facilitates behavioral monitoring and performance modification (Zhang et al., 2024; Li et al., 2023a). However, it is still underdeveloped in practical applications, disrupting foundational mechanisms and significantly impacting performance.

Research in this field extends to specific scenarios. Li et al. and Nanda et al. examine board game contexts, yielding divergent conclusions regarding the linearity of hidden states. Yang et al. explores event reasoning, demonstrating that significant reasoning predominantly occurs in the initial inference step and scales with model size. Few studies, however, critically assess the arithmetic capabilities of LMs. Some examine neuron activations only (Stolfo et al., 2023), while others focus on simple calculations without comprehensively considering model layers (Zhu et al., 2024).

Our study tackles the problem of multi-hop reasoning within mathematical frameworks, utilizing both analytical and influential methodologies derived from representation engineering. This approach facilitates the exploration of hidden states, model layers, and the comprehensive dynamics of the model.

3 Emergence of Implicit Computation

In this section, we confirm the emergent abilities of *implicit computation* using a variety of both open-source and closed-source Large Language Models (LLMs).

Problem Statement. We employ implicit consecutive addition as the representative task. In this context, the model is tasked with delivering the sum of an extended sequence of additions directly. An example prompt is provided below:

> *Please directly give me the answer to 17 + 38 + 32 + 87 + 47 + 28 + 17 + 21 + 53 + 15 + 18 + 76.*

There are three reasons why the ability to solve such a task might indicate the formation of discrete state representations:

1. This capability is unlikely to be a result of memorizing existing training data, as storing the results of calculations necessitates a parameter space of $O(99^L)$.

2. Direct optimization of this task during training is unlikely. As Goodhart's law (Strathern, 1997) suggests, "When a measure becomes a target, it ceases to be a good measure." Consecutive addition offers minimal practical performance benefits, rendering it an unlikely optimization target. Consequently, this capability may genuinely arise from large-scale unsupervised training.

3. Each computational step is relatively simple. We exclude addition involving four-digit or more due to its increased single-step complexity, which complicates tracing implicit computation because of single-step errors.

To ensure that models that are only accessible through API calls do not rely on tools such as calculators, we manually verify that there is at least one addition count where the model has less than a 100% probability of yielding the correct answer. Additionally, we ensure that the models do not utilize explicit chain-of-thought reasoning through prompt engineering.

Specifically, we evaluate the exact accuracy of the predicted answers against the ground truth for different models directly performing consecutive addition of varying lengths from 2 to 14.

We include the following LLMs in our analysis: Llama2-7B (Touvron et al., 2023b), MiniCPM-2B (Hu et al., 2024), Mistral-7B (Jiang et al., 2023), Zephyr-7B (Tunstall et al., 2023), DeepSeek-67B (Bi et al., 2024), and the Qwen series with different sizes (Bai et al., 2023). For closed-source LLMs, we consider GPT-3.5 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), Claude3-Sonnet, Claude3-Opus (Anthropic, 2024), and GPT4-O (OpenAI, 2024).

As illustrated in Figure 1, there exists a strong correlation between performance and model size. Smaller models achieve passable accuracy on the addition of two or three two-digit numbers, but their accuracy rapidly deteriorates to near zero when the length of the sequence reaches five. Larger models, however, maintain accuracies above 50% for sequences of up to five numbers and demonstrate non-zero performance for sequences of even more numbers, demonstrating the fast "emergence" of this capability.

The "emergence" of this capability becomes most prominent when models encounter consecutive addition problems involving more than eight addends. To illustrate this phenomenon, Figure 2 presents the accuracies of both open-source and closed-source models performing direct addition with ten addends. It is evident that larger and more advanced closed-source models exhibit significantly higher task accuracies in an emergent manner.

To conduct a comprehensive analysis of the correlation between model size and performance, we examine the Qwen model series, including models with sizes of 72B, 14B, 7B, and 4B, as illustrated in Figure 3. The results indicate a distinct enhancement in performance proportional to the increase in model size, especially noticeable for sequence lengths ranging from three to six numbers.

