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

Muhan Gao[✩] **Taiming Lu**[✩] **Kuai Yu**[✩]

Adam Byerly **Daniel Khashabi**

Johns Hopkins University

{mgao38, tlu37, kyu25}@jhu.edu

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 its *communication*, a '*know but don't tell*' phenomenon. We further analyze task accuracy vs layers, offering insights into the underlying mechanics of transformer models.[*]

1 Introduction

The advent of Transformer-based Large Language Models (LLMs) has delivered marked improvement in language processing capabilities (Vaswani et al., 2017; Dubey et al., 2024). 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; Goldman et al., 2024), yet the underlying causes and potential solutions remain unclear. This underscores the need for a deeper investigation into how LLMs handle long-context integration. To fully assess the capabilities of LLMs in handling extended
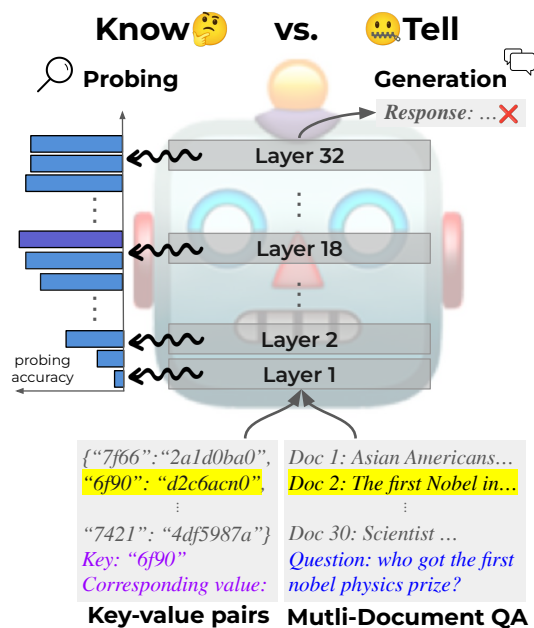

Figure 1: Following prompts by Liu et al. (2023b), we train a probing classifier for each transformer layer to probe the model's ability to identify useful information. The peak accuracy among layers indicates the model's long-context processing effectiveness.

contexts, it is not enough to merely evaluate their final performance: some important information is hidden in models' representations.

This work presents a probing analysis of LLMs' long-context generalization. We develop probes based on LLMs' internal representations for various layers and positions to measure the accuracy of reconstructing the position they correspond to (Fig. 1). Our working hypothesis is that, *for LLMs to effectively process long contexts, they must 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' *generations* and *the probes* on their repre-

---

sentations. Notably, while LLM representations identify the position of crucial information within the context (surfaced via probes), 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 LLMs' long-context challenges.

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. 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 the location of crucial context information influences their performance (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 Bengio, 2016; Azaria and Mitchell, 2023; Jin et al., 2024; Ju et al., 2024; Templeton et al., 2024; Jiang et al., 2024; Sky 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. Our study employs probing as a proxy to determine whether the LLMs accurately identify and represent crucial parts of the context.

3 Experimental Setup

We conduct a layer-wise probing analysis to determine if the model accurately identifies target information from a given prompt. We expect that higher probing accuracy shows a stronger connection between the model's hidden representations and its internal understanding 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, the LLM is prompted to output the value/answer given the key/question. Details (e.g., prompts) are in §A.

Probing classifiers. For each input prompt, we extract the last token embedding x^l from *each layer* l. We then train a separate linear classifier for each layer l, characterized by weight $\mathbf{w}_l$ and bias b_l. The classifier takes x^l as input to predict the gold kv-pair or document ID (position among all pairs/documents). The classifier aims to minimize the following objective:

