Functional Subspace, where language models can use vector algebra to solve problems

Jung H. Lee [1] **Sujith Vijayan** [2]

Abstract

Large language models (LLMs) were invented for natural language tasks such as translation, but they have proved that they can perform highly complex functions across domains. Additionally, they have been thought to develop new skills without being trained on them. These learning capabilities lead to LLMs adoption in a wide range of domains. Thus, it is imperative that we understand their operating mechanisms and limitations for proper diagnostics and repair. The earlier studies proposed that high level concepts are encoded as linear directions in LLMs activation space and that the geometry of embeddings have semantic meanings. Inspired by these studies, we hypothesize that LLMs may use subspaces and vector algebra in subspaces to perform tasks. To address this hypothesis, we analyze LLMs' functional modules and residual streams collected from LLMs engaging in in-context learning (ICL), one of the emergent abilities. Our analyses suggest that 1) LLMs can create subspaces, where evidence can be accumulated and 2) ICL tasks can be solved via simple algebraic operations in subspaces.

1. Introduction

The explosive growth in LLMs applications indicates that LLMs are highly capable learners (Naveed et al., 2025; Laskar et al., 2024; Minaee et al., 2025). After being pretrained to predict the next tokens, LLMs can be fine-tuned for complex tasks. Notably, a line of studies suggests that fine-tuning may not be strictly necessary, as LLMs can obtain "emergent" abilities (Matarazzo & Torlone, 2025; Lu et al., 2024; Brown et al., 2020; Wei et al., 2022). Pretraining does not aim to train LLMs to write code, but still, after pretraining, they can generate code. However, it remains poorly understood how LLMs can generalize and develop emergent abilities, making the diagnosis of LLMs' operations almost impossible. Before LLMs are being fully utilized in safety-critical domains, we need a better understanding of the principles behind LLMs' generalization and emergent abilities to diagnose and repair their operations.

To this end, we investigate how LLMs support in-context learning (ICL), one of LLMs' emergent abilities (Brown et al., 2020; Wei et al., 2022). ICL allows users to reconfigure LLMs to perform specific tasks without retraining or fine-tuning. For instance, a pretrained LLM can function as a thesaurus to an input prompt query, if the input prompt provides a pair of synonyms. ICL tasks are often simpler than general tasks such as coding, but LLMs still need to 1) extract task information from input prompts and 2) apply the information to answer users' queries. These two basic functions are essential for LLMs to interpret users' intentions and generate correct responses, which means that investigating ICL may lead us to a better understanding of how LLMs perform more general tasks.

Our study is inspired by two lines of studies. First, linear representation hypothesis posits that high level concepts are encoded as linear directions in LLMs' activation space (Park et al., 2023; Jiang et al., 2024). According to this hypothesis, tokens (i.e., sub-words) associated with 'happiness' are mapped onto vectors pointing to a 'happy' direction, and those associated with 'unhappiness' are mapped onto vectors pointing to a 'sad' direction. Second, the geometry of word embeddings may be linked to semantic meanings. For example, in the embedding space, the word 'king' can be converted to the word 'queen' if it is subtracted by 'man' and added by 'woman' (Mikolov et al., 2013b;a).

Based on these studies, we hypothesize that LLMs can support ICL by projecting input tokens into a subspace, where a desired task can be solved by using vector algebra. To address this hypothesis, we use principal component analysis (PCA) and linear regression analysis to determine the existence of a LLM subspace, where the answer can be inferred by vector algebra. Our empirical evaluations suggest that ICL tasks can be considered vector algebra problems in some of the LLMs' subspaces.

[1]Pacific Northwest National Laboratory, Seattle, WA, USA [2]Virginia Tech, Blacksburg, VA, USA. Correspondence to: Jung H. Lee <jung.lee@pnnl.gov>.

Accepted to 1st Workshop on Combining Theory and Benchmarks, CTB@ICML2026, Seoul, South Korea. Copyright 2026 by the author(s).

