Analyzing Key Neurons in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) possess vast amounts of knowledge within their parameters, prompting research into methods for locating and editing this knowledge. Previous investigations have primarily focused on fill-in-the-blank tasks and locating entity-related (*usually single-token facts*) information in relatively small-scale language models. However, several key questions remain unanswered: (1) *How can we effectively locate query-relevant neurons in contemporary autoregressive LLMs, such as LLaMA and Mistral?* (2) *How can we address the challenge of long-form text generation?* (3) *Are there localized knowledge regions in LLMs?* In this study, we introduce Neuron Attribution-Inverse Cluster Attribution (NA-ICA), a novel architecture-agnostic framework capable of identifying key neurons in LLMs. NA-ICA allows for the examination of long-form answers beyond single tokens by employing the proxy task of multi-choice question answering. To evaluate the effectiveness of our detected key neurons, we construct two multi-choice QA datasets spanning diverse domains and languages. Empirical evaluations demonstrate that NA-ICA outperforms baseline methods significantly. Moreover, analysis of neuron distributions reveals the presence of visible localized regions, particularly within different domains. Finally, we demonstrate the potential applications of our detected key neurons in knowledge editing and neuron-based prediction.

1 Introduction

Large Language Models (LLMs) contain substantial amounts of knowledge within their parameters. Existing research endeavors to locate and edit this knowledge through gradient-based methods (Dai et al., 2022) or causality-based methods (Meng et al., 2022a). These methods typically employ fill-in-the-blank tasks, such as "`Paris is the capital of ___`", to ascertain the correlation between the query and neurons or layers in the Feed-forward Networks (FFNs) of BERT (Kenton and Toutanova, 2019) and GPT (Radford et al.). Another branch of pioneering research attempts to locate functional regions in small-size language models such as BERT and GPT-small, including linguistic regions (Zhang et al., 2024b), factual subnetworks (Ren and Zhu, 2022; Bayazit et al., 2023), and modular structures (Zhang et al., 2023; Conmy et al., 2023).

While these studies successfully analyze the internal reasoning behaviors of LLMs, three significant questions remain underexplored: (1) How can we effectively locate query-relevant neurons in contemporary autoregressive LLMs, such as LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023), given that their FFNs architectures differ from those of BERT and GPT? (2) How can we address the challenge of long-form text generation, as previous methods have been limited to single-token entity facts? (3) Are there localized knowledge regions in LLMs analogous to the localized functional regions observed in human brains (Brett et al., 2002)?

To address the first two questions, we introduce a novel framework named ***Neuron Attribution-Inverse Cluster Attribution (NA-ICA)*** designed to identify key neurons in LLMs. The principal advantages of NA-ICA are its architecture-agnostic nature and its capability of handling long-form text generation effectively. The overall structure of the framework is depicted in Figure 1. NA-ICA draws inspiration from the TF-IDF keyword extraction method (Salton, 1983), aiming to extract significant neurons for each input query. The process begins by transforming an open-ended generation task into a multiple-choice question-answering format. By employing prompt engineering, we constrain LLMs to generate only the option letter rather than the complete answer. This approach allows for the examination of long-form generation beyond sin-

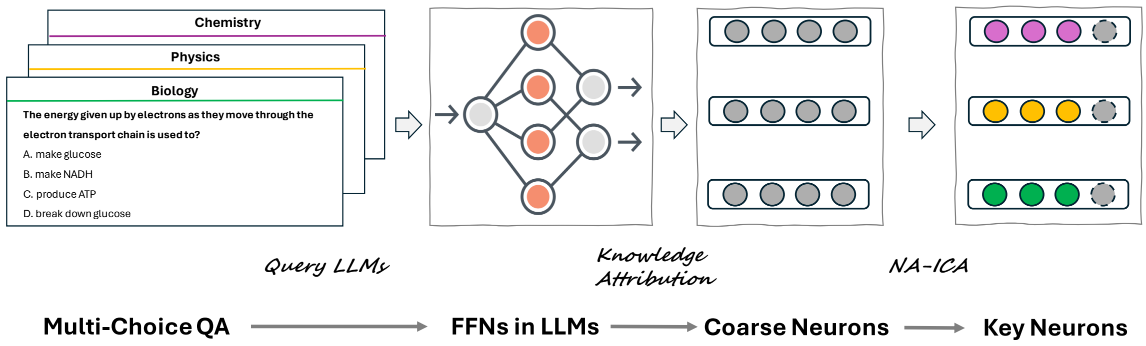


Figure 1: The overall framework of our Attribution-Inverse Cluster Attribution (NA-ICA) which aims to detect query-related key neurons. Neurons with solid lines mean key neurons while dashed ones mean common neurons that are shared across different queries.

gle tokens and extends previous methodologies to autoregressive LLMs. Subsequently, we adapt the Knowledge Attribution method (Dai et al., 2022) to compute *Neuron Attribution*, which elucidates the relationship between neurons and the input query. We then gather clusters for a series of queries and calculate the *Inverse Cluster Attribution*. This step mitigates the influence of neurons that recur across clusters (or queries). The final step involves multiplying the neuron attribution and inverse cluster attribution values to pinpoint key neurons. Additionally, we identify certain *Common Neurons* that are associated with common words, punctuation marks, and option letters. Excluding these common neurons enhances the detection of key neurons. Empirical evaluations demonstrate that our proposed method outperforms baseline approaches.

To investigate the existence of localised knowledge regions, we construct two multi-choice QA datasets encompassing various domains and languages. Then, we visualize the geographical locations of the detected key neurons in LLaMA. Our findings indicate that distinct localized regions emerge in the middle layers, particularly for domain-specific neurons. Language neurons are more sparse but show a certain degree of regionality. Additionally, we observed that common neurons are concentrated in the top layer, predominantly expressing frequently used tokens.