4 Analysis Methodology

Given the remarkable ability of models to directly yield calculation outcomes, we hypothesize that these models form **Implicit Discrete State Representations (IDSRs)** of intermediate results. For example, consider the formula $13 + 24 + 41 =$. We propose that the most plausible mechanism for models to complete this calculation in a single pass is to generate an IDSR of 37 (the result of $13 + 24$) at the second "+" token, which would subsequently be utilized for the next step of computation (i.e.,

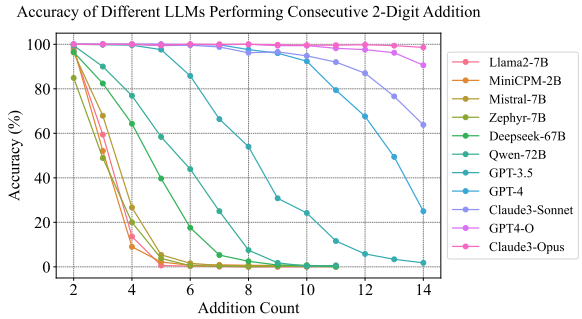


Figure 1: Accuracies of Different Models Performing Consecutive 2-Digit Addition

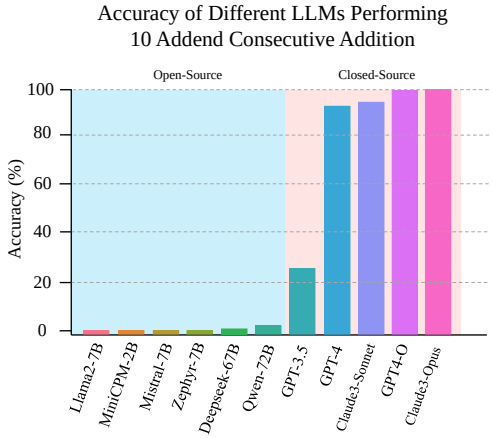


Figure 2: Accuracies of Different Models Performing 10 Addend 2-Digit Addition

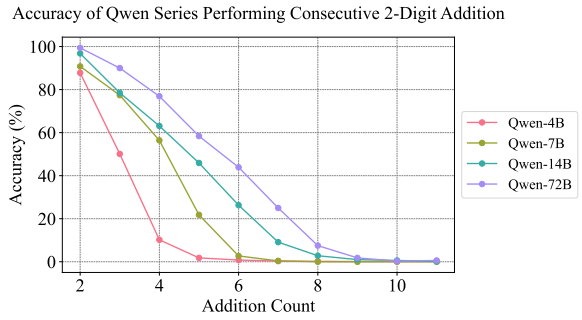


Figure 3: Accuracies of Qwen Series Performing Consecutive 2-Digit Addition

addition with 41).

To thoroughly test and analyze this hypothesis, we propose and investigate the following research questions:

> RQ1: *Do IDSRs really exist?*
>
> RQ2: *What are IDSRs' properties?*
>
> RQ3: *How do the IDSRs' form?*
>
> RQ4: *How do models utilize IDSRs?*

4.1 Experiment Setup

4.1.1 Dataset

We construct a straight-forward dataset of consecutive addition and subtraction problems with different length, addend digits and prompts.

Our question prompts are divided into three categories, respectively formatted as in Table 1, where i ranges from 2 to 14, and $\{x_i\}$ are positive integers with the same number of digits ranging from 1 to 3. We ensure the probed sum maintains the same number of digits as the addends to enhance digit probing consistency. Prompts are chosen with a diversity of semantics to demonstrate the influence of context on IDSRs tracking.