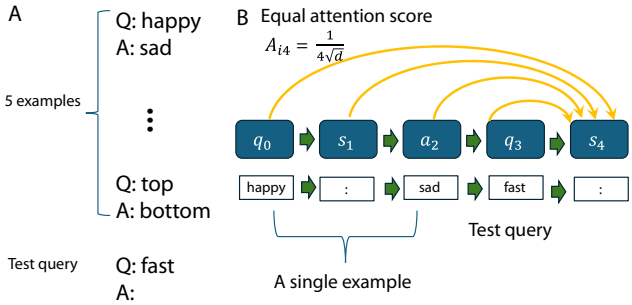

Figure 1. ICL prompts. (A), The structure of the used ICL prompts. (B), The structure of a single example ICL with uniform attention score A_{ij}. This panel illustrates how symbols q_i, s_i, a_i are mapped onto ICL tokens.

2. Subspaces naturally generated by LLMs

In this section, we discuss how transformers' functional modules generate subspaces and use them to support ICL. Section 2.1 briefly discusses ICL tasks used in this study. In Sections 2.2, 2.3, 2.4, we show how LLMs' functional modules generate subspaces, how evidence can be accumulated and how ICL tasks are translated to vector algebra problems, respectively.

2.1. ICL tasks

Brown et al. (2020) found that LLMs can infer correct answers, if they are exposed to a few examples demonstrating a desired task; see (Wei et al., 2022) as well. When a few pairs of synonyms are included in the input prompts, LLMs can return a synonym to the last query word. Thus, ICL prompts contain multiple examples demonstrating desired tasks and a query. In our study, we use the dataset generator released by an earlier study (Todd et al., 2024) to create ICL prompts for 4 different tasks, antonyms, synonyms, countries-their capital cities (country-capital), and English-French words. All 4 tasks can be demonstrated with two words separated by a separator. Throughout this study, we refer to them as queries, separators and answers. For all tasks, we provide 5 example pairs of queries and answers and a single test query (without a corresponding answer) and direct LLMs to find its counterpart using example pairs. Each query and answer are marked by the separators, 'Q:' and 'A:' respectively (Fig. 1A).

2.2. Transformer blocks and subspaces

LLMs first convert words in input prompts into tokens (i.e., sub-words) and use a sequence of transformer layers to progressively process them, each of which consists of self-attention (SA) layer and Feed-Forward Network (FFN) (Vaswani et al., 2023). A transformer layer l receives inputs h^{l-1} from its previous layer and generates h^l using its own SA and FFN. As the transfer layer uses residual connections, h^l can be recursively expressed as $h^l = h^{l-1} + a^l + m^l(a^l + h^{l-1})$ (Meng et al., 2023), where a^l and m^l denote the output of SA layer and FFN in transformer layer l, respectively.

2.2.1. Feed Forward Networks as Associative Memory

How do LLMs support ICL? We made two notes on FFNs' potential roles. First, the two consecutive tokens in the language prompts are semantically related. That is, for a given token, there are a finite number of possible choices for the next token. Second, the earlier studies (Geva et al., 2021; Meng et al., 2023; 2022) suggested that FFNs work as associative memory, storing pairs of keys and values. That is, FFNs can retrieve a preconfigured value for a given key, as associative memory does. These observations can lead us to assume that FFNs may learn possible answers for a given input. Importantly, we also note that a single word (i.e., entity) can be linked to a number of different words depending on the context. The city of London is the capital city of the United Kingdom but is also well known as the global financial hub. That is, instead of a single associated value, FFNs should memorize multiple associated values for a single key to succeed in supporting language tasks.