(a) LLaMa3-8B-Instruct
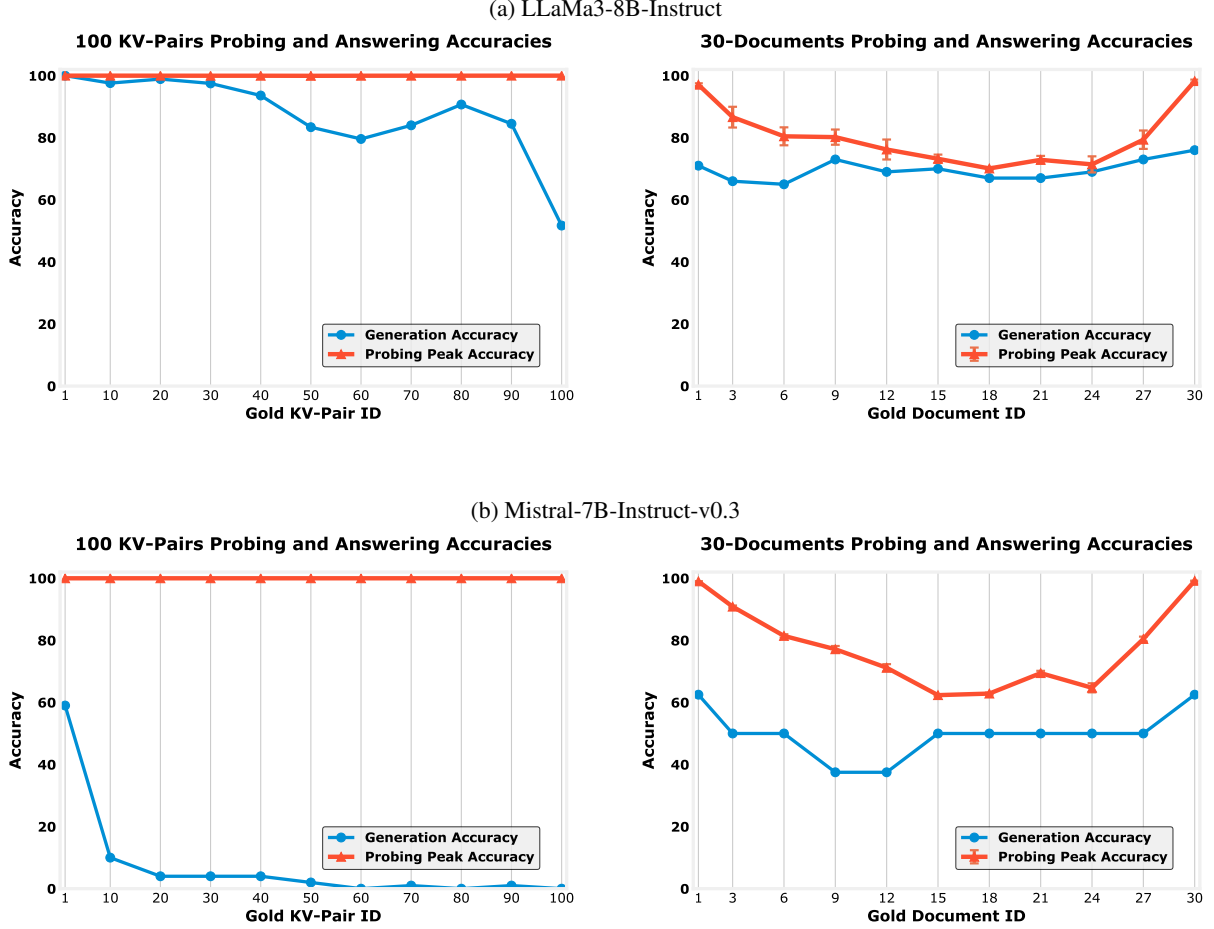
(b) Mistral-7B-Instruct-v0.3

Figure 2: Accuracy of LLMs in directly generating answers (blue line) compared to the maximum probing accuracy across layers by our probing classifiers (red line). In both tasks, our probing classifiers surpass the model's generated answers across all gold positions. **This highlights a distinction between *knowing* the context and *utilizing* it.**

$$J(\mathbf{\Theta}) = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{w_{y_i}^\top x_i^l + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^\top x_j^l + b_j}} \right),$$

where $\mathbf{\Theta} = \{\mathbf{w}_l, b_l\}_{l=1}^{L}$ is the union of all parameters, L is the number of layers, N is the number of data points, C is the number of different gold IDs (classes), x_i^l represents the input embedding of the l-th layer from the i-th data point, y_i is the label for the i-th data point. Ultimately, this recipe gives one probing classifier for embeddings of *each layer*. Using these models, we show results per layer and across layers. We select training and test datasets that are *mutually exclusive* across each other, containing entirely different sets of key-value pairs/documents. Thus, neither classifiers nor the language model had access to the test samples' gold pair/document ID during the evaluation. We provide implementation details in §B.

Models and hyperparameters. We employ two SOTA models, `LLaMa3-8B-Instruct` (AI@Meta, 2024) and `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023) to do our probing analysis. The results for the third model, `Gemma` (Team et al., 2024a), are provided in the Appendix due to space constraints. To minimize uncertainty from random initialization, each classifier is trained ten times. We report the mean and standard deviation (error bars) from these ten independent experiments. As we observe in §4, consistency across the three models confirms that the *'Know but Don't Tell'* phenomenon is not specific to a single model's architecture.