In summary, our main contributions are fourfold: (1) **A scalable method:** we propose NA-ICA to detect key neurons in LLMs; NA-ICA method is architecture-agnostic and can deal with long-form generations. (2) **Two new datasets:** we curate two multi-choice QA datasets that contain different types of knowledge, namely Domain Knowledge and Language knowledge. (3) **In-depth studies:** we are the first to show that there are visible localized regions in LLaMA. (4) **Potential applications:** we show that NA-ICA might be useful for knowledge editing and neuron-based prediction.

2 Related Work

2.1 Locating Knowledge in LLMs

LLMs contain extensive knowledge within their parameters, encompassing factual (Petroni et al., 2019; Zhou et al., 2020; Jiang et al., 2020; Roberts et al., 2020; Pezeshkpour, 2023), linguistic (Liu et al., 2019; Jawahar et al., 2019; Chen et al., 2023) and domain-specific knowledge (Sung et al., 2021; Frieder et al., 2024). Despite this, the mechanisms and locations of knowledge storage within these models remain unclear. Recent mechanistic studies suggest that knowledge is primarily stored in the FFNs (Feed-forward Networks) layers of Transformers (Geva et al., 2021, 2022). Ongoing research is focused on developing methods to precisely identify and locate this knowledge within the FFNs layers. Given an input, *gradient-based methods* (Ancona et al., 2019; Dai et al., 2022) quantify the sensitivity of model outputs to internal model components, identifying relevant neurons. However, these studies focus exclusively on traditional neural architectures and encoder-only models like BERT, leaving decoder-only models such as GPT and LLaMA underexplored. *Causality-based methods* employ causal mediation analysis to discern the particular layers associated with a given factual input (Meng et al., 2022a). Subsequent research adopts the locate-and-edit paradigm to refine the knowledge within LLMs (Meng et al., 2022b; Ju and Zhang, 2023; Zhang et al., 2024a).

While previous approaches have effectively identified specific information in LLMs, they commonly rely on the fill-in-the-blank cloze task to evaluate the factual capabilities of language models. For instance, they use a prompt query like "`Paris is the capital of` ___" to locate weights associated with the `France` entity. However, this methodology has limited applicability, as language models exhibit the capacity to generate long-form and open-ended responses to diverse queries. In contrast to prior methodologies, our approach leverages the proxy task of multiple-choice QA for knowledge localization. This alternative strategy renders the localization process architecture-agnostic and facilitates the handling of long-form content generation.

2.2 Analyzing Knowledge Distribution in LLMs

Given the human-like reasoning capabilities observed in LLMs across various tasks (Zhao et al., 2023), and since our brain contains functional locations associated with distinct cognitive processes (Brett et al., 2002; Bjaalie, 2002; Gholipour et al., 2007), we ask whether there are similar regions in LLMs. Previous investigations have explored the behaviors of individual neurons indicating that a neuron can encode multiple concepts (Bolukbasi et al., 2021) while a concept can also be distributed across multiple neurons (Dalvi et al., 2019; Durrani et al., 2020; Chen et al., 2024). Subsequent endeavors have sought to identify functional regions in LLMs, encompassing linguistic regions (Zhang et al., 2024b), factual subnetworks (Ren and Zhu, 2022; Bayazit et al., 2023), and modular structures (Zhang et al., 2023; Conmy et al., 2023). These studies have systematically investigated localized behaviors in smaller-scale language models, such as BERT and GPT-small. Building upon these foundations, our research embarks on the examination of knowledge locations in larger-size LLMs, specifically those with 7B parameters, spanning multiple knowledge domains.

3 Background

Feed-forward Networks in LLMs Feed-forward networks (FFNs) are widely used by transformer-based language models. Geva et al. (2021) reveal that FFNs emulate key-value memories and their outputs are responsible for refining the final output distribution over the vocabulary. Although traditional two-layer FFNs in BERT (Kenton and Toutanova, 2019) and GPT-2 (Radford et al.) have been studied well, the behaviors of FFNs in modern LLMs such as LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Gemma (Team et al., 2024) are not well-explored. These LLMs adopt Gated Linear Units (GLUs) (Dauphin et al., 2017) in their FFNs, which can be formulated as follows:

$$\mathrm{FFN}(\mathbf{X}) = (\mathbf{X}\mathbf{W}^U \odot \mathrm{SiLU}(\mathbf{X}\mathbf{W}^G))\,\mathbf{W}^D \quad (1)$$

Here, $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the input sequence, n is the number of tokens and d is the dimension of input vectors; $\mathbf{W}^U \in \mathbb{R}^{d\times m}$, $\mathbf{W}^G \in \mathbb{R}^{d\times m}$, $\mathbf{W}^D \in \mathbb{R}^{m\times d}$ are parameter matrices and $\odot$ is the Hadamard product; finally SiLU (Elfwing et al., 2018) is the activation function.

Knowledge Neurons Dai et al. (2022) propose a gradient-based *Knowledge Attribution* to identify the knowledge neurons in BERT by using the fill-in-the-blank cloze task. Their method evaluates the contribution of each neuron in FFNs to the knowledge predictions. Given a prompt q "`Paris is the capital of` ___", the probability of the correct answer predicted by a language model can be formulated as:

$$P_q(\hat{w}_i^l) = p(y^*|x, w_i^l = \hat{w}_i^l) \quad (2)$$

where y^* is the correct answer (`France`); w_i^l denotes the i-th intermediate neuron in the l-th layer in FFNs; $\hat{w}_i^l$ is a constant we assign to w_i^l.

In order to measure the attribution score (or contribution) of a neuron, they gradually change the w_i^l from 0 to its original value computed during the forward pass through the LLM and integrate the gradients (Sundararajan et al., 2017):

$$\mathrm{Attr}(w_i^l) = \bar{w}_i^l \int_{\alpha=0}^{1} \frac{\partial P_q(\alpha \bar{w}_i^l)}{\partial w_i^l} \mathrm{d}\alpha \quad (3)$$

where $\frac{\partial P_q(\alpha \bar{w}_i^l)}{\partial w_i^l}$ is the gradient with regard to w_i^l. $\mathrm{Attr}(\cdot)$ accumulates the output probability change as α gradually varies from 0 to 1. The attribution measures the contribution of the neuron w_i^l to the correct answer. In practice, the score is estimated by using Riemann Approximation:

$$\hat{\mathrm{Attr}}(w_i^l) = \frac{\bar{w}_i^l}{m} \sum_{k=1}^{m} \frac{\partial P_q(\frac{k}{m} \bar{w}_i^l)}{\partial w_i^l} \quad (4)$$

where m is the number of the estimation steps. Finally, they identify a coarse set of knowledge neurons whose attribution scores are greater than a threshold t.