Notably, FFNs in LLMs consist of two synaptic layers, ($\boldsymbol{w}^{1st}$ and $\boldsymbol{w}^{2nd}$) and one hidden layer of memory cells m_i (Eq. 1).

$$\begin{aligned} m_i &= \Sigma_j \boldsymbol{w}_{ij}^{1st} x_j, \\ O_k &= \Sigma_i \boldsymbol{w}_{ki}^{2nd} g(m_i), \end{aligned} \tag{1}$$

where g is the activation function. If an input h_i activates a single m_i only, the output O_k is the second synaptic layer weight $\boldsymbol{w}_{ki}^{2nd}$ originating from m_i, scaled by $g(m_i)$. h_i, however, likely activate multiple m_is, and thus, the output is likely the weighted sum of $\boldsymbol{w}_{ki}^{2nd}$ (values of associative memory in (Meng et al., 2023)). That is, the outputs of FFNs exist in a space S^l spanned by $\boldsymbol{w}_{ki}^{2nd}$. Vectors in subspace S^l can be decomposed into a set of components, and we propose that these components can correspond to potential predictions (i.e, values associated with keys). Additionally, since LLMs use residual streams, they can store potential predictions of all individual layers.

2.2.2. Self-attention selector of subspaces

Even if LLMs can encode multiple predictions for a given input using subspaces in residual streams, they still need to choose the right one to make the proper response depending on the context defined by the neighboring words. In LLMs, SA layers mediate interactions between tokens (i.e., words), making it natural to assume that SA layers can generate the context and allow transformers to choose the correct prediction aligned with the context. SA layers evaluate

attention score A^l_{ij}, which determines the strength of the influence of token i onto token j (Eq, 2).

$$A^l_{ij} = (\boldsymbol{K}\vec{h}^{l-1}_i)^T \cdot \boldsymbol{Q}\vec{h}^{l-1}_j \tag{2}$$

Then, SA layer's outputs a^l_{ij} on token j are modulated by value matrix $\boldsymbol{V}$.

$$a^l_{ij} = \frac{1}{\sqrt{d}} softmax(A^l_{ij})\boldsymbol{V}h^{l-1}_i \tag{3}$$

, where $softmax$ is estimated over i to normalize the influence from all tokens i, and d is the model dimension, which indicates the size of residual streams[1].

2.3. Accumulating evidence in subspaces

It has been observed that the accuracy of LLMs and the number of examples are positively correlated, which suggests that LLMs may effectively accumulate the evidence presented in ICL prompts. To gain insights into the mechanism underlying this evidence accumulation and the potential roles of subspaces, we consider a simple scenario, where LLMs can access a single in-context example (Fig. 1 B). For simplicity, we consider 5 tokens (T_0, T_1, T_2, T_3 , T_4), which correspond to the first query (q_0), the first separator (s_1), answer (a_2), the test query (q_3) and the last separator (s_4).

For layer l and the last place (i.e., the place of the last separator), the input of the FFN can be summarized as Eq. 4.

$$x^l_{j\equiv 4} = h^{l-1}_j + \Sigma^{i=4}_{i=0} a^l_{ij\equiv 4} \tag{4}$$

, where a^l_{ij} denotes attention score and h^l denotes the residual stream in layer l. As our goal is to gain insights into how LLMs accumulate evidence, we further simplify attention score a^l_{ij} by assuming that all 4 preceding tokens (T_0, T_1, T_2, T_3) are equally important [2]. In this ideal case, $a^l_{i4} = \alpha \boldsymbol{V} h^{l-1}_i$, where $\alpha = 1/(4\sqrt{d})$. Together with the fact that h^{l-1}_i can be decomposed into potential answers An^{l-1}_is , we obtain the following (Eq. 5):

$$\begin{aligned} x^l_{j\equiv 4} =& h^{l-1}_j + \alpha \boldsymbol{V}\{\Sigma_k \beta_{0k} An^0_k \\ &+ \Sigma_k \beta_{1k} An^1_k + \Sigma_k \beta_{2k} An^2_k \\ &+ \Sigma_k \beta_{3k} An^3_k\} \end{aligned} \tag{5}$$