Metrics. Following Liu et al. (2023b), we use *accuracy* to evaluate our models' success. We measure two types of accuracy: (a) Generation accuracy, which quantifies how well LLMs generate the correct value in kv-pair retrieval or the correct answer string in MDQA, and (b) Probing accuracy, which evaluates how accurately classifiers predict

(a) LLaMa3-8B-Instruct

100 KV-Pairs Probing Accuracies Across Layers

30-Documents Probing Accuracies Across Layers

(b) Mistral-7B-Instruct-v0.3

100 KV-Pairs Probing Accuracies Across Layers

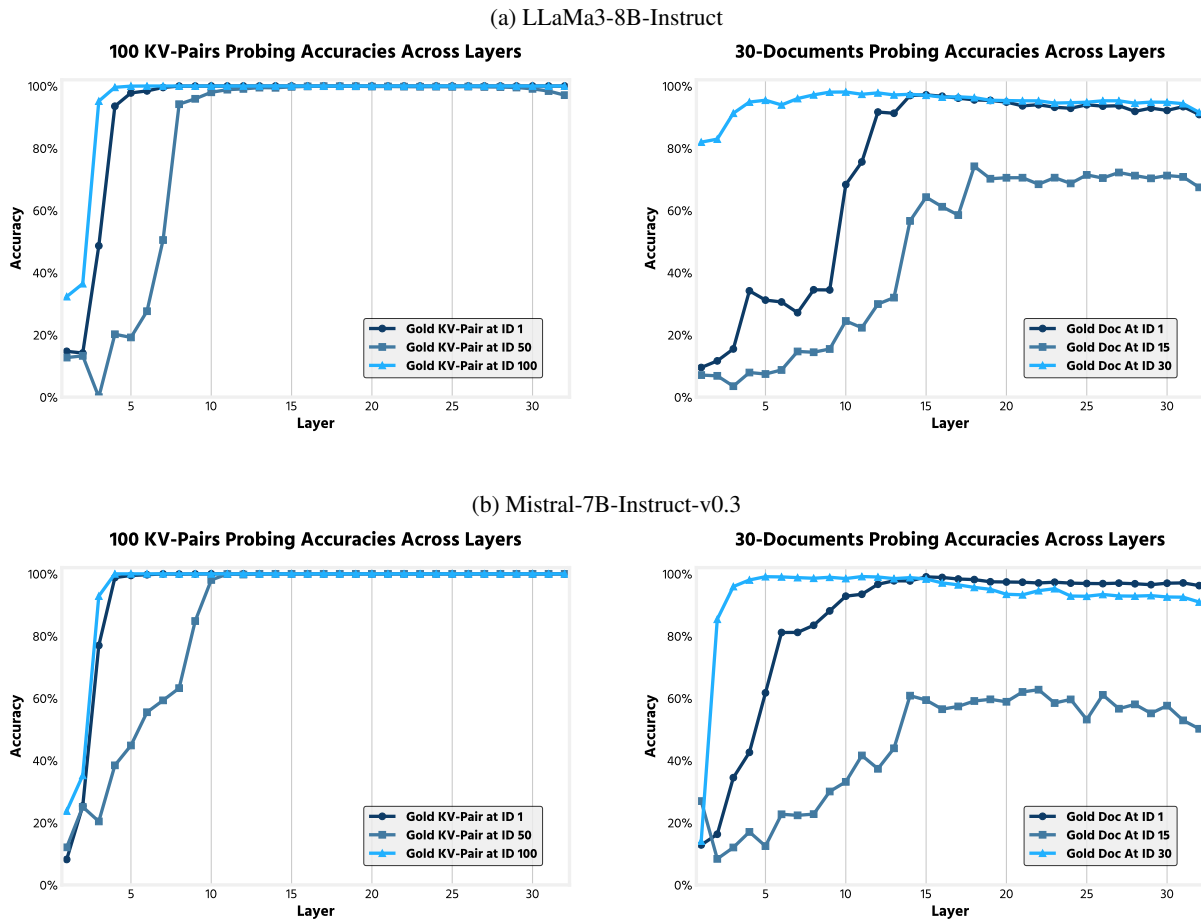**30-Documents Probing Accuracies Across Layers**

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

the gold kv-pair or document ID, quantifying the extent to which layers encode information from the input context.

