Insights into LLM Long-Context Failures: When Transformers Know but Don't Tell

Taiming Lu[✰] **Muhan Gao**[✰] **Kuai Yu**[✰]
Adam Byerly **Daniel Khashabi**
Johns Hopkins University

Abstract

Large Language Models (LLMs) exhibit positional bias, struggling to utilize information from the middle or end of long contexts. Our study explores LLMs' long-context reasoning by probing their hidden representations. We find that while LLMs encode the position of target information, they often fail to leverage this in generating accurate responses. This reveals a disconnect between information retrieval and utilization, a '*know but don't tell*' phenomenon. We further analyze the relationship between extraction time and final accuracy, offering insights into the underlying mechanics of transformer models. The code is accessible here: https://github.com/TaiMingLu/know-dont-tell.

1 Introduction

The advent of Large Language Models (LLMs), optimized with advanced transformer architectures, has delivered marked improvement in language processing capabilities. These models excel at simultaneously processing extended contexts (Ding et al., 2024; Chen et al., 2023), significantly benefiting various downstream tasks like long-text question answering, summarization, and inference (Wang et al., 2024; Zhang et al., 2024a; Shaham et al., 2022, 2023).

Despite their advanced capabilities, LLMs often struggle to utilize long inputs fully. This tendency, known as positional bias, leads LLMs to disproportionately prioritize information at the beginning or end of the input sequence (Wang et al., 2023) while crucial details in the middle are frequently overlooked (Liu et al., 2023b). Numerous strategies have been proposed to address these biases (Tang et al., 2024; Li et al., 2023; Zhang et al., 2024b), yet the underlying causes and potential solutions remain unclear. This underscores the need for a

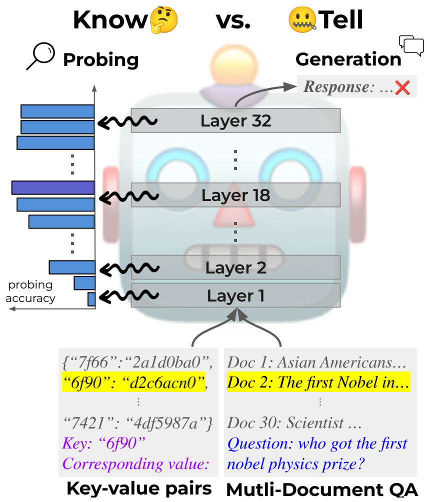

Figure 1: Following prompts by Liu et al. (2023b), for each transformer layer, we train a probing classifier to probe the model's ability to identify useful information. The peak accuracy among layers indicates the effectiveness of the model's processing of context.

deeper investigation into how LLMs handle long-context integration. To fully assess the capabilities of LLMs in handling extended contexts, it is not enough to merely evaluate their final performance: some important information is hidden in models' representations.

In this work, we present a probing analysis of LLMs long-context generalization. Specifically, we build probes based on the internal representation of LLMs for various layers and positions to measure the accuracy of reconstructing the position they correspond to (see Figure 1). A necessary condition for effective long-context processing by LLMs is their ability to encode positional information in their intermediate representations.

We conduct experiments on two tasks from Liu et al. (2023b) and three recent open-source models. Our findings reveal a gap between the accuracy of LLMs' predictions and the probes on their representations. Notably, while LLMs can accurately

✰ indicates equal contributions.
Emails: {tlu37, mgao38, kyu25}@jhu.edu

identify the position of crucial information within the context, they often fail to utilize this information effectively in their responses, leading to what we term the '*know but don't tell*' phenomenon. To our knowledge, this is the first work to use probing analysis to highlight this observation. We hope that our work on distinguishing "knowing" and "telling" motivates future work on tackling the long-context challenges of LLMs.