, where An^i_k encode the possible answers of FFNs for a given input token x_i; where k runs for all possible answers, and β_{ik} is a constant. As the pretraining aims to train LLMs (and FFNs as well) to predict the next tokens, the possible number of An^i_k is finite, as the two consecutive tokens are semantically related with each other. We note that the first query (q_0) and separator (s_1) can predict the answer (a_2) due to the semantic relationships between query and answer tokens. Thus, one can expect some of An^0_k, An^1_k and An^2_k to be overlapped, and their intersections $\{\tilde{An}_k\}$ can make the first query and separators (q_0 and s_1) predict the answer a_2. With intersections $\{\tilde{An}_k\}$, we can reorganize $x^l_{j=4}$ as follows (Eq. 6):

$$\begin{aligned} x^l_{j\equiv 4} =& h^{l-1}_j + \\ &+ \alpha \boldsymbol{V} \Sigma_m \tilde{An}_m \{\Sigma_k \beta_{0k} + \beta_{1k} + \beta_{2k}\} \\ &+ \alpha \boldsymbol{V}\{\Sigma_k \beta_{0k} An^0_k \\ &+ \Sigma_k \beta_{1k} An^1_k + \Sigma_k \beta_{2k} An^2_k \\ &+ \Sigma_k \beta_{3k} An^3_k\} \end{aligned} \tag{6}$$

In Eq. 6, one can see that the evidence can be effectively accumulated along $\{\tilde{An}_k\}$. Eq. 6 may be valid only under an ideal condition, but we still can speculate that $\{\tilde{An}_k\}$ intersections of possible answers also exist in general cases, in which multiple examples have non-homogeneous influence on the last token[3] and allow LLMs to accumulate evidence in examples. Further, we argue that $\{\tilde{An}_k\}$ could be the bases for subspaces, where the evidence could be effectively accumulated.

To address this possibility, we use PCA to obtain a subset of potential bases for LLMs' activation space, which can model $\{\tilde{An}_k\}$.

2.4. ICL as a vector operation

ICL tasks are usually simple and may be readily realized by vector operation. Let's consider a case, in which ICL examples are pairs of antonyms, and thus LLMs need to convert the last query into its antonym. If LLMs can map all examples in the prompt onto vectors in a subspace, where a semantic meaning is represented by a unique direction and the antonyms are mapped onto vectors pointing in opposite directions, LLMs can find the antonym of the last query by simply reversing the direction of the query word.

If LLMs use vector algebra to solve ICL tasks, an answer token ($\vec{a}$) can be described as a linear function of a query token ($\vec{q}$) and a separator token ($\vec{s}$) (Eq. 7).

$$\vec{a} = \alpha' \vec{q} + \beta' \vec{s} \tag{7}$$

[1] For brevity, we ignore the output matrix $\boldsymbol{O}$ here

[2] As T_4 is a separator, which does not mediate semantic meanings, we assume that it does not have meaningful self-influence on T_4.

[3] Keep in mind that we are assuming a single example and that a query, a separator and an answer in this example have homogeneous relevance above.

Based on this line of reasoning, we conduct linear regression analysis along 30 principal components using scikit-learn (Pedregosa et al., 2011). If $\vec{a}$ is well described by linear function of $\vec{q}$ and $\vec{s}$, the quality of linear regression would be high. Otherwise, its quality would be low. Below, we examine R^2 to evaluate the quality of linear regression along 30 principal components (i.e., the potential bases for LLMs' subspaces).

3. Empirical evaluation

In this study, we probe 6 pretrained LLMs publicly available: GPT-j-6B (Wang & Komatsuzaki, 2021), Meta-Llama-3.1-8B(Grattafiori et al., 2024), OLMo-2-0325-32B(OLMo et al., 2024), Phythia-12B(Biderman et al., 2023), gemma-3-27b-it (Team et al., 2025) and GPT-NEOX-20B(Black et al., 2022). All models are instantiated and tested with the publicly available machine learning libraries, Pytorch (Paszke et al., 2017) and Transformers (Wolf et al., 2020).