4 LLMs Know but Don't Tell

4.1 Experiment: Peak Probing Accuracy Across LLM Layers

We examine the maximum accuracy across all transformer layers as an indicator of whether the model successfully identifies key information within the prompt during the forward pass. Specifically, we choose the probing classifier with the highest accuracy among all layers. This peak layer probing accuracy is displayed in Fig. 2. For comparison, we also present 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 *do* contain information about the location of the target information. Specifically, in the kv-pairs setup (Fig. 2; left) there is always a layer where its probe can almost perfectly identify the location of the correct key-value pair associated with the prompt. This holds true even when the LLM provides an incorrect answer or suggests no gold information is present in the input. 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. 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 Layers

We focus on the probing classifiers' accuracy across LLM layers to understand the flow of information across an LLM's layers. Fig. 3 visualizes the probing classifier accuracy per layer. In both kv-pairs and MDQA setups, we show this accuracy for three positions: when the target information is at the start, middle, or end of the input.

Middle-context information requires more layers to be located. Our results reveal that LLMs locate 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 (Fig. 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 or end of the input context, with early layers near-perfect prediction maintaining across subsequent layers. However, it takes more layers for the probing classifier to achieve peak accuracy when the gold information is in the middle or tail of the context. Interestingly, classifier accuracy decreases after the peak when the target document is in the middle.

Related to the results in this section, in §D we present visualizations of LLM intermediate representations using PCA and UMAP dimension reduction. In these visualizations, projected representations corresponding to adjacent gold documents are connected by lines. We observe that as the number of layers increases, particularly in the middle layers, the projected representations are spaced apart and arranged in a path. In contrast, these embeddings are more entangled in the earlier layers. This further corroborates our probing results.

4.3 Experiment: Number of Layers Taken for Locating Target Information

Our probing experiments (§4.2) reveal that the model's encoding of target information position improves with more layers but then it degrades. This motivates, the investigation of the relationship between the number of layers needed by the model to locate target information from the prompt and the LLM's accuracy in generating that target
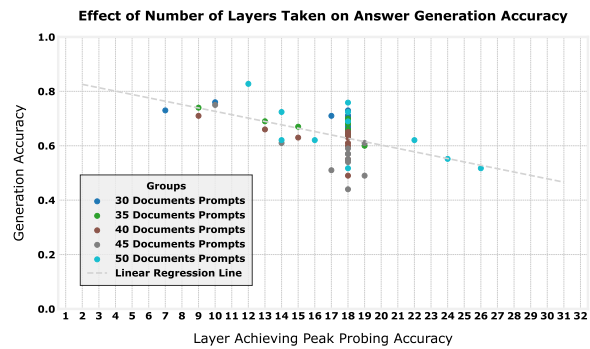

Figure 4: The LLM layer that achieves the peak probing accuracy (x-axis) vs. the accuracy of LLM in generating the correct answer (y-axis). We observe that a *later* peak correlates with *lower* accuracy in the language model's final output. This implies that the earlier an LLM encodes information from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

information.

We conduct additional multi-document probing tasks with 35, 40, 45, and 50 documents. In Fig. 4, for all IDs with probing accuracy exceeding 60% (to minimize the impact of outliers), the x-axis represents the layer where peak probing accuracy is achieved, while the y-axis displays the LLM's generation accuracy (without involving probes).

Early-layer information localization leads to higher generation accuracy. As Fig. 4 shows, there is a negative correlation between the layer with peak accuracy of locating target information and its final output accuracy. A two-sided t-test was employed to confirm the statistical significance of this observation (p-value < 5e-5; null hypothesis: zero slope). 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 demonstrates that while LLMs effectively capture context internally, they often fail to generate the correct answer due to positional bias. Through probing experiments, we reveal that information from the middle of the context requires deeper layers for retrieval. Additionally, we found that the delayed extraction of this information leads to a noticeable drop in accuracy. We hope these findings shed more light on the underlying cause behind positional bias and point to new areas for improving long-range information processing in LLMs.

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 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.

Acknowledgements

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*.

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. *arXiv 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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva

Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao,

Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. 2024. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp. *arXiv preprint arXiv:2407.00402*.

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.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. 2024. Do llms dream of elephants (when told not to)? latent concept association and associative memory in transformers. *arXiv preprint arXiv:2406.18400*.

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.

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*.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

nostalgebraist. 2021. logit lens on non-gpt2 models + extensions.

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.

Jonathon Shlens. 2014. A tutorial on principal component analysis. *Preprint*, arXiv:1404.1100.

CH-Wang Sky, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4401–4420.

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, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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. 2024b. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems* (NeurIPS).

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*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning* (ICML), pages 12697–12706.