4 Locating Key Neurons in Autoregressive LLMs

While Knowledge Attribution (Dai et al., 2022) effectively identifies neurons linked to factual queries, its applicability is limited to Encoder-only architectures, and it mandates the output to be a single-token word. In response to these constraints, we propose a simple yet effective pipeline named **Attribution-Inverse Cluster Attribution (NA-ICA)**, which is architecture-agnostic and capable of handling long-form generation. The overall framework is shown in Figure 1. NA-IQA first resorts to the proxy task of multi-choice QA to deal with long-form answers. Subsequently, the framework extracts key neurons for each query using our designed NA-ICA score.

4.1 Multi-Choice QA Transformation

Given the biological question *"The energy given up by electrons as they move through the electron transport chain is used to?"*, the correct answer can be the long-form text *"produce ATP"*. To deal with long-form answers, we advocate for the transformation of questions and their corresponding answers into a multiple-choice framework, as illustrated in Figure 1. This approach involves the generation of incorrect options by randomly sampling answers within the same domain. Following this, the LLM is prompted to produce only the option letter. Subsequently, we investigate the key knowledge neurons correlated with the input query. To mitigate the impact of randomness, we devise multiple prompt templates and systematically shuffle the order of options to prevent the model from learning spurious correlations based on option letters. These prompt templates are detailed in Table A1.

4.2 Neuron Attribution-Inverse Cluster Attribution

In our pursuit of locating neurons associated with specific queries, we compute the score of NA-ICA for each neuron, drawing inspiration from the principles of TF-IDF (Salton, 1983) for keyword extraction. Beginning with a given query, NA-ICA employs *neuron attribution* to derive a coarse group key neurons, termed as *clusters*. Each neuron within this cluster is assigned an attribution score indicative of its relevance to the query, akin to the computation of term frequency. Given our objective of identifying critical neurons closely correlated with their respective queries, we use *inverse cluster attribution* to filter out neurons shared across different clusters (or queries). Finally, we find some neurons appear across multiple clusters, embodying common knowledge or sense, which we denote as *Common Neurons*. Further refinement of key neuron extraction involves the exclusion of these common neurons, which can enhance the precision of identifying critical neural correlates.

Neuron Attribution To extend our methodology to Gated Linear Units (GLUs), which comprise two linear transformations followed by a gating mechanism, we adapt the Knowledge Attribution approach (Eq 5). In GLUs, the linear transformations involve computing a linear combination of input features, denoted by $f = \mathbf{X}\mathbf{W}^U$. Additionally, the gating mechanism, represented by $g = \text{SiLU}(\mathbf{X}\mathbf{W}^G)$, determines the extent to which each input component should be forwarded, thereby enabling the model to emphasize important features while suppressing irrelevant ones. To compute the relevant attribution, we can use either $\frac{\partial P_q}{\partial f}$ or $\frac{\partial P_q}{\partial g}$ and we choose to use the former since our empirical study shows it can obtain better key neurons (see details in the Table A3). Given a query q, instantiation using our templates yields a query set $\mathcal{Q} = \{q_1, q_2, ..., q_{|\mathcal{Q}|}\}$, and the attribution score of the neuron n_i^l can be denoted as:

$$\text{na}(n_i^l) = \sum_{j=1}^{|\mathcal{Q}|} \frac{\bar{f}_i^l}{m} \sum_{k=1}^{m} \frac{\partial P_{q_j}(\frac{k}{m}\bar{f}_i^l)}{\partial f_i^l} \qquad (5)$$

Here, we sum up the scores of different instantiated templates together as the final attribution score.

Inverse Cluster Attribution With the attribution score, we can obtain a list of coarse clusters for each query $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|})\}$, where c is a cluster that consists of neurons whose attribution score is higher than some threshold t. The frequent appearance of some neurons across queries of different fields reveals that they are not critical neurons to the input query. To decrease their impact, we calculate the inverse cluster attribution:

$$\text{ica}(n_i^l) = \log \frac{|\mathcal{C}|}{|\{c : c \in \mathcal{C} \text{ and } n_i^l \in c\}| + 1} \qquad (6)$$

Common Neurons We observe that some neurons with a relatively high NA-ICA score are still shared across clusters. Through case studies (as shown in Table 4), we demonstrate that they express commonly used concepts such as option letters (“`A`” and “`B`”) or stop words (“`and`” and “`the`”). Therefore, we count the frequency of each neuron across clusters. If the frequency is higher than the $u\%$ of total clusters, we assign the given neuron into the common neuron set.

Key Neurons Given a query, the NA-ICA of a neuron can be computed as :

$$\text{naica}(n_i^l) = \text{na}(n_i^l) \times \text{ica}(n_i^l) \tag{7}$$

We select top-v neurons with the highest score from the detected cluster and further remove common neurons to refine the key neuron set.

5 Analyzing Detected Key Neurons

5.1 Dataset Construction

We construct two datasets to locate knowledge neurons that cover two different categories: *subject domains and languages*.

Domain Dataset is derived from MMLU (Hendrycks et al., 2020), a multiple-choice QA benchmark designed to evaluate models across a wide array of subjects with varying difficulty levels. The subjects encompass traditional disciplines such as mathematics and history, as well as specialized fields like law and ethics. In our study, we select six high school exam subjects from the test set: *Biology, Physics, Chemistry, Mathematics, Computer Science*, and *Geography*.

Language Dataset is adapted from Multilingual LAMA (Kassner et al., 2021), which is a dataset to investigate knowledge in language models in a multilingual setting covering 53 languages. We select six languages for the `birth_place` relation: *Arabic, English, French, Japanese, Russian* and *Chinese*.