3.1. Principal components as potential bases for subspaces

Since residual streams of LLMs store all hidden representations, we analyze them to identify subspaces (i.e., components). Specifically, we created 200 ICL prompts for 4 different tasks (Todd et al., 2024), antonyms, synonyms, countries-their capital cities (country-capital), English-French words. Each prompt contains 5 examples and 1 question, and queries and answers are separated by separators (':'); see Fig. 1A. In the experiment, we collect residual streams of all tokens from all layers, while LLMs process ICL prompts. After collecting residual streams, we convert them into $2D$ arrays. Each row of the arrays is a residual stream, and thus the size of the row vectors is the same as the model dimension d. For instance, the model dimension d of Meta-LLama-3.1-8B is 4096. The number of rows is the same as the total number of tokens. As some words are tokenized into multiple tokens, the exact number of tokens vary from one model to another. As the residual streams are not bounded, each row is normalized to have a unit norm. Additionally, all columns are centered, as it is required for PCA.

Next, 30 principal components are evaluated using scikit-learn, an open-source machine learning library(Pedregosa et al., 2011). Fig. 2 shows the accumulated explained variance for 30 principal components. As shown in the figure, $40 - 90\%$ of variability is accounted for by just 30 principal components, although $d \approx 5000$, suggesting that LLMs can be mapped onto low-dimensional subspaces.

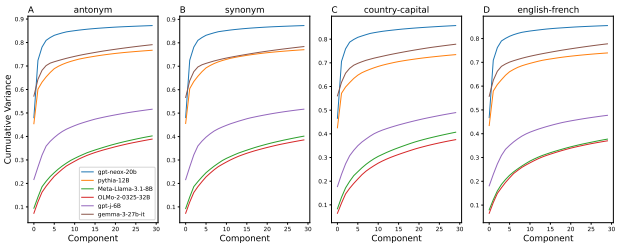

Figure 2. The explained variance accounted for by 30 principal components. x-axis denotes principal components, and y-axis denotes the cumulative explained variance. (A), The explained variance evaluated from LLMs engaging the antonym task. In the panel, 6 models are clarified using different colors (see the inset). (B)-(D), the same as (A), but the task is synonym, country-capital and English-French, respectively.

3.2. ICL Tasks translated as vector algebra problems in subspaces

Although 30 principal components can explain the majority of residual streams' variances, it cannot guarantee that LLMs would use them to support ICL. Thus, to further test whether LLMs actively utilize subspaces, we ask if the answer token $\vec{a}$ can be a linear combination of query $\vec{q}$ and separator $\vec{s}$ tokens. When the words are tokenized into multiple tokens, only the last tokens of query and answer words are used in this analysis, which is consistent with the earlier studies (Todd et al., 2024; Yang et al., 2024; Meng et al., 2023).

Specifically, for a transformer layer l and a principal component PC_i, we project answers, queries and separators in 1000 examples (5 in-context examples in 200 prompts) to obtain $\vec{a}_i^l$, $\vec{q}_i^l$, $\vec{s}_i^l$ and regress $\vec{a}_i^l$ into $\vec{q}_i^l$, $\vec{s}_i^l$ (Eq. 8).

$$\vec{a}_i^l = \alpha \vec{q}_i^l + \beta \vec{s}_i^l, \tag{8}$$

where $\vec{a}_i^l$, $\vec{q}_i^l$, $\vec{s}_i^l$ denote 1-d vectors whose length is 1000 (5 in-context examples in 200 prompts).

To evaluate the quality of regression and identify a subspace, where the answer token can be described by the linear combination of query and separator tokens, we use R^2. When the regression is perfect, $R^2 = 1$. By contrast, R^2 approaches $-\infty$, when the regression fails. That is, if R^2 is close to 1, the principal component can be a basis for a subspace that we aim to identify. Fig. 3 shows R^2 collected from all layers of 6 models engaging in the 'antonym' task. For clarity, we clip R^2 smaller than -1 to -1. Across all 6 models, R^2 is high along with a few components. That is, answer tokens are well approximated by linear combinations of query and separator tokens (Eq. 8) in a subspace spanned by these components, supporting that LLMs do use subspaces and that ICL tasks can be solved via simple vector operations.