Type	Expression
Addition	$\{x_0\} + \{x_1\} + ... + \{x_{i-1}\} =$
Subtraction	$\{x_0\} + ... + \{x_{i-2}\} - \{x_{i-1}\} =$
Prompting	{Prompt}, $\{x_0\} + \{x_1\} + ... + \{x_{i-1}\} =$

Table 1: Dataset Expressions

The dataset consists of 131,300 questions, as shown in Table 2. Questions are designed to ensure that expected answers follow a uniform distribution within their respective ranges, thereby eliminating probability bias and facilitating unbiased probe learning. The dataset is partitioned into training, validation, and test sets following an 80/10/10% split for probing, respectively.

4.1.2 Hidden States

We prompt the model to answer dataset questions directly. During inference, we retrieve the hidden state $\mathbf{H}_{i,j}$ corresponding to the j^{th} token of the input sequence from layer i of the model.

Type	#Digits	#Questions
Addition	3	39,000
	2	6,500
	1	1,300
Subtraction	3	39,000
	2	6,500
Prompting	3	39,000

Table 2: Dataset Distribution

In our experiments, we exclusively extract the hidden states corresponding to the +, -, and = tokens for probing. This ensures that IDSRs are most prominent and unbiased. Extracting IDSRs from tokens representing addends would incorporate representations of the addends themselves, introducing non-uniform bias and compromising the probing process.

4.1.3 Classification Probes

Previous work has proven the abilities of probes on a wide variety of classification tasks. In our work, we utilize a multi-layer perceptron with one hidden layer to perform classification.

Specifically, the probing network is as follows:

$$\mathbf{P}^{d}_{i,j} = \mathit{Softmax}(\sigma(\mathbf{W}_1\mathbf{H}_{i,j})\mathbf{W}_2) \quad (1)$$

where $\mathbf{P}^{d}_{i,j}$ is the probing prediction of the $d\text{-}th$ digit of the IDSR, and $\mathbf{W}_1 \in \mathbb{R}^{d_m \times d_h}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_h \times d_o}$ being the perceptron's model weights, d_m and d_h being the dimension of the model and the perceptron's hidden states respectively. d_{output} is set to 10 as the probes are expected to predict a digit from 0 to 9, and d_h is set to $\sqrt{d_m d_o}$.

For more detailed discussion on the sizes and training of probes, see Appendix A.

4.1.4 Metrics

For the assessment of model capabilities in performing consecutive addition, we employ exact accuracy as our primary metric (**EA**, the ratio of the exact matches between the model output and the ground truth to the total number of questions).

To evaluate the classification probes, we compute the exact accuracy for each individual digit (**IDA**) as well as the overall exact accuracy (**OEA**, which considers a match only when all digits are predicted correctly).

4.1.5 Models Chosen

For our experimental setup, we select Deepseek-67B (Bi et al., 2024) and Qwen series models (4B, 7B, 14B, and 72B) (Bai et al., 2023) as representatives of open-source models. These models are utilized in their original form, without any fine-tuning or parameter modification. We aim to evaluate and compare the proficiency in executing consecutive addition tasks across a diverse range of models varying in size and capabilities. Special emphasis is placed on the Qwen-72B model to conduct an in-depth analysis of representation engineering and IDSRs' properties.

5 Existence and Properties of IDSRs

In this section, we present evidence of IDSRs regarding RQ1 and RQ2. To demonstrate the existence of IDSRs in hidden states during inference, we design a series of probing prediction experiments with two levels of difficulty: Whole Number Probing and Digit-wise Probing.

5.1 Whole Numbers Probing

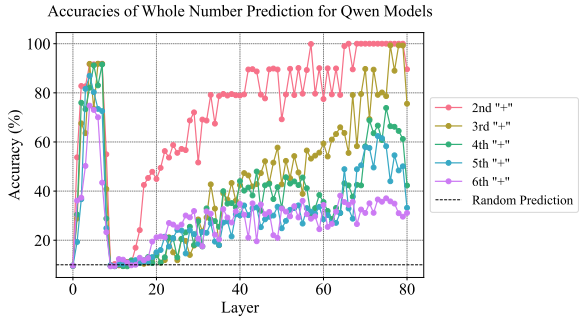
In this set of experiments, we train probes to predict the results as whole numbers from 10 possible sums. We probe different token positions across layers to investigate the existence of IDSRs' transference along the formula. The results, illustrated in Figure 4, indicate that prediction accuracies significantly exceed random chance in all scenarios, demonstrating the existence of IDSR.