In summary, our contributions are as follows: (1) Probing analysis: We introduce a novel framework to investigate the long-context reasoning capabilities of LLMs. This framework allows us to measure how accurately LLMs encode positional information across various layers and positions within their intermediate representations. (2) Empirical evaluation: We conduct comprehensive experiments using tasks from Liu et al. (2023b) and three recent open-source models. Our empirical evaluation provides new insights into the positional biases of LLMs and their impact on model performance. (3) 'Know but Don't Tell' phenomenon: Our analysis reveals a critical gap between LLMs' ability to encode and utilize positional information. We identify the "*know but don't tell*" phenomenon, where LLMs accurately identify the position of crucial information but fail to leverage this knowledge in generating accurate responses.

We believe these contributions provide a significant step towards understanding and improving the long-context processing capabilities of LLMs. By distinguishing between the encoding and utilization of positional information, our work lays the foundation for future advancements in LLM performance and reliability.

2 Related Work

Positional bias. LLMs exhibit a positional bias, where their performance is influenced by the location of crucial context information (Zhao et al., 2021). One prominent example is the "lost in the middle" phenomenon, where comprehension declines for information in the center of a long context (Liu et al., 2023b). Additionally, recency bias is observed, particularly in few-shot learning scenarios, where models tend to favor information near the end of the prompt (Zhao et al., 2021). Such biases could stem from the positioning of key data in pre-training sets, which often places important elements near critical points (Peysakhovich and Lerer, 2023). Our work delves into this phenomenon by examining the underlying mechanisms within the transformer layers of LLMs.

Probing. Probing classifiers are extensively used to elucidate the inner workings of LLMs (Alain and 解释，说明
Bengio, 2016; Azaria and Mitchell, 2023; Jin et al., 2024; Ju et al., 2024; Templeton et al., 2024). Various works train probes on model representations to assess how well they encode various linguistic features, such as phrase-level, syntactic, and semantic information (Liu et al., 2023a; Marks and Tegmark, 2023; Li et al., 2024). The efficacy of a classifier in a given task indicates the degree to which that layer successfully captures pertinent information. 相关
In our study, we employ probing as a proxy to determine whether the LLMs accurately identify and represent crucial parts of the context.

3 Experimental Setup

We design a layer-wise probing task to examine if the model successfully identifies the target information from the given prompt. We expect that higher probing accuracy signifies a stronger connection between the model's hidden representations and its internal knowledge of the target information.

Datasets and prompts. We follow the datasets and prompts used by Liu et al. (2023b). Our datasets include: (1) Key-Value pairs retrieval (kv-pairs) where the context contains a collection of keys and their corresponding values (128-bit randomly generated UUIDs). The goal of this task is to identify a value given its key. Each prompt for this task contains 100 kv-pairs and a target key. (2) Multi-document question answering (MDQA) where the context contains multiple sets of evidence paragraphs. The goal of this task is to, given a question, identify the relevant document and produce an answer. Each prompt for this task contains 30 documents, and a target question. Given a set of key-value pairs/documents, with only one containing target information, LLM is prompted to output the value/answer given the key/question. Further details on prompt construction can be found in §A.

Probing classifiers. For each input prompt, we collect the last token embedding of *each layer*. We then train separate linear classifiers for each layer would receive the embedding (of the last token) as input and the gold kv-pair/document ID (position among all pairs/documents) as the target output. The classifier minimizes the following objective:

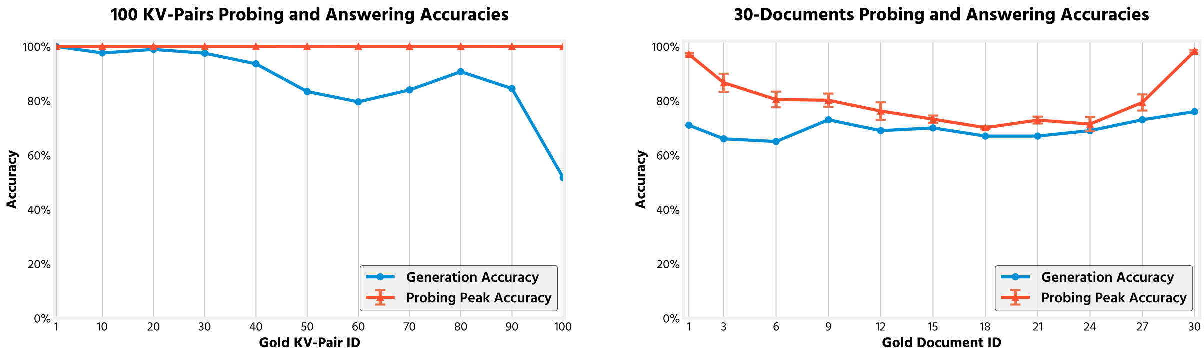

Figure 2: Accuracy of directly generating answers by LLMs (blue line) vs maximum probing accuracy across layers by our probing classifiers (red line). In both tasks, our probing classifiers outperform the model's generated answers, across all the gold positions. **This indicates a discrepancy between *knowing* the context and *using* it.**

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log\left(\frac{e^{\theta_j^\top x_i}}{\sum_{k=1}^{C} e^{\theta_k^\top x_i}}\right),$$

where N is the number of data points, C is the number of different gold IDs, x_i represents the input embedding, y_{ij} is the one-hot encoded label for the i-th data point and j-th ID. Ultimately, this recipe gives one probing classifier for embeddings of *each layer*. Using these models, we show results per layer and across layers.

Models and hyperparameters. We employ `LLaMa3-8B-Instruct` (AI@Meta, 2024) to our probing analysis. Results for two other models are in shown §C, where we at the same conclusion. To reduce uncertainty caused by random initialization, each classifier is trained ten times. In the results, we report the mean and std (error bars) of independent ten experiments.

Metrics. Following Liu et al. (2023b), we use accuracy to quantify the success of our models. Specifically, we quantify accuracy for two types of targets: (a) Generation accuracy quantifies how well LLMs generate correct value in kv-pair retrieval or generate correct answer string in MDQA. (b) Probing accuracy quantifies how accurately classifiers can predict the gold kv-pair or document ID, indicating whether the layers sufficiently encode information from the input context.

4 LLMs Know but Don't Tell

4.1 Experiment: maximum probing accuracy across all LLM layers

We focus on the peak accuracy across all transformer layers as a proxy to determine if the model ever correctly identifies the useful information within the prompt during the forward pass. Specifically, we select the probing classifier with the highest accuracy across all layers. In Figure 2, we show this peak layer probing accuracy. For comparison, we also show the accuracy of LLMs in generating the answer (independent of our probing classifiers).

LLMs know but don't tell. Our results indicate that the model's hidden representations indeed contain information about the location of the target information. Specifically, in kv-pairs setup (Fig. 2; left) there is always a layer such that its probe can near-perfectly identify the location of the correct key-value pair associated with the prompt. This is true, even for instances where the LLM does not return the correct answer or abstains from producing any answer. This suggests a disconnect between the model's ability to locate the information and generate a response based on that information.

A similar trend is also observed for MDQA (Fig. 2; right) where the peak probing accuracy is consistently higher than the direct answer accuracy, indicating the same disconnect from document grounding to response generation. These findings highlight that while the model can recognize and encode the location of relevant information within its layers, this knowledge does not always translate into an accurate generation answer.

4.2 Experiment: probing across per layers

To understand the flow of information across LLMs' layers, we shift our focus to probing classifiers' accuracy across LLM layers. Figure 3 visualizes probing classifier accuracy per layer. For comparison, in both kv-pairs/MDQA setups, we show this accuracy for three positions: target infor-