We made two more observations. First, R^2 is low in early layers and becomes higher in late layers, raising the possibil-

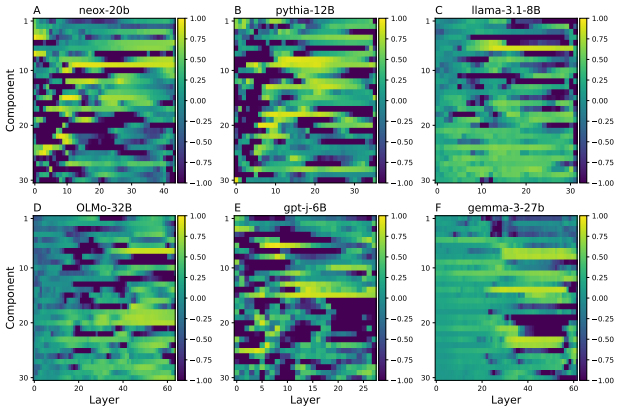

Figure 3. Quality of linear regression evaluated. R^2 is evaluated using LLMs engaging in 'antonym' task. x-axis denotes the transformer layer, and y-axis denotes the principal components. (A)-(F) show R^2 estimated from GPT-j-6B, Meta-Llama-3.1-8B, OLMo-2-0325-32B, Phythia-12B, gemma-3-27b-it and GPT-NEOX-20B. For clarity, R^2 smaller than -1 is clipped to -1.

ity that the early layers of LLMs gradually transform LLM's activation (sub)-spaces into subspaces, where ICL can be solved by simple vector operations. Second, the principal components with high R^2 appear to be largely consistent across the layers. For instance, R^2 is consistently high along component 9 of GPT-NEOX-20B (Fig. 3A).

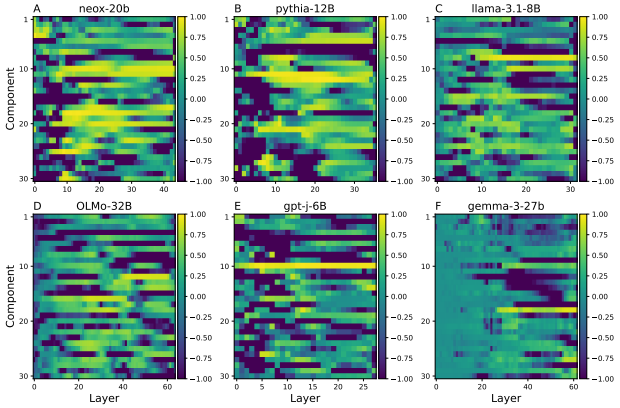

Figure 4. The same as Fig. 3, but the task is country-capital.

These observations suggest that the components with high R^2 encode the concept 'antonym', which is consistent with linear representation hypothesis (Park et al., 2023) but further suggests that 1) there are multiple directions associated with concepts and 2) actual computations associated with concepts occur according to these directions. We repeat the same analysis with 3 more tasks and observe the equivalent results with different tasks, 'synonym' (Fig. 4), 'country-capital' (Fig. 5) and 'English-French' (Fig. 6).

If LLMs do utilize subspaces and vector operations in it, one can expect that tokens are clustered together depending

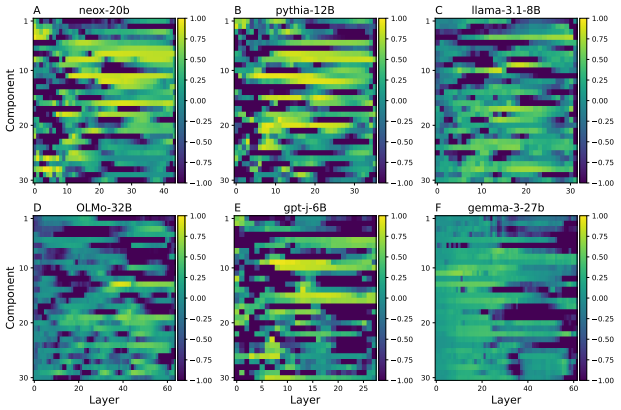

Figure 5. The same as Fig. 3, but the task is English-French.