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 $\{1, 0.1n, 0.2n, \dots, 1.0n\}$-th position as gold ID, where n is the number of documents ($0.1n$ for $n = 30$ means at ID 3). 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 `Gemma-7b-it` (Team et al., 2024b)

Following §4.1 and §4.2, we conduct same experiment procedure on one additional model, which produces the same pattern. The experiment is running on one A100 GPU.
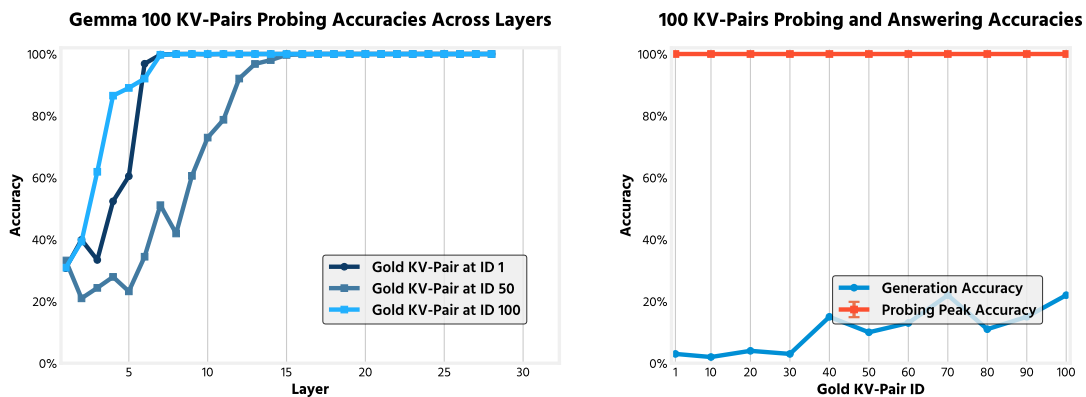

Figure 5: Replicating the results of Fig.2 and Fig.3 using the Gemma model with 100 kv-pairs. The findings for this model also align with the observations in §4.1 and §4.2. On the right, there is a notable gap between generation accuracy and peak probing accuracy, mirroring the results observed with Mistral in the main text.
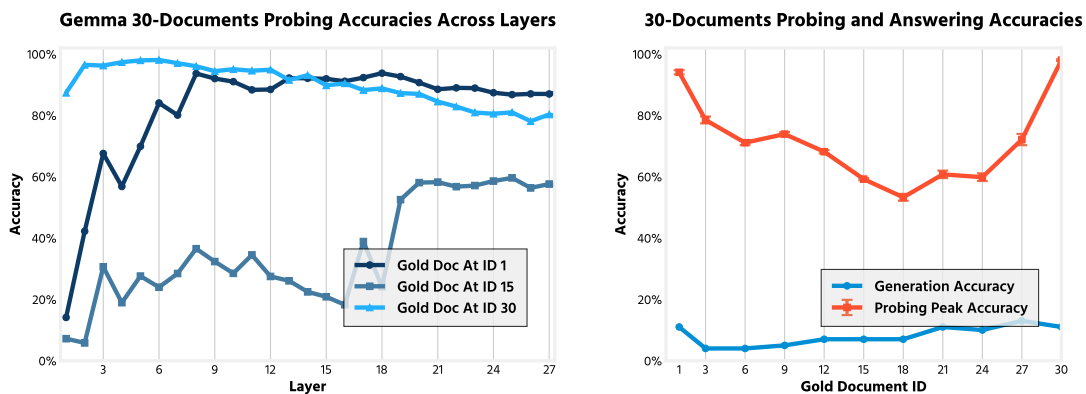

Figure 6: Replicating the results of Fig.2 and Fig.3 using the Gemma model with 30-document MDQA. The findings for this model also align with the observations in §4.1 and §4.2. On the left, while the beginning and end contexts follow the same pattern, the middle context exhibits a sudden drop in accuracy, indicating a brief loss of information, which is quickly regained.