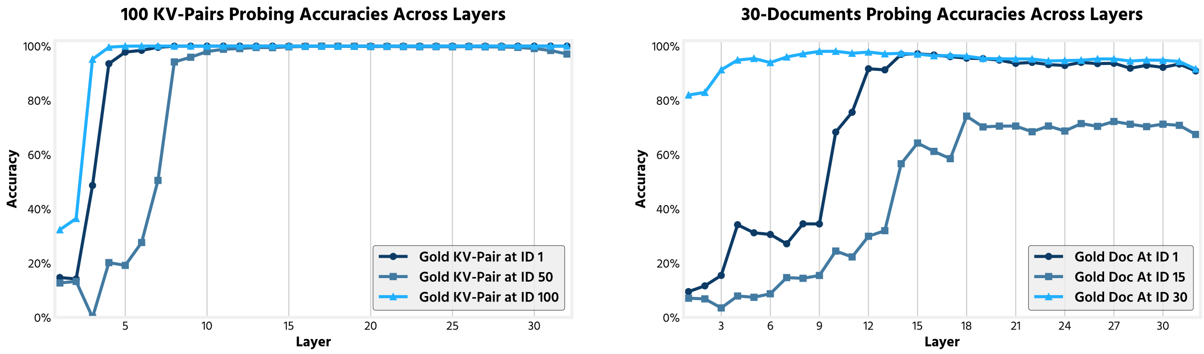

Figure 3: The figures show the probing accuracy for each layer for the two tasks: kv-pairs (left) and MDQA (right). Different colors indicate the position of target information in the input context. In both tasks, **mid-context information requires more layers to be extracted.**

mation at the start, middle, or end of the input.

Mid-context information requires more layers to be located. Our results reveal that LLM locates target information gradually at early layers. Specifically, in the kv-pair setup (Fig. 3; left), probing accuracy consistently increases until it reaches perfect accuracy at layer 13. Notably, when the target kv-pair is at the middle position of the input prompt, LLM requires more layers to locate the target information.

The general trends of the MDQA scenario (Figure 3; right) are similar in principle, but with nuanced differences. The patterns vary significantly with the position of target information. Classifiers perform best when the target document is at the start of the input context, with near-perfect prediction since early layers , and maintain it in subsequent layers. However, it takes more layers for the probing classifier to achieve peak accuracy for the middle and tail gold context. Interestingly, when the target document is in the middle, classifier accuracy decreases after the peak. As MDQA task requires a higher level of reasoning, the model is shifting from locating documents to generating language output.

4.3 Experiment: the number of layers taken for locating target information

Our probing experiments (§4.2) reveal that the model's encoding of target information position initially improves but then degrades as layer depth increases. This motivates, the investigation of the relationship between the number of layers taken by the model to locate target information from the prompt and the LLM's generation accuracy of generating the target information.

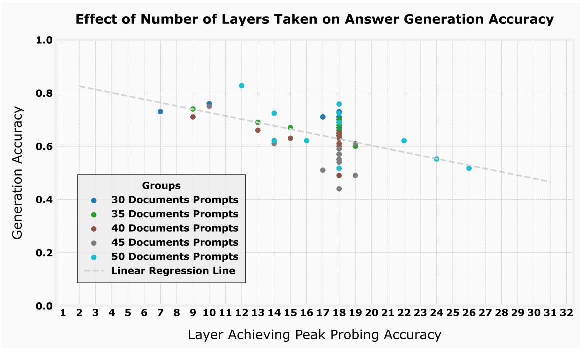

Figure 4: The LLM layer that achieves the peak probing accuracy (x-axis) vs. the accuracy of LLM in generating the correct answer (y-axis). We observe that a *later* peak correlates with *lower* accuracy in the language model's final output. This implies that the earlier an LLM encodes information from a specific index, the higher the accuracy of the final output for that position.

We run additional 35, 40, 45, and 50 multi-document probing tasks. In Fig. 4, for all IDs that achieve probing accuracy greater than 60% (to suppress the outliers), on the x-axis we show the layer at which probing achieves the peak accuracy. On the y-axis we show the the LLM's generation accuracy (no probes involved).

Early-layer information localization leads to higher accuracy in LLM output. As Fig. 4 shows, there is a statistically significant (two-sided t-test) negative correlation between the layer with peak accuracy of locating target information and its final output accuracy ($p < 5e - 5$). This negative correlation implies that the earlier the model identifies the target document within its layers, the more likely it is to generate an accurate final answer.

5 Conclusion

Our study investigates LLMs positional bias, indicating that LLM could capture context information, but do not tell the correct answer. The experiment results demonstrate that the input context is embedded in the model's hidden representation, but such information is not decoded into anticipated output.