To mitigate sensitivity to prompts and option orders, each query is instantiated with multiple distinct templates (as shown in Table A1), and the option orders are shuffled each time. The statistics of our datasets are shown in Table 1 and examples can be found in Table A2.

Domain	Bio	Phy	Chem	Math	CS	Geo	Total
Num	100	100	100	100	52	100	552
Language	Ar	En	Fr	Ja	Ru	Zh	Total
Num	100	100	100	100	100	100	600

Table 1: Statistics of our constructed datasets.

5.2 Baselines

We compare our NA-ICA to three other neuron-level baselines[1]: **Random Neurons** are randomly selected from FFNs and we make sure they have the same number of neurons of NA-ICA; **Kowledge Neurons*** is adapted from knowledge attribution (Dai et al., 2022) by using multi-choice QA task; **NA-ICA *w/* Common Neurons** is a variant without removing common neurons.

5.3 Experimental Settings

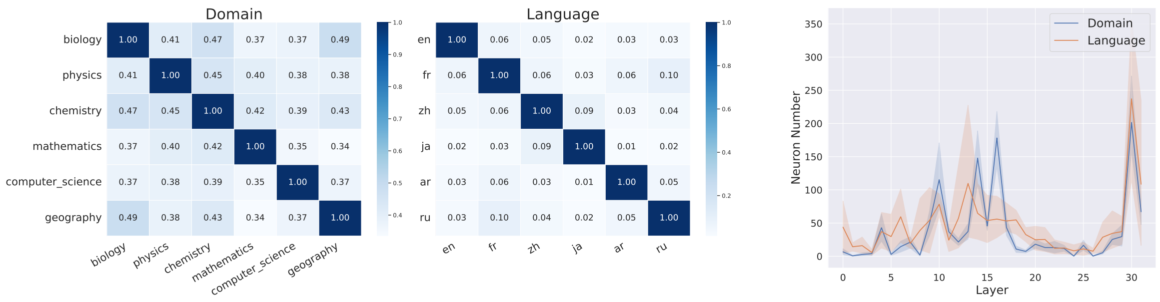
We mainly study the knowledge neurons in LLaMA-7B (Touvron et al., 2023) and we use the instruction-tuned version so that the model is more responsive to our prompts. LLaMA-7B consists of 32 layers with the FFN hidden dimension of 11008. Besides, we also conduct experiments for Mistral-7B (Jiang et al., 2023) to validate whether that our method can obtain consistent findings over different models. Note that our framework can be easily extended to larger-size LLMs.

As for the used hyper-parameters, the number of estimation steps was set to $m = 16$ and the attribution threshold t to 0.2 times the maximum attribution score. The template number was $|\mathcal{Q}| = 3$, the frequency u for obtaining common neurons was 30%, and the top-v for select key neurons was 20. We ran all experiments on three NVIDIA-V100. It took 120 seconds on average to locate neurons for a query with three prompt templates.

5.4 Statistics of Detected Key Neurons

Table 3 presents the number of detected key neurons for each domain and language, averaging between 12 and 17 neurons. Figure 2a illustrates the overlap rates among different domains and languages. It is evident that domains exhibit higher overlap rates compared to languages, reflecting interconnected and interdisciplinary nature. For instance, the overlap rate between biology and geography is 0.49, attributable to fields like biogeography, which examines the distribution of species

[1]We do not compare to ROME (Meng et al., 2022a) since it locates layers instead of neurons



(a) Overlap Rate

(b) Layer Distribution

Figure 2: Overlap rates and distributions of found key neurons.

	Domain						Language					
	Boost			Suppress			Boost			Suppress		
Model	⇑ Related	⇑ Unrelated	Ratio	⇓ Related	⇓ Unrelated	Ratio	⇑ Related	⇑ Unrelated	Ratio	⇓ Related	⇓ Unrelated	Ratio
Random Neurons	-0.03	-0.03	1.0	+0.06	+0.11	0.55	+0.08	+0.04	2.0	-0.01	-0.01	1.0
Knowledge Neurons* (2022)	+932.05	+921.84	1.0	-85.70	-85.34	1.0	+1081.33	+161.98	6.7	-86.74	-48.18	1.8
NA-ICA *w/* Common Neurons	+919.03	+328.49	2.8	-59.34	-33.59	1.8	+606.54	+54.84	10.4	-71.45	-8.40	8.5
NA-ICA	+77.23	+17.55	4.4	-27.65	-4.95	5.6	+218.03	+5.20	41.9	-54.64	+3.71	15.2

Table 2: Average probability percentage changes of the correct answers by boosting (⇑) or suppressing (⇓) the key neurons. The Ratio metric is calculated by $\frac{|\text{Related}|}{|\text{Unrelated}|}$, and a bigger value shows a higher impact of the detected neurons. The LLM here is LLaMA-7B (Touvron et al., 2023)

Domain	Bio	Phy	Chem	Math	CS	Geo	Avg
Num	13.1	13.3	12.8	11.1	14.3	12.7	12.9
Language	Ar	En	Fr	Ja	Ru	Zh	Avg
Num	12.4	14.4	12.7	16.6	15.8	15.0	14.5

Table 3: Average number of key neurons.

and ecosystems in geographic space. Regarding layer distribution, the key neurons are predominantly located in the middle layers (15-18) and the top layers (around 30), as depicted in Figure 2b.