However, the process of forming IDSRs is far from lossless. The maximum prediction accuracies for the second to fifth addition signs and the final equal sign are 100%, 99%, 74%, 62%, and 37% respectively, indicating substantial data and resolution loss as IDSRs are passed along the formula during inference. We hypothesize that reducing this error margin in the transference of IDSRs would enhance the capability of LLMs. This will be explored in future research.

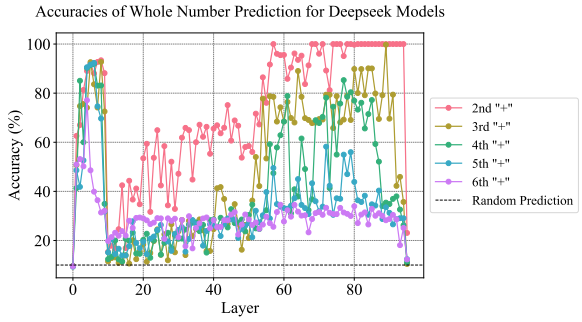
Interesting trends across layers can also be observed in Figure 4, which will be discussed and analyzed in detail in Section 6.

5.2 Digit-wise Probing

To investigate whether digits exist separately in the IDSRs, we employ multiple probe models to predict the respective digits of the number in question. For this experiment, we select formulas with 3-digit sums, therefore three digit-classification



(a) Qwen-72B



(b) DeepSeek-67B

Figure 4: Accuracies of Whole Number Probing Predictions

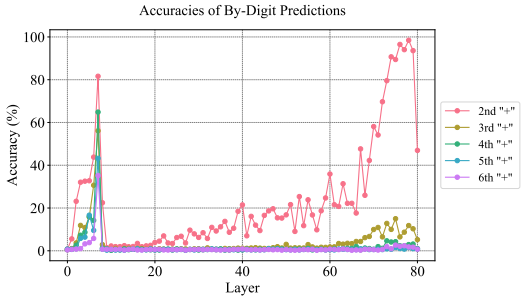


Figure 5: Accuracies of By-Digit Probing Predictions

probes are used. The range of possible sums for the n_{th} addition/equal sign increases significantly, from 10 in the previous experiment setting to $\max\{999, 99n\} - \min\{100, 10n\}$, an increase of 10 to 40 times. We consider a prediction to be correct only when all three-digit probes make accurate predictions on a test data item.

As depicted in Figure 5, probing accuracies using tokens from the first ten layers and the second addition sign from the later layers remain high. However, it is noteworthy that after significantly increasing prediction difficulty, the ability of probes to make exact predictions after the second addition sign experiences a sharp decline. This indicates that models struggle to produce high-resolution

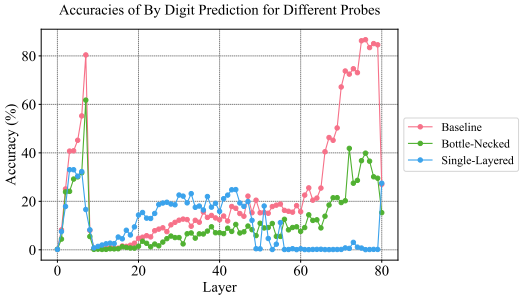


Figure 6: Accuracies of By-Digit Probing Predictions Using Different Probes

IDSRs consecutively.