6 Limitation

Knowledge of the gold document's location and the ability to cite from it are distinct but connected; the model might know the location but still, fail to integrate it into a coherent and accurate answer. This comparison does not fully capture the nuanced interactions between the model's internal attention mechanisms and output generation capabilities. While these limitations are acknowledged, they do not detract from the core contributions of our work. Our findings provide valuable insights into the positional effects on model performance and highlight the importance of document sequence in information retrieval tasks. By identifying specific areas where the model struggles, we lay the groundwork for future improvements and optimizations in model design and training.

7 Ethical Considerations

Currently the consequences of misinterpretations or errors in long context processing can be significant, in fields like healthcare, legal, and public services. In other cases, LLM long-context failure results in harmful, biased, and misleading generations (Anil et al., 2024). Our research considers the potential negative impacts of these errors and actively works to uncover the mechanisms that could minimize such risk.

References

AI@Meta. 2024. Llama 3 model card.

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

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-shot jailbreaking. *Anthropic, April*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *Preprint*, arXiv:2402.13753.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. 2024. Exploring concept depth: How large language models acquire knowledge at different layers? *arXiv preprint arXiv:2404.07066*.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? a layer-wise probing study. *arXiv preprint arXiv:2402.16061*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2023. Split and merge: Aligning position biases in large language model based evaluators. *Preprint*, arXiv:2310.01432.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. 2023a. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? *arXiv preprint arXiv:2312.03729*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023b. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Alexander Peysakhovich and Adam Lerer. 2023. Attention sorting combats recency bias in long context language models. *Preprint*, arXiv:2310.01427.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2024. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. *Preprint*, arXiv:2310.07712.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, and Trenton Bricke. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024. Novelqa: A benchmark for long-range novel question answering. *Preprint*, arXiv:2403.12766.

Yiwei Wang, Yujun Cai, Muhao Chen, Yuxuan Liang, and Bryan Hooi. 2023. Primacy effect of chatgpt. *arXiv preprint arXiv:2310.13206*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024a. ∞bench: Extending long context evaluation beyond 100k tokens. *Preprint*, arXiv:2402.13718.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2024b. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *arXiv preprint arXiv:2403.04797*.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

A Prompting Details

Following setup by Liu et al. (2023b), we construct key-value pairs retrieval and multi-document question answering prompting dataset.

Key-Value pairs retrieval (kv-pairs) We generate n pairs of 128-bit randomly generated UUID.

> **Example Key-Value pair**
> "7f666c61-573f-4212-a0a9-6f90d487cd4a" : "2a1d0ba0-cfe4-4df5-987a-6ee1be2c6ac0"

The n kv-pairs are composed into one single JSON object. To test at ID k, we choose one pair as gold, insert it at ID k, and then construct as a prompt in the format:

> Extract the value corresponding to the specified key in the JSON object below.
>
> JSON data:
> { "key^1": "value^1",
> "key^2": "value^2",
> ...
> **"key^k": "value^k",**
> ...
> "key^n": "value^n",
> }
>
> Key: "key^k"
> Corresponding value:

Multi-document question answering (MDQA) In the n document setting, we randomly select one question answer pair from the dataset by Liu et al. (2023b). Subsequently we retrieve the document containing this answer and mark it as gold.

> **Example retrieval**
> **Question**: who got the first nobel prize in physics
> **Answer**: Wilhelm Conrad Röntgen
> **Document**: (Title: List of Nobel laureates in Physics) The first Nobel Prize in Physics was awarded in 1901 to Wilhelm Conrad Röntgen, of Germany, who received...

We then sample $n - 1$ distractors, relevant documents that do not contain the answer. To test at ID k, we randomly shuffle the distractors and then insert the gold document at ID k. Example prompt with gold document at ID k is like:

> Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).
>
> Document [1](Title: Asian Americans in science and technology) Prize in physics for discovery of the subatomic...
> ...
> **Document [k](Title: List of Nobel laureates in Physics) The first Nobel Prize in Physics was awarded in 1901...**
> ...
> Document [n] (Title: Scientist) and pursued through a unique method, was essentially in place. Ramón y Cajal won ...
>
> Question: who got the first nobel prize in physics
> Answer:

B Probing Setup

In the experiment described in §3, we employ linear classifiers as our probing method.