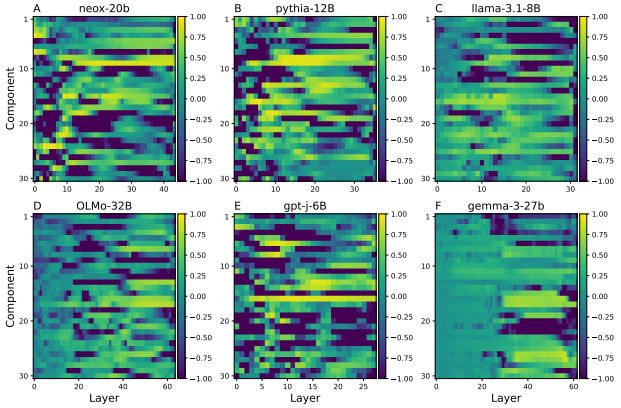

Figure 6. The same as Fig. 3, but the task is synonym.

on types. We test this possibility by visualizing answer, separator and query tokens in $3D$ subspace spanned by the top 3 principal components according to R^2. In this analysis, we choose a single layer with the highest R^2 from individual models engaging in 4 tasks (Figs. 7, 8, 9 and 10). As shown in the figures, answer, separator and query tokens are well clustered with the same token type, and the token types are separated from other types. These results suggest that tokens encode task specific information (i.e., query, separator or answer), confirming that the subspace is associated with LLMs' decision making.

4. Discussion

Can LLMs use subspaces to solve ICL tasks? We analyze LLMs' functional modules under an ideal condition, and our analysis suggests that LLMs architecture may natively create subspaces and use them to accumulate evidence. Our empirical evaluation further shows that ICL tasks can be solved by vector algebra in low dimensional subspace(s) of LLMs. ICL tasks may be simpler than general tasks given to LLMs, but ICL still requires LLMs to extract task infor-

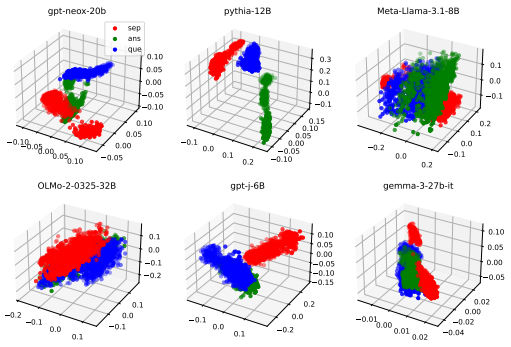

Figure 7. Query (que), separator (sep) and answer (ans) tokens in the subspace spanned by 3 principal components with the highest R^2s. Red, green and blue dots represent separator, answer and query tokens, respectively (see the inset). All tokens are collected from 6 LLMs engaging in the antonym task. The model is specified by the name above each plot.

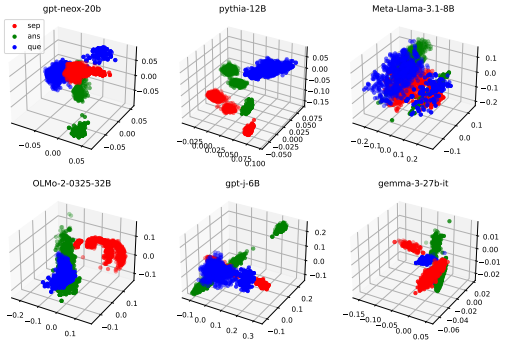

Figure 8. The same as Fig. 7, but the task is country-capital.

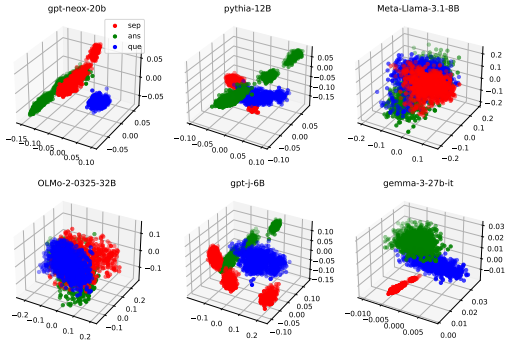

Figure 9. The same as Fig. 7, but the task is English-French.