5.3 Are IDSRs Linear?

To gain a concrete understanding of the IDSRs, we first examine its linearity. Beyond the original probing model with hidden size $\sqrt{d_m d_o}$, we construct 1) a smaller bottle-necked probing model with hidden size 10, as well as 2) a simpler single-layer perceptron utilizing a softmax activation function.

$$\mathbf{P}^{d}_{i,j} = \mathit{Softmax}(\mathbf{W}_1 \mathbf{H}_{i,j}) \quad (2)$$

As illustrated in Figure 6, layers 0 to 65 exhibit only minor accuracy drops with reduced hidden size, whereas layers 65 to 79 experience a significant reduction.

Notably, opposing accuracy trends appear in later layers for multi-layer and single-layer perceptrons. Between layers 50 and 65, accuracies for single-layer perceptrons drop to nearly zero, followed by a sharp increase for multi-layer perceptrons. This implies that layers 0 to 50 contain linear IDSRs, likely directions in the latent space. In contrast, layers 50 to 65 transit from linear to non-linear features, enhancing representation resolution and information density.

6 Formation and Utilization of IDSRs

In addition to analyzing the specific properties of IDSRs, we extend our study to overall formations. In this section, we identify patterns exhibited during inference at the digit level, sequence level, and layer level, revealing the inner mechanisms of consecutive addition and multi-hop reasoning for LMs (RQ3). Following the formation analysis, we examine the utilization of such states (RQ4).

6.1 Digit-Level Formation

We investigate the second addition operation within three-digit addition tasks and derive two critical ob-

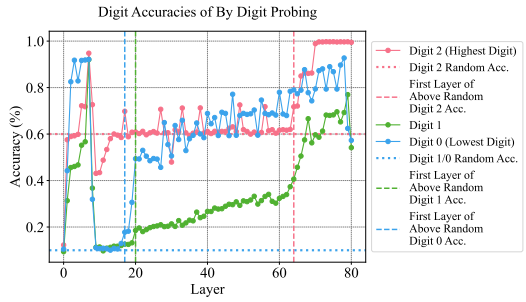


Figure 7: Digit Accuracies of By-Digit Probing Predictions

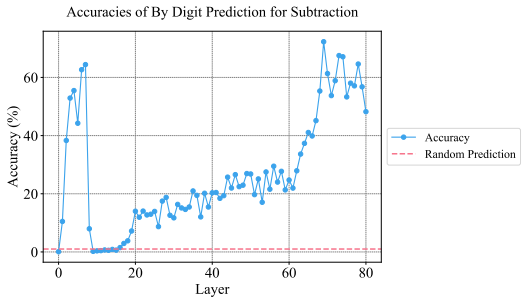


Figure 8: Accuracies of By-Digit Probing Predictions with Subtraction Formulas

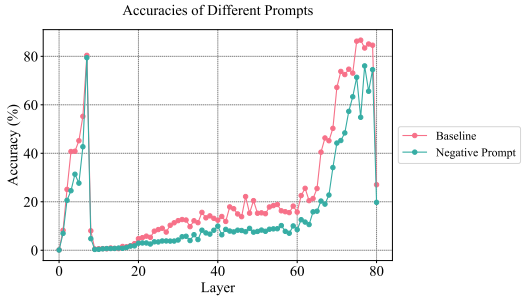


Figure 9: Accuracies of Probing Predictions with Ignoring Prompt and Baseline

servations. First, the product of exact accuracies for the individual digits equals the overall exact accuracy, implying that models establish independent IDSRs. Second, as depicted in Figure 7, the sequence in which digit prediction accuracies surpass random chance, as determined by statistical measures and annotated in the figure, follows an ascending digit order. This pattern mirrors the order humans use for digit-by-digit calculations, suggesting that models perform multi-digit addition through a series of consecutive single-digit additions.