5.5 Key Neurons Can Impact the Prediction

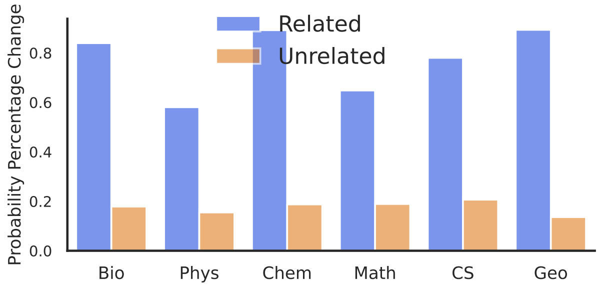
To validate the impact of our identified key neurons, we replicate the experiments by Dai et al. (2022), updating the values of key neurons using two methods: given a query and the value of $\bar{f}_i^l$, we either (1) boost the key neurons by doubling the value $f_i^l = 2 \times \bar{f}_i^l$; or (2) suppress the key neuron by making $f_i^l = 0$. For each query, we record the percentage change in the probability of the correct answer, thereby assessing the extent to which the key neurons influence the predictions of LLMs. We compare our NA-ICA approach to other baseline methods and include a control group to determine whether the same key neurons affect the predictions of randomly selected queries from unrelated fields (*Unrelated*).

Table 2 presents the overall performance of various methods. Our NA-ICA method consistently outperforms other baselines, evidenced by its higher impact ratio. This indicates that our identified key neurons significantly affect the probability of correct answers while exerting a relatively low impact on unrelated queries. For instance, our method achieves a boosting ratio of 41.9 on the language dataset, the highest among the baselines. Additionally, common neurons affect both related and unrelated queries, and their removal results in clear performance improvements.

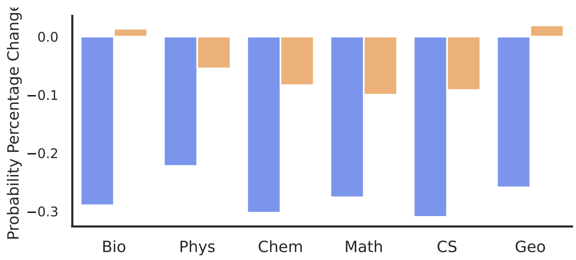
Furthermore, Figure 3 illustrates the percentage change in probability for each domain and language. Again, we can clearly observe the effectiveness of our detected key neurons. Additionally, we performed supplementary experiments on Mistral-7B. The results, presented in Figure A2, consistently support our conclusions.

5.6 Are There Localized Regions in LLMs?

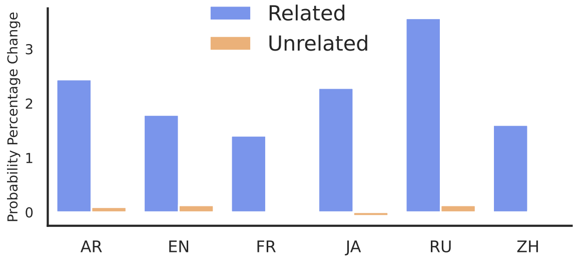
Given our ability to identify key neurons for each query, it is intriguing to explore whether LLMs exhibit localized regions for each domain or language, analogous to the functional localizations in the human brain (Brett et al., 2002). To investigate this, we visualize domain- or language-specific neurons on a 2D geographical heatmap. The width of the heatmap corresponds to the dimension of FFNs in LLaMA-7B (11008), and the length represents



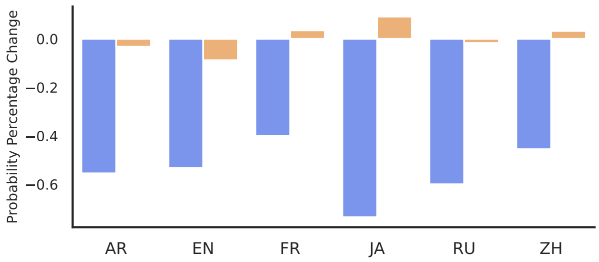
(a) Boosting key neurons of domains



(b) Suppressing key neurons of domains



(c) Boosting key neurons of languages



(d) Suppressing key neurons of languages

Figure 3: The correct probability percentage change across different domains and languages. The LLM here is LLaMA-7B (Touvron et al., 2023)

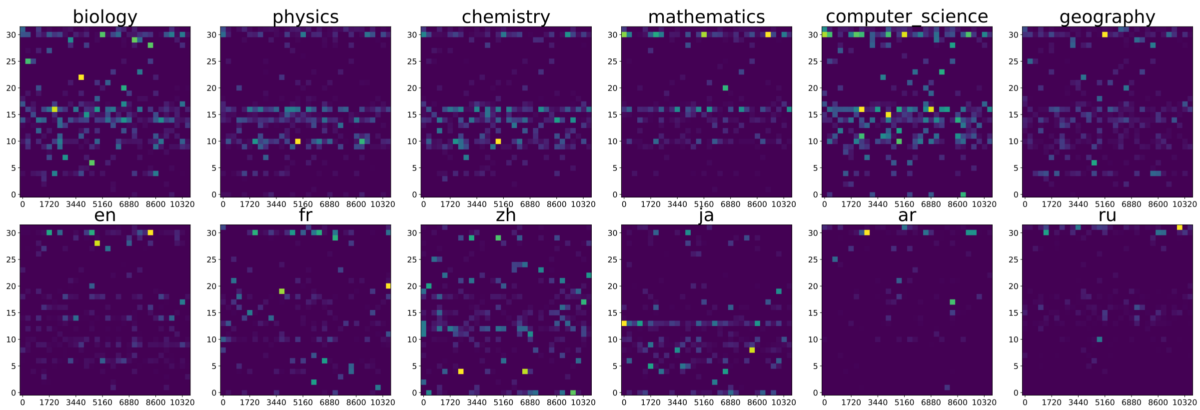


Figure 4: Geographical heatmap of detected key neurons for different domains and languages. The value is calculated by our naica(n_i^l). The LLM here is LLaMA-7B (11008×32) (Touvron et al., 2023)

the layer depth (32). We accumulate the value of naica(n_i^l) to populate the heatmap. Figure 4 displays the geographical locations of key neurons in LLaMA-7B across various academic domains and languages. The distribution of key neurons appears sparse but with distinct regions, particularly for different domains. Notably, certain regions are visible in the middle layers (10-15), suggesting specific neuron patterns. In contrast, language neurons are more sparsely distributed with smaller regions, and languages like Arabic and Russian exhibit less localized properties. Apart from visualizing the geographical location of key neurons, we also analyze the semantic location using their associated vector values in $\mathbf{W}^D$. Our findings suggest that there are no apparent clusters across different domains, as these values likely represent intermediate states in a subspace distinct from the one used for final token prediction. Future work should consider new ways to map neurons to more discriminative semantic spaces. The details are provided in Appendix A.