For any given task, we choose $\{1, 0.1n, 0.2n, \ldots, 1.0n\}$-th position as gold ID. Following the prompt format in §A, we generate prompts with all chosen IDs, for $10,000$ iterations, resulting in a set of 110,000 prompts.

Each prompt is fed into language model, and the embedding from each layer's last token is collected. For each layer, separately, we have $110,000$ embeddings corresponding to 11 IDs and train a classifier for ten times, with embedding as input and ID as output. We calculate their mean accuracy and standard deviation.

C Experiments Results on `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023) and `Gemma-7b-it` (Team et al., 2024)

We conduct same experiment procedure on additional two models, which produce the same pattern. The experiment is running on one A100 GPU.

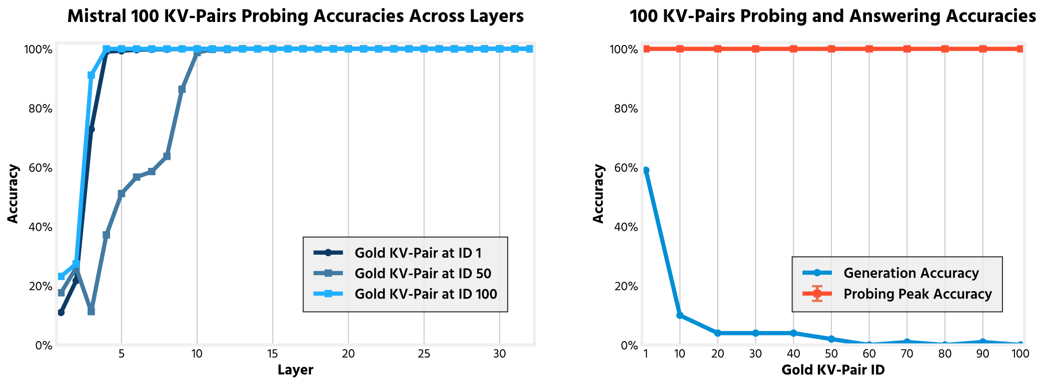

Figure 5: In the left figure, Mistral's probing result in 100 kv-pairs retrieval task resembles with Llama3's in most ways, but there're still a few differences. First, the trends of locating information from the head and end are more similar compared to those in Llama3. Also the model takes more layers than Llama3 to well locate information in middle context. The right figure highlights the significant discrepancy between the probing peak accuracy and generation accuracy, indicating a severe 'know don't tell' phenomenon.

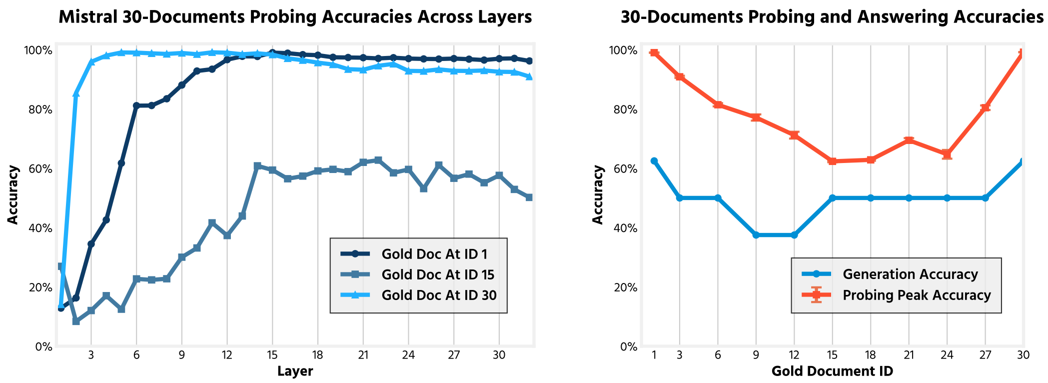

Figure 6: Mistral's layer-wise probing classifier accuracy performs in the same pattern as Llama3. Middle context takes more layers to encode and results in a lower peak accuracy. In right figure, both its peak accuracy and generation accuracy shows a U-shape curve, with still probing consistently outperforms generation.