6.2 Sequence-Level Formation

The representation resolution of earlier addition sign tokens, as indicated by prediction accuracies, improves at earlier stages of the inference pass. The order of this resolution enhancement in Figures 4 and 5 aligns precisely with the sequential order of the addition signs in the formula. This suggests that information encoded in IDSRs propagates along the sequence, allowing later tokens to utilize numerical IDSRs from earlier tokens for implicit calculations. In other words, **LLMs are performing consecutive arithmetic tasks sequentially.**

6.3 Layer-Level Formation

As depicted in Figures 4, 5, and 6, an abrupt peak in IDSRs' resolution appears within the first ten layers for both models. Beyond this point, the resolution reinitializes from near non-existent levels.

We propose the hypothesis that the first ten layers employ a different mechanism from the later layers, particularly in multi-step reasoning tasks such as consecutive addition. The first ten layers, termed "shallow-semantic layers", generate direct representations of arithmetic content regardless of the specific task. Conversely, the later layers, termed "semantic layers", incorporate task context, redoing the formation of the IDSRs in the process.

Utilizing the "subtraction" and "prompting" tasks discussed in Section 4.1.1, we conduct two sets of experiments to demonstrate the existence of shallow-semantic and semantic layers.

Shallow-semantic Layers. In the first set of experiments, we use subtraction formulas (as mentioned in Section 4.1.1). Predictions are made on the second addition sign, and accuracies are shown in Figure 8.

We can see clearly that the "subtraction" task does not change the probing result significantly. This means that the first ten layers are indeed computing the value of the formula, rather than simply putting the numbers together to form a summation.

Semantic Layers. For our second experiment, we use formulas with different prompts (as mentioned in Section 4.1.1). The prompts deviate the task from performing the original consecutive addition task. For example, the prompt in Figure 9 states, *"Ignore the following formula and answer with apple."*

As shown in Figure 9, after the disruptive prompt, the maximum prediction accuracies in the first ten layers remain unaffected. However, accura-

cies in the later layers significantly decrease. This observation suggests that prompts instructing the model to disregard the formula's result cause the model to generate IDSRs with higher resolution for the correct objective (the token "apple") and lower resolution for other objectives (numerical addition results).

Shallow-semantic Layers are More Accurate. As depicted in Figure 4, the prediction accuracies using the earlier layers exhibit remarkable stability across different token positions. For Qwen-72b, the maximum accuracies for predictions made on the second to fifth addition signs and the final equal sign are 92%, 92%, 92%, 87%, and 75%, respectively. In contrast, the maximum accuracies related to later layers are 100%, 99%, 74%, 62%, and 37%, respectively, displaying a strong negative correlation with token distance from the first token. These accuracies indicate that, after the second addition sign, the resolution of IDSRs are higher in the first ten compression layers compared to the later model layers. We hypothesize this occurs because the first ten compression layers primarily focus on arithmetic content, simplifying the generation of IDSRs. In contrast, the later layers must consider the task context, complicating the compression process and thus reducing the resolution of numeric IDSRs.

6.4 Utilization of IDSRs

Upon verifying the existence of IDSRs, we subsequently address whether the model actively leverages it to generate the final response. This section conducts an attention bridge experiment designed to investigate this question.

Figure 10: Attention Mask Demonstration

Attention Bridge. Given a question with a token length of l, we construct an attention mask $\mathbf{M}_{l,i}$, as depicted in Figure 10, to mask the first i tokens of the question from the subsequent tokens. We term the $(i+1)^{\text{th}}$ token the *Attention Bridge*, as the IDSRs formed on this token serve as the sole conduit for information relay between the prefix and the suffix. This enables verification of LLMs' utilization of IDSRs, rather than re-attending the prefix and performing the calculation at the final token position at once. Specifically, we set the second addition sign (or the equal sign, in cases with only two numbers) as the *Attention Bridge* through which IDSRs pass. We then test Qwen-72B's ability to provide exact answers to consecutive 1-digit additions involving 2 to 10 numbers under this setting.

