Cracking Factual Knowledge: A Comprehensive Analysis of Degenerate Knowledge Neurons in Large Language Models

Anonymous ACL submission

Abstract

Knowledge neuron theory provides a key approach to understanding the mechanisms of factual knowledge in Large Language Models (LLMs), which suggests that facts are stored within multi-layer perceptron neurons. This paper further explores Degenerate Knowledge Neurons (**DKNs**), where distinct sets of neurons can store identical facts, but unlike simple redundancy, they also participate in storing other different facts. Despite the novelty and unique properties of this concept, it has not been rigorously defined and systematically studied. Our contributions are: (1) We pioneer the study of structures in knowledge neurons by analyzing weight connection patterns, providing a comprehensive definition of DKNs from both functional and structural aspects. (2) Based on this definition, we develop the *Neuronal Topology Clustering* method, leading to a more accurate DKN identification. (3) We confirm the practical applications of DKNs: guiding LLMs to learn new knowledge and relating to LLMs' robustness against input errors[1].

1 Introduction

Large language models (LLMs) are believed to store extensive factual knowledge (Touvron et al., 2023; OpenAI et al., 2023; Han et al., 2021), yet the mechanisms of knowledge storage in LLMs remain largely unexplored. Dai et al. (2022) propose that some multi-layer perceptron (MLP) neurons can store "knowledge". As shown in Figure 1, neurons a through g that consistently activate in response to the fact $\langle$*COVID-19*, *dominant variant*, *Delta*$\rangle$ are termed knowledge neurons (KNs). Chen et al. (2024a) further explore the properties of knowledge neurons and discover degenerate knowledge neurons (**DKNs**). In Figure 1, distinct pairs of knowledge neurons (e.g., $\{a, b, e, f\}$ and $\{c, d\}$) can store identical facts, exhibiting a form of redundancy. However, this goes beyond simple redundancy, as each subset may also store other facts - for example, $\{a, b, e, f\}$ stores "other facts" that are distinct from those encoded by $\{c, d\}$. This property aligns with the definition of **degeneracy**.

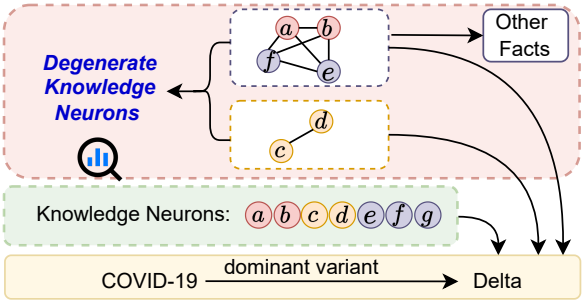

Figure 1: Knowledge neurons $\{a, \ldots, g\}$ are identified for storing a fact, with distinct subsets of these KNs ($\{a, b, e, f\}$ and $\{c, d\}$) capable of independently storing the same fact. These subsets constitute a DKN.

While Chen et al. (2024a) have conducted some exploration on DKNs, their definition and acquisition method for DKNs still face two issues. (1) *Numerical Limitation*: They constrain each DKN's element to contain just two knowledge neurons. However, factual knowledge often requires the coordination of more than two neurons for storage (Allen-Zhu and Li, 2023). (2) *Connectivity Oversight*: Their analysis focuses solely on individual neurons, overlooking the role of inter-neuronal connections. However, knowledge expression requires the interaction of multiple neurons (Zhu and Li, 2023), and thus it is necessary to consider the connectivity structure between neurons.

To address these two issues, **we first provide a comprehensive definition of degenerate knowledge neurons from two perspectives(§3).** (1) *Functionally*, we define Base Degenerate Components (**BDCs**) as subsets of KNs that can independently express the same fact. For example, $\{a, b, e, f\}$ and $\{c, d\}$ in Figure 1 constitute two

[1]Code and dataset will be de-anonymized: https://anonymous.4open.science/r/DKN-0E6C.

BDCs. A DKN is defined as the set of these mutually degenerate BDCs. (2) *Structurally*, as shown in Figure 1, BDCs like $\{a, b, e, f\}$ and $\{c, d\}$ differ in KN number and connection tightness. To quantify DKNs' structural properties, we define neuron distances based on connection weights and analyze the structural properties of neuron sets accordingly.

Based on our definition, **we introduce the Neuronal Topology Clustering (NTC) method to identify degenerate knowledge neurons (§4).** NTC comprises two steps: (1) *Structurally*, clustering neurons into stable structural sets based on connection weights; (2) *Functionally*, filtering these sets to retain those that can effectively express facts. By incorporating structural information of neuron connections, NTC enables the formation of BDCs with flexible neuron cardinality and structures, addressing the previous two limitations and identifying DKNs more accurately.