5.7 The Function of Common Neurons

To gain insights into the function of common neurons, we also visualize their locations within LLaMA-7B. Figure 5 shows the common neurons for the domain and language dataset. We can observe that they tend to appear at the top layer. To further understand their meanings, we project the matrix $\mathbf{W}^D$ in Equation 1 to the vocabulary space and select the top-k tokens with the highest probability. Table 4 lists the predicted tokens, which

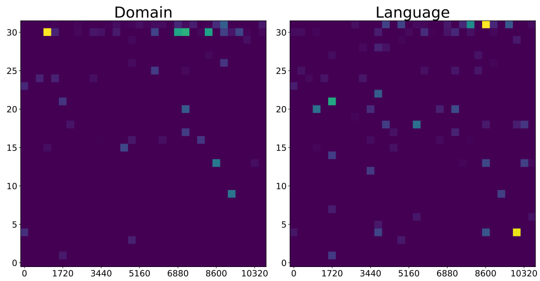


Figure 5: The distribution of common neurons.

Neuron	Top-k tokens
n^{31}_{2725}	_in, _and, _to, _for, _today, _at, _as
n^{31}_{10676}	_July, _June, _March, _April, _November
n^{30}_{10075}	., _, (, :,), [, -
n^{31}_{5202}	_respectively, _while, _and, _initially
n^{31}_{5778}	_C, C, _c, c, '_ced'
n^{31}_{7670}	_B, B, _Bill, _Bh, '_Bureau'

Table 4: Tokens predicted by the common neurons.

include common words, punctuation marks, and option letters. These findings reinforce the notion that common neurons are not critical for specific queries.

6 Potential Applications

We provide two usage examples to showcase the potential applications of our detected key neurons: *Knowledge Editing* and *Neuron-Based Prediction*.

6.1 Knowledge Editing

We adjust the values of key neurons by either boosting or suppressing them to determine if we can change the prediction of a query from incorrect to correct or vice versa. Table 5 presents the success rates of knowledge editing on our constructed language datasets. Our observations indicate that NA-ICA achieves higher success rates on related queries and lower rates on unrelated queries, demonstrating that our method outperforms other baselines. In contrast, the baseline of knowledge neurons cannot significantly differentiate related and unrelated queries.

6.2 Neuron-Based Prediction

In our second case study, we test whether the correct answers to domain-specific questions can be predicted solely based on the activity of the associated domain-specific neurons. To this end, we make predictions on multiple-choice questions by selecting the option with the overall highest gradient to the key neurons for the given domain. We experiment on a specifically constructed

	Boost		Suppress	
Model	⇑ Related	⇑ Unrelated	⇓ Related	⇓ Unrelated
Random Neurons	0.37	0.10	0.54	0.27
Knowledge Neurons (2022)	14.73	11.76	16.19	14.78
Ours	10.06	1.78	20.14	1.60

Table 5: Successful rates of knowledge editing.

Model	Biology (Acc.)	Chemistry (Acc.)	Geography (Acc.)
Random guess	0.25	0.25	0.25
Prompt-based model pred.	0.96	0.71	0.89
Neuron-based pred.	0.96	0.67	0.89

Table 6: Accuracy of neuron-based prediction on selected domains in comparison with the standard prompt-based model prediction. The LLM here is LLaMA-7B.

MMLU (Hendrycks et al., 2020) validation set with a different set of questions than those used to determine the key domain neurons (see Appendix B for details on our experimental strategy). The results are summarised in Table 6. We observe that the accuracy of the neuron-based predictions is very close to the accuracy of the prompt-based method of using the entire model (the used templates are shown in Table A1). This suggests that the activity of identified neurons can be indicative of the model's performance on a given task. Investigating how this finding could be leveraged in applications like fact-checking and hallucination detection presents a promising line of future work.

7 Conclusion

In this study, we introduce a novel framework, NA-ICA, for identifying key neurons in contemporary autoregressive language models, such as LLaMA and Mistral. NA-ICA leverages a multi-choice QA proxy task to address the complexity of long-form answers, extending beyond simple factual entities. Meanwhile, it adopts strategies of inverse cluster attribution and common neuron removal to refine key neurons. To validate our approach, we curated two datasets encompassing diverse domains and languages. Our experimental results show that NA-ICA outperforms existing baselines in identifying query-relevant neurons. Additionally, this study pioneers the exploration of localized knowledge regions in LLMs and demonstrates the potential usages of identified key neurons in applications such as knowledge editing and neuron-based prediction. We hope that our findings are beneficial for further research in understanding the knowledge mechanisms underlying LLMs.

Limitations

In our study, we employ a multi-choice QA proxy task to investigate the long-form knowledge stored in LLMs. Although our framework can effectively detect key neurons, future research needs to address the challenge of authentic open-ended generation, which remains a significant area for development. Additionally, despite our efforts to eliminate common neurons, some neurons within the identified key neuron set still correspond to option letters. This indicates that our current method requires further refinement to remove these spurious key neurons. Moreover, the language dataset used in our study is limited to the `Birth_place` relation. To gain a more comprehensive understanding of multilingual knowledge in LLMs, future work should include a broader range of relations. This expansion will enable a more thorough investigation into the diverse types of knowledge encoded in these models across different languages.

References

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. Gradient-based attribution methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 169–191.

Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. 2023. Discovering knowledge-critical subnetworks in pretrained language models. *arXiv preprint arXiv:2310.03084*.

Jan G Bjaalie. 2002. Localization in the brain: new solutions emerging. *Nature reviews neuroscience*, 3(4):322–325.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. An interpretability illusion for bert. *arXiv preprint arXiv:2104.07143*.