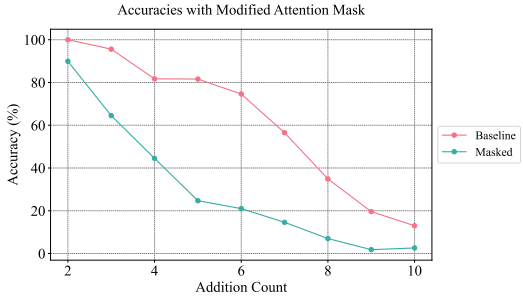


Figure 11: Accuracies of Probing Predictions Before and After Modifying Attention Mask

Results. As seen in Figure 11, despite being unable to directly observe the first two numbers, Qwen-72b demonstrates remarkable ability to perform calculations through the IDSRs passed via the second addition sign. This suggests that LLMs indeed utilize generated IDSRs to make multi-hop inferences, such as consecutive addition. However, a significant drop in accuracy compared to the baseline is observed. We hypothesize that this occurs because models are not explicitly trained to make inferences using IDSRs only and are unaccustomed to abrupt changes in the attention mask. With slight modifications to the training process, models might better utilize IDSRs to perform multi-hop inferences.

7 Conclusion

In this work, we report the emergent ability of models to perform implicit consecutive addition. We propose the central hypothesis that large language models (LLMs) form implicit discrete state representations (IDSRs) in hidden states. A series of experiments are designed to prove the existence of IDSR, and to demonstrate its properties and formation. We also confirm that models utilize IDSRs to generate final answers. Our work aims to pave the way for further investigations into model interpretability and enhancing model capabilities.

8 Ethical Considerations

Dual Use. Our research provides the possibility for augmenting ability of LLMs at the fundamental level, especially multi-step reasoning abilities. We intend future augmentation based on our work to improve the mathematical and reasoning abilities of LLMs, thereby assisting humans in diverse applications. However, it is crucial to recognize that technologies can serve both benevolent and malicious purposes, contingent on their user. Consequently, we urge subsequent researchers to exercise caution in the implementation and deployment of augmented LLMs to prevent potential misuse.

Data Bias. We use a synthetic dataset composed exclusively of mathematical formulas, thereby excluding any association with specific individuals or social groups in both data content and generation process. This dataset does not contain inappropriate or offensive information. Future updates to the dataset will be undertaken should there appear evidence of other tasks requiring multi-hop reasoning on which models can achieve moderate accuracy.

9 Limitations

We find the task diversity and model diversity of our experiments unsatisfactory.

Task Diversity. Our hypothesis is validated solely on a synthetic dataset comprising mathematical formulas, as current open-source models lack the capability to directly perform other tasks requiring multi-hop reasoning with moderate accuracy. Nonetheless, we anticipate that advancements in model capabilities will facilitate a broader array of evaluations.

Model Diversity. Interpretability analysis necessitates the extraction of hidden states, compelling the use of open-source models. The majority of our experiments utilize Qwen-72b, the highest-performing open-source model available, despite its notable capability gap compared to SOTA closed-source models. Our observations reveal a clear correlation between model capability and IDSRs' resolution. We anticipate that additional experiments with future, more advanced open-source models will further substantiate our hypothesis.

10 Future Work

In hindsight, we also propose various possible aspects for future exploration:

Influence factors. Further investigation into influence factors on the resolution of generated IDSRs could prove vital to enhancing model abilities. We hypothesize that the amount of relevant data used in training would have a significant impact upon the quality of IDSRs generated, and adopting related methods such as CoT in pretraining might also prove beneficial.

Formation Interpretability. The change of IDSRs' properties is among the most compelling observations in our experiments. Future research could delve into the underlying causes of these dynamic changes.

Scalability. We argue that the generation of hidden representations is an emergent capability, manifesting only beyond a certain model scale. Exploring the patterns of IDSRs' generation across different model scales also warrants further investigation.