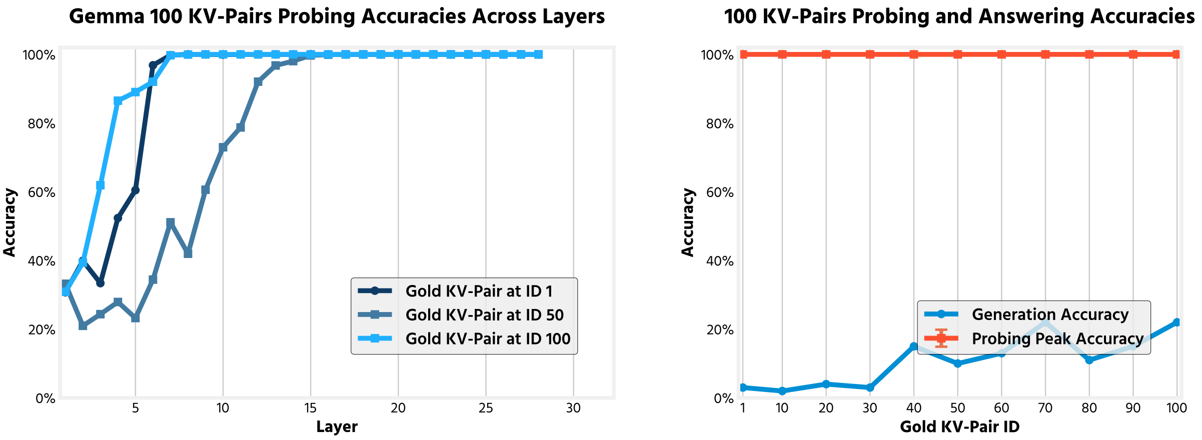

Figure 7: In the left figure, Gemma shows a similar pattern like other models in 100 kv-pairs retrieval task. However, information from all positions is located more slowly than other models: Information in the head or end position of input context is located after 5 layers, while information in the middle of input takes 15 layers to be located. In the right figure, it shows a significant gap between generation accuracy and probing peak accuracy, which is similar with Mistral. This highlights a significant 'know don't tell' phenomenon.

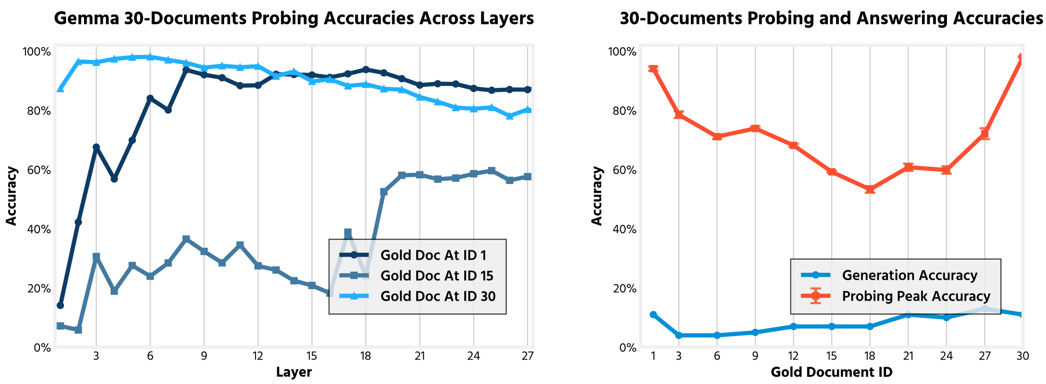

Figure 8: For Gemma MDQA task, although head and end context follows same observation, it middle context shows a sudden decrease in accuracy, indicating a sudden information lost but soon retrieve it back. The right figure follows the same pattern, where generation accuracy is consistently low, disconnecting from the high U-shape probing accuracy curve.