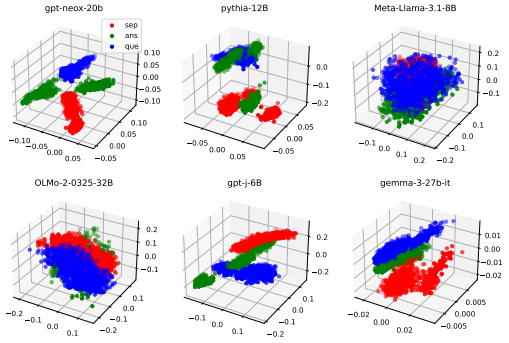

Figure 10. The same as Fig. 7, but the task is synonym.

mation and apply them to solve unseen questions, which is essential for LLMs to succeed in more general tasks. This means it may be possible for LLMs to use subspaces to perform general tasks.

If LLMs rely on subspaces for their operations and decision-making, identifying subspaces associated with desired tasks and probing interactions among them may provide us with new ways to evaluate, diagnose and intervene in LLMs' operations. In our study, we choose ICL, since ICL prompts are supposed to mediate the same contextual information, and thus, a single subspace aligned with it would likely be dominant. As a result, identifying the subspaces becomes relatively straightforward.

However, when LLMs engage in general tasks, multiple subspaces would interact with one another in complex ways, which makes probing subspaces associated with LLMs' operations rather difficult. An effective way to overcome this challenge would be to develop a new set of benchmarks that could probe LLMs' subspaces. To this end, it is necessary to build datasets that provide explicit and extensive contextual information, which can be used to identify 'functional' subspaces. We believe that such benchmarks will advance our understanding of LLMs' operating principles including generalization and emergent abilities.

References

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U. S., Purohit, S., Reynolds, L.,

Tow, J., Wang, B., and Weinbach, S. Gpt-neox-20b: An open-source autoregressive language model, 2022. URL https://arxiv.org/abs/2204.06745.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg,

L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jiang, Y., Rajendran, G., Ravikumar, P. K., Aragam, B., and Veitch, V. On the origins of linear representations in large language models. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21879–21911. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jiang24d.html.

Laskar, M. T. R., Alqahtani, S., Bari, M. S., Rahman, M., Khan, M. A. M., Khan, H., Jahan, I., Bhuiyan, A., Tan, C. W., Parvez, M. R., Hoque, E., Joty, S., and Huang, J. X. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13785–13816, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.764. URL https://aclanthology.org/2024.emnlp-main.764/.

Lu, S., Bigoulaeva, I., Sachdeva, R., Tayyar Madabushi, H., and Gurevych, I. Are emergent abilities in large language models just in-context learning? In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5098–5139, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.279. URL https://aclanthology.org/2024.acl-long.279/.

Matarazzo, A. and Torlone, R. A survey on large language models with some insights on their capabilities and limitations, 2025. URL https://arxiv.org/abs/2501.04040.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35, 2022.

Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL http://arxiv.org/abs/1301.3781.

Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In Vanderwende, L., Daumé III, H., and Kirchhoff, K. (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090/.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. Large language models: A survey, 2025. URL https://arxiv.org/abs/2402.06196.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. A comprehensive overview of large language models. *ACM*

Trans. Intell. Syst. Technol., 16(5), August 2025. ISSN 2157-6904. doi: 10.1145/3744746. URL https://doi.org/10.1145/3744746.

OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjonsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 OLMo 2 Furious, 2024. URL https://arxiv.org/abs/2501.00656.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

Paszke, A., Gross, S., Chintala, S., Chanan, Gregory Yang, E., DeVito, Z., Lin, Zeming Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A. M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A. S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C. L., Choquette-Choo, C. A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D. S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H. T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P. K., Culliton, P., Schmid, P., Sessa, P. G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S. R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L. G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., and Hussenot, L. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Todd, E., Li, M., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

Wang, B. and Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers:

State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Yang, S., Gribovskaya, E., Kassner, N., Geva, M., and Riedel, S. Do large language models latently perform multi-hop reasoning? In *Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.550.