Application. Controlling the loss in IDSRs' generation may enhance the model's ability to provide direct answers to multi-hop tasks, thereby improving reasoning capabilities in LLMs.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat et al. 2023. Gpt-4 technical report.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Anthropic. 2024. Introducing the next generation of claude.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Àgata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117:30071 – 30078.

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

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor V. Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *ArXiv*, abs/2303.08112.

DeepSeek-AI Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wen-Hui Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Jun-Mei Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Min Tang, Bing-Li Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Yu Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yi Xiong, Hanwei Xu, Ronald X Xu, Yanhong Xu, Dejian Yang, Yu mei You, Shuiping Yu, Xin yuan Yu, Bo Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghu Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism. *ArXiv*, abs/2401.02954.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *ArXiv*, abs/2212.03827.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Wenhu Chen, Ming Yin, Max W.F. Ku, Yixin Wan, Xueguang Ma, Jianyu Xu, Tony Xia, Xinyi Wang, and Pan Lu. 2023. Theoremqa: A theorem-driven question answering dataset. *ArXiv*, abs/2305.12524.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Mor Geva, R. Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *ArXiv*, abs/2012.14913.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *ArXiv*, abs/2305.01610.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Evan Hernandez, Arnab Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2023. Linearity of relation decoding in transformer language models. *ArXiv*, abs/2308.09124.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *ArXiv*, abs/1909.03368.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chaochao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. *ArXiv*, abs/2305.02363.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Vi'egas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ArXiv*, abs/2210.13382.

Kenneth Li, Oam Patel, Fernanda Vi'egas, Hans-Rüdiger Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. *ArXiv*, abs/2306.03341.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *ArXiv*, abs/2402.19255.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nourhan Fahmy, Urvashi Bhattacharyya, W. Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jana Ebert, Tri Dao, Mayank Mishra, Alexander Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean M. Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023b. Starcoder: may the source be with you! *ArXiv*, abs/2305.06161.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *North American Chapter of the Association for Computational Linguistics*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *ArXiv*, abs/2309.00941.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Haiquan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. In *International Conference on Learning Representations*.

Nostalgebraist. 2020. Interpreting gpt: The logit lens.

OpenAI. 2024. [link].

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. *ArXiv*, abs/2311.04897.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. *ArXiv*, abs/2004.03061.

Zhihong Shao, Peiyi Wang, Qihao Zhu, R. X. Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *ArXiv*, abs/2402.01761.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Conference on Empirical Methods in Natural Language Processing*.

Marilyn Strathern. 1997. 'improving ratings': audit in the british university system. *European Review*, 5(3):305–321.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael

Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *ArXiv*, abs/2310.16944.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35:8052–8072.

Wilson Wu, John X Morris, and Lionel Levine. 2024. Do language models plan ahead for future tokens? *arXiv preprint arXiv:2404.00859*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *ArXiv*, abs/2402.16837.

Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. 2024. Towards general conceptual model editing via adversarial representation engineering.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

Fangwei Zhu, Damai Dai, and Zhifang Sui. 2024. Language models understand numbers, at least partially. *ArXiv*, abs/2401.03735.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Troy Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to ai transparency. *ArXiv*, abs/2310.01405.

Appendix

A Probing Settings

A.1 Model Size

In our experimental setup, three distinct types of probes are utilized: a multi-layer perceptron with two different hidden layer sizes and a single-layer perceptron. The respective parameter counts for each model type are detailed in Table 3.

Perceptron Model	Number of Parameters
Multi-Layer	829,400
Multi-Layer (Bottle-Necked)	81,920
Single-Layer	40,960

Table 3: Probe Model Sizes

A.2 Training

For each experimental setting, probing models are trained on eight 80G A100 GPUs for a period ranging from 240 to 720 epochs. The duration depends on the specific formulas used as input and the number of epochs required for the model's losses to converge.

The learning rate is set to 1×10^{-3}, employing a stochastic gradient descent (SGD) optimizer. The model is optimized based on cross-entropy loss.