D   2D Visualization of Hidden States Per Layer

We illustrate the model's internal representations by employing Principal Component Analysis (PCA) (Shlens, 2014) (Fig. 7) and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2020) (Fig. 8) to reduce the hidden states to two dimensions. Each subplot depicts the hidden states of a *single layer* within the Transformer. The subplots feature 11 colored points, each representing a *different gold document ID*. The color-to-ID mapping can be found in the color bar located in the upper right corner. We reuse the 30-documents prompts. Among this same set of documents, we rotate the ID of the gold document to 11 equally separated positions, as in §B. For the 11 prompts, we extract the last token dimensions from LLaMa3-8B-Instruct and perform dimensional reduction. Each dots in the plot represent a single data point.
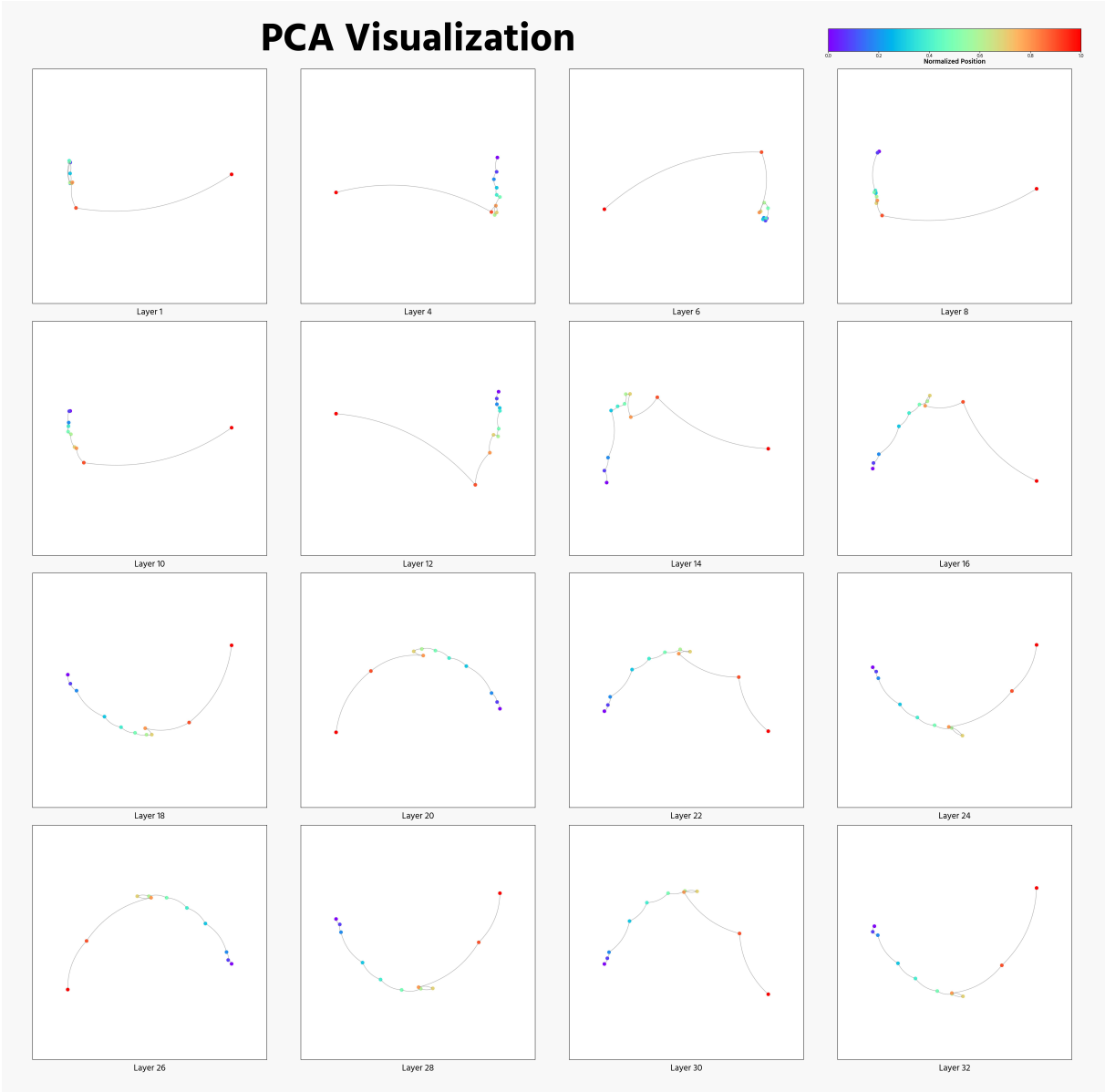