Matthew Brett, Ingrid S Johnsrude, and Adrian M Owen. 2002. The problem of functional localization in the human brain. *Nature reviews neuroscience*, 3(3):243–249.

Lihu Chen, Gael Varoquaux, and Fabian Suchanek. 2023. The locality and symmetry of positional encodings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14313–14331.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17817–17825.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2024. Mathematical capabilities of chatgpt. *Advances in Neural Information Processing Systems*, 36.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Ali Gholipour, Nasser Kehtarnavaz, Richard Briggs, Michael Devous, and Kaundinya Gopinath. 2007. Brain functional localization: a survey of image registration techniques. *IEEE transactions on medical imaging*, 26(4):427–451.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020. X-factr: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959.

Yiming Ju and Zheng Zhang. 2023. Klob: a benchmark for assessing knowledge locating methods in language models. *arXiv preprint arXiv:2309.16535*.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual lama: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, 3(29):861.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Pouya Pezeshkpour. 2023. Measuring and modifying factual knowledge in large language models. *arXiv preprint arXiv:2306.06264*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Siyu Ren and Kenny Zhu. 2022. Specializing pre-trained language models for better relational reasoning via network pruning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2195–2207.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Gerard Salton. 1983. Introduction to modern information retrieval. *McGraw-Hill*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. 2021. Can language models be biomedical knowledge bases? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4723–4734.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024a. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, Zhiyuan Liu, Ruobing Xie, Maosong Sun, and Jie Zhou. 2023. Emergent modularity in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4066–4083.

Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024b. Unveiling linguistic regions in large language models. *arXiv preprint arXiv:2402.14700*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

A Latent Value Analysis

As previously found by (Geva et al., 2021), transformer feed-forward layers can be viewed as key-value memory units, with hidden activations acting as coefficients for the individual memories stored in $\mathbf{W}^D$. Thus, a natural question to explore is what are the properties of the memory cells associated with the key neurons for the different domains and if they are clustered in the corresponding semantic space.

As a first step in our analysis, we visualise the $\mathbf{W}^D$ vectors associated with the knowledge neurons from the different domains using UMAP (McInnes et al., 2018) for dimensionality reduction (with cosine similarity used as the distance metric). For comparison, we additionally include the vectors from the unembedding matrix. The results are shown in Figure A1. As can be seen from the figure, the distribution of the vectors associated with key knowledge neurons appears to be significantly different from that of vector unembeddings. Thus, it appears that the contents of the internal memory cells used by LLaMA 2 are not directly aligned with the candidate output tokens.

Since the 2D visualisation produced by UMAP might not accurately reflect the true properties of the data manifold, we additionally examined the highest-likelihood tokens for the key domain neuron memory cells. These were computed by directly applying the Llama 2 unembedding layer to the vectors stored in these cells. We found the resulting tokens rather uninterpretable, including tokens like `textt`, `archivi`, `_Kontrola`, `_totalité` or `_Einzeln`. Upon closer investigation, we found these to be closely matching the set of unembedding vectors with the largest vector norms (which we would expect to generally receive higher likelihoods when multiplied with vectors not aligned with any of the unembeddings). This seems to further support a conjecture that the memory cell vectors associated with the located domain-specific neurons might capture intermediate data in a subspace different from the one used for the final token prediction. Apart from the above tokens, we also found option letters `A`, `B`, `C` and `D` to be represented in the highest-likelihood tokens. This suggests that some neurons within the identified key neuron set may still correspond to option letters, as mentioned in the Limitations section.

We leave further investigation and confirmation of these findings for future work.

B Neuron-Based Prediction Details

In the neuron-based prediction case study, we experiment on the MMLU (Hendrycks et al., 2020) validation set to ensure there is no overlap between the dataset used to mine the key neurons and the test set. Thus, the considered domain neurons were determined based on queries not used for this experiment. As a further post-processing step, we randomly select three options from other domains to replace the incorrect options in each query. Additionally, we manually remove questions that become invalid due to this post-processing, including queries such as *“Which of the following is LEAST valid?”* and *“All of the following statements are true EXCEPT”*. These operations result in $\sim$20 test samples per domain. To perform the neuron-based prediction, we compute the gradient of the probability of each option token with respect to the key neurons for the domain of the considered query, and select the option with the highest total gradient.

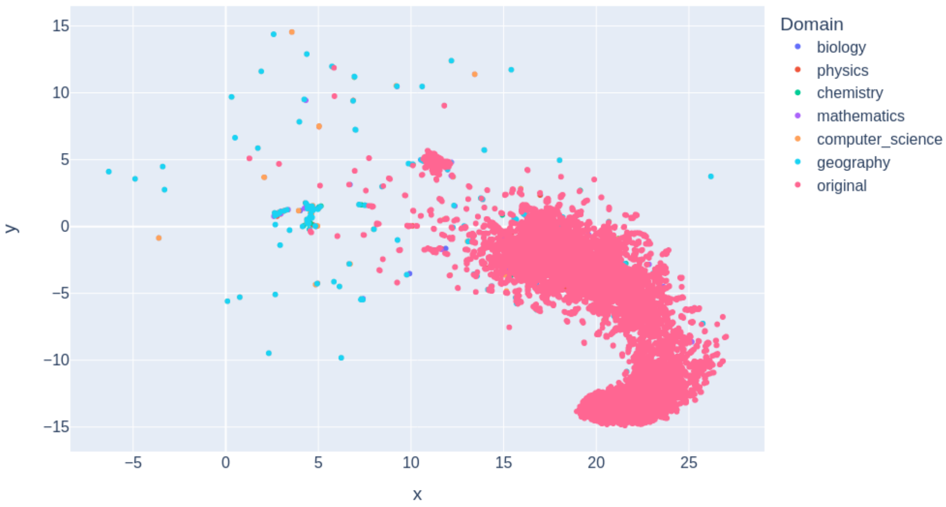
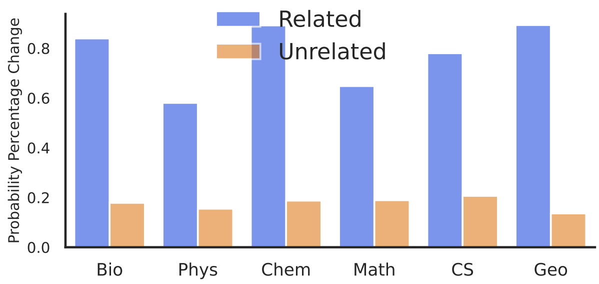
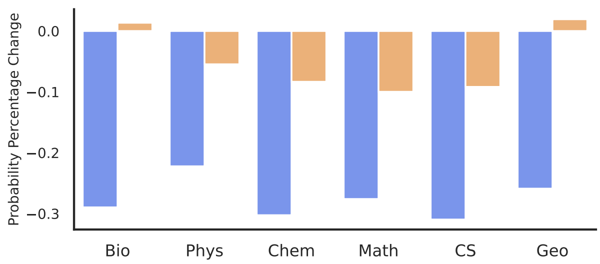