Figure 7: PCA visualization across layers: We apply PCA dimension reduction to the last embedding of each layer and visualize the results. The color of the dots indicates the position of the gold document in the prompt, ranging from 0% (gold document at the beginning) to 100% (gold document at the end). We observe that the dots are entangled in the early layers, start to form a path in the order of gold document ID in the middle layers, and this order diminishes in the later layers. This observation further supports our findings in §4.2, indicating that the internal representation of LLMs becomes richer with more layers, although the final layer may not exhibit the peak of representation richness.
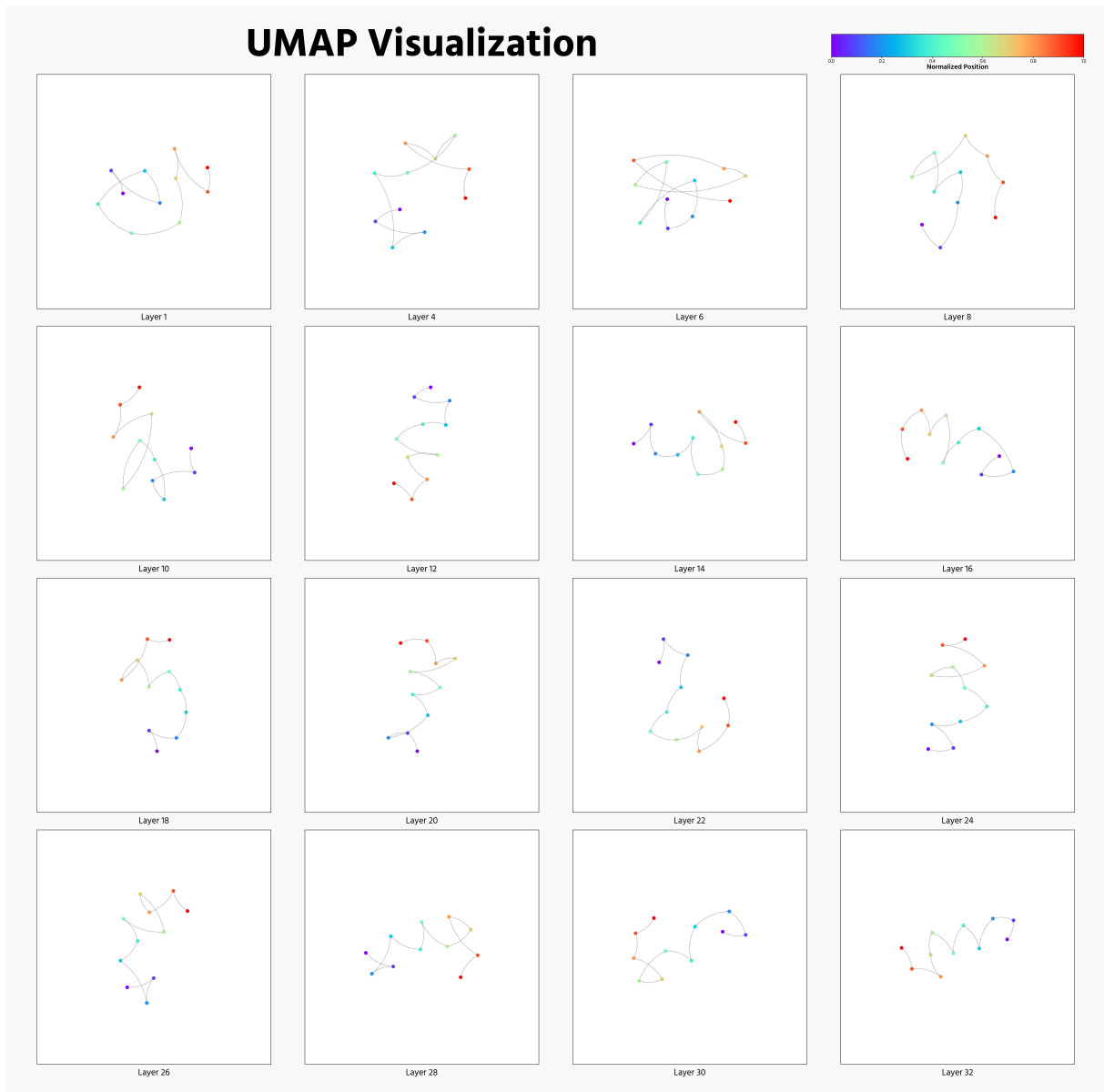
Figure 8: UMAP visualization across layers reveals a pattern similar to that in PCA Fig. 7. The embeddings become less entangled in the middle layers and appear more random in the extreme (earlier and final) layers.