Figure A1: UMAP visualisation of $\mathbf{W}^D$ vectors associated with the knowledge neurons and the token unembeddings

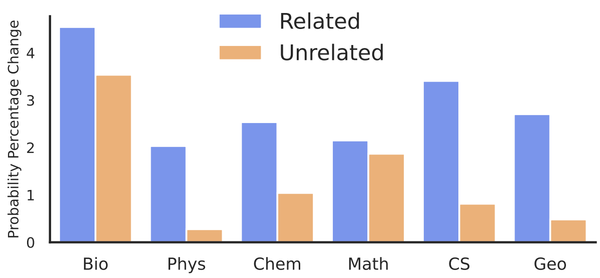


(a) Boosting key neurons of domains

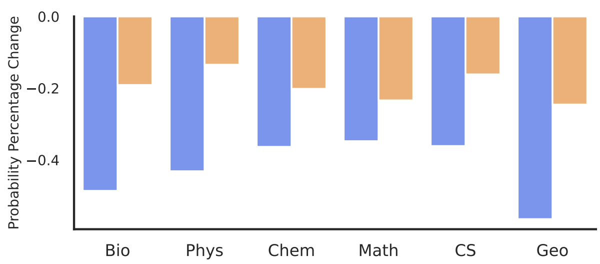


(b) Suppressing key neurons of domains

Figure A2: The correct probability percentage change across different domains. The LLM here is Mistral-7B (Jiang et al., 2023)



(a) Boosting key neurons of domains



(b) Suppressing key neurons of domains

Figure A3: A comparison study of using $\frac{\partial P_q}{\partial g}$ to compute NA-ICA scores. The LLM here is LLaMA-7B (Touvron et al., 2023).

Num	Template
Domain Prompt 1	*You will be asked a multiple-choice question. Respond with the letter which corresponds to the correct answer, followed by a period. There is no need to provide an explanation, so your response should be very short.\nNow here is the question:\n*`{Question}`*\n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*
Domain Prompt 2	*Prepare to answer a multiple-choice question. Provide the letter that corresponds to the correct answer, followed by a period. Keep your response brief; no explanations are necessary.\nHere is the question:\n*`{Question}`*\n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*
Domain Prompt 3	*Below is a multiple-choice question. Respond with the letter that best answers the question. Keep your response brief, stating only the letter corresponding to your answer, followed by a period, with no explanation.\nThe question is:\n*`{Question}`*\n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*
Language Prompt 1	*You will be asked a multiple-choice question. Respond with the letter which corresponds to the correct answer, followed by a period. There is no need to provide an explanation, so your response should be very short. \nNow here is the question:\n*`{Question}` *\nHere the [Y] is most likely to be? \n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*
Language Prompt 2	*Prepare to answer a multiple-choice question. Provide the letter that corresponds to the correct answer, followed by a period. Keep your response brief; no explanations are necessary. \nNow here is the question:\n*`{Question}` *\nHere the [Y] is most likely to be? \n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*
Language Prompt 3	*Below is a multiple-choice question. Respond with the letter that best answers the question. Keep your response brief, stating only the letter corresponding to your answer, followed by a period, with no explanation. \nNow here is the question:\n*`{Question}` *\nHere the [Y] is most likely to be? \n A.* `{A}`*\n B.* `{B}`*\n C.* `{C}`*\n D.* `{D}`*\nResponse:*

Table A1: Prompt templates for constructing multi-choice QA datasets. We use ChatGPT to translate English templates to other languages.

Field	Question	Options
Biology	*The energy given up by electrons as they move through the electron transport chain is used to?*	A. make glucose B. make NADH C. produce ATP D. break down glucose
Physics	*An object is placed 100 cm from a plane mirror. How far is the image from the object?*	A. 50 cm B. 200 cm C. 100 cm D. 300 cm
Chemistry	*Three half-lives after an isotope is prepared:*	A. 12.5% of the isotope decayed B. 25% of the isotope decayed C. 25% of the isotope is left D. 12.5% of the isotope is left
Mathematics	*Suppose the graph of f is both increasing and concave up on a <= x <= b. Then, using the same number of subdivisions, and with L, R, M, and T denoting, respectively, left, right, midpoint, and trapezoid sums, it follows that:*	A. R <= T <= M <= L B. L <= M <= T <= R C. R <= M <= T <= L D. L <= T <= M <= R
Computer Science	*A programmer is writing a program that is intended to be able to process large amounts of data. Which of the following considerations is LEAST likely to affect the ability of the program to process larger data sets?*	A. How long the program takes to run B. How many programming statements the program contains C. How much storage space the program requires as it runs D. How much memory the program requires as it runs
Geography	*The tendency for migration to decrease with distance is called?*	A. push factors. B. migration selectivity. C. distance decay. D. pull factors.
English	*Sergey Lavrov was born in [Y]. Here the [Y] is most likely to be?*	A. Montevideo B. Bengaluru C. Parsons D. Moscow

Table A2: Examples in our constructed datasets. For the language dataset, we only show one English example as multilingual samples are obtained bu using traslator (Kassner et al., 2021)