Below, we calculate the average distance between points in the PCA visualization (Fig.7). For each layer l (each subfigure), there are 11 points $x_1^l, x_2^l, x_3^l, \ldots, x_{11}^l$, each corresponding to a different gold ID. We compute the average distance for layer l as $d_l = \sum_{i=1}^{10} \text{distance}(x_i^l, x_{i+1}^l)/10$. As shown in Fig.9, the average distance d_l increases up to layer 21, indicating a growing degree of separability. After reaching the peak distance, it decreases until the final layer. This pattern further supports our probing results. Additionally, as observed in §4.3, most gold IDs achieve peak accuracy around layer 18, which is very close to the peak distance layer (layer 20) identified here.
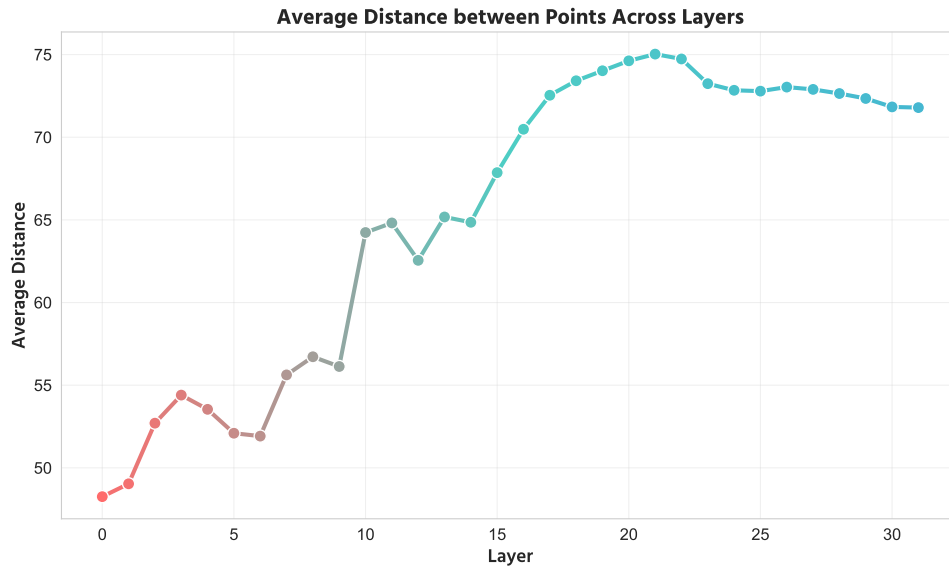

Figure 9: Average distance between scatter points across layers. The average distance has an obvious increasing trend during early layers, after achieves peak value around layer 21, it gradually decreases. The overall trend is similar to the probing accuracy in Fig. 3.

E   Language Generation Logits Trends Across Layers

We apply the *logit lens* (nostalgebraist, 2021) to examine the layers where the model begins to produce more probability mass over the correct answer. Specifically, for each layer, we multiply the last token embedding x^l with the LM Head and apply SoftMax to obtain the generation distribution over the entire vocabulary. We then record the probability of generating the correct token. The logit lens enables internal analysis of the LLM output distribution flow across layers. We reuse the 30-document prompts and apply the logit lens to each. Fig. 10 shows how the logits for the correct answer evolve across layers, with visualizations for prompts containing the gold document at different locations.
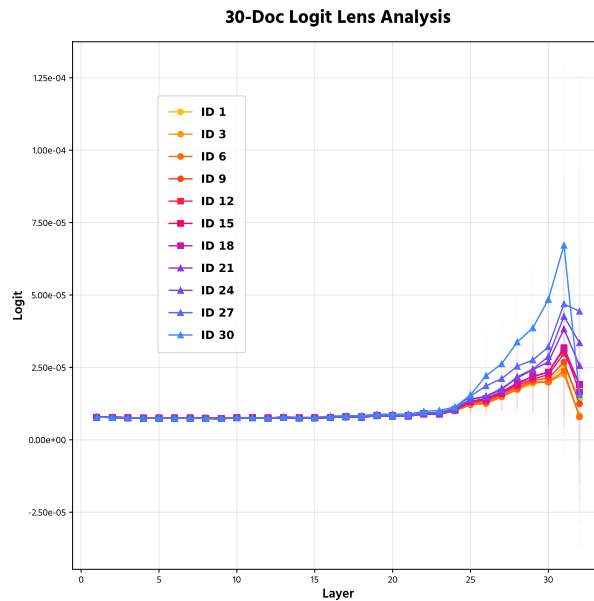

Figure 10: Logits of generating the first token of the correct answer across layers. The lines indicate different gold document ID. x-axis is the layer index, and y-axis is the logits across 1000 different prompts. During middle layers, blue line(ID 30) rises much more significantly than other IDs, however it also drops as the lowest at the last layer. For other lines, they share similar patterns in most layers but diverge in last few layers.

Layer-wise logit divergence.   The changes in logits across layers reveal certain key observations:

1. In all prompts, the logits follow an almost identical pattern before layer 20, showing a steady, minimal increase. The divergence begins at layer 20.

2. At layer 20, when the gold document is placed at the end of the prompt, there is a sharp and significant increase in the logits. This is supported by Fig. 3, where the same prompt achieves near-perfect probing accuracy.

3. All prompts reach their peak logit at layer 31, but exhibit a positional bias. The later the gold document appears in the prompt, the higher the peak logit.

These findings reveal that positional bias emerges during the model's internal processing, with a noticeable shift at a certain layer, where the position of gold information influences the quality of language output.