Furthermore, we explore the applications of degenerate knowledge neurons, leading to two additional findings.

(1) **DKNs can guide LLMs to learn new knowledge (§5)**. Using timestamped facts,we first identify their corresponding DKNs at specific timestamps. Then, we use different timestamps and answers to fine-tune the model to learn this new knowledge. Our findings are: *(A)* Through full fine-tuning of the LLMs, we find that the neurons showing significant parameter changes largely overlap with the regions of DKNs. This demonstrates that LLMs indeed utilize DKNs to learn new knowledge. *(B)* Based on finding (A), we employ an efficient fine-tuning technique, freezing all neurons except DKNs. Compared to baselines, our DKN-guided fine-tuning approach achieves superior performance in knowledge update tasks.

(2) **DKNs relate to LLMs' robustness against input perturbations (§6)**. LLMs exhibit an intrinsic ability to resist perturbations, preserving partial accuracy when faced with deliberately perturbed queries. However, when we suppress (or enhance) the activation values or connection weights of DKNs, we observe resulting decrease (or increase) in the LLMs' answer probability for the perturbed queries. This indicates that DKNs relate to LLMs' robustness against input perturbations.

Our contributions can be summarized as follows:

- We pioneer the study of structures in knowledge neurons, providing a comprehensive definition of DKNs from both functional and structural aspects.
- We introduce the neuronal topology clustering method, leading to a more accurate DKN identification.
- We confirm the practical applications of DKNs: guiding LLMs to learn new knowledge and relating to LLMs' robustness against input errors.

2 Datasets and Models

We utilize the TempLama dataset (Dhingra et al., 2022) to analyze DKNs. Each data instance includes a relation name, a date, a query, and an answer, such as $\langle$*P37*, *September 2021*, *COVID-19*, *dominant variant*, _$\rangle$. Except for timestamps, our dataset matches the Lama (Petroni et al., 2019a, 2020) and mLama (Kassner et al., 2021) format used by Dai et al. (2022) and Chen et al. (2024a). See appendix B for further details. Regarding model selection, we choose GPT-2 (Radford et al., 2019) and LLaMA2-7b (Touvron et al., 2023), allowing us to test the generalization and of our methods and conclusions.

3 Definition of Degenerate Knowledge Neurons

Formalization Given a fact, we utilize the AMIG method (Chen et al., 2024a) to obtain knowledge neurons (KNs), denoting them as $\mathcal{N} = \{n_1, n_2, \ldots, n_k\}$, where n_i is a KN. For details of this method, see Appendix C. Let degenerate knowledge neurons (DKNs) be denoted as $\mathcal{D}$, containing s elements, $\mathcal{D} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$, where $\mathcal{B}_j = \{n_{j1}, n_{j2}, \ldots, n_{j,|\mathcal{B}_j|}\}$ is named as the Base Degenerate Component (BDC). Thus, this fact ultimately corresponds to a set of DKNs:

$$\mathcal{D} = \{\mathcal{B}_1, \ldots, \mathcal{B}_s\} = \{(n_{11}, \ldots, n_{1,|\mathcal{B}_1|}), \ldots, (n_{s1}, \ldots, n_{s,|\mathcal{B}_s|})\} \quad (1)$$

With this formalization, we now define DKNs through their functional and structural properties.

Functional Definition Degeneracy requires that each base degenerate component ($\mathcal{B}$) can independently express a fact. Let $Prob(\mathcal{B})$ be the LLMs' answer prediction probability when $\mathcal{B}$ is activated, then the functional definition of DKNs is:

$$Prob(\mathcal{D}) \approx Prob(\mathcal{B}_i), \forall i = 1, 2, \ldots, s \quad (2)$$

$$Prob(\emptyset) \ll Prob(\mathcal{B}_i), \forall i = 1, 2, \ldots, s \quad (3)$$

where Equation 2 indicates that activating any single $\mathcal{B}_i$ is sufficient to express the fact, and Equation 3 suggests that if all $\mathcal{B}$ are suppressed (i.e., activating the empty set $\emptyset$), the fact cannot be correctly expressed.

Structural Definition Zhu and Li (2023) argue that tightly connected neurons tend to store knowledge centrally, suggesting that DKNs may exhibit more closely interconnected connectivity patterns. Motivated by this insight, we introduce structural information that can characterize connectivity patterns between any neurons, and define DKNs from a structural perspective. For neurons A and B in layer l_A and layer l_B respectively, we define their distance d_{AB} under three distinct scenarios:

(1) *Adjacent Layer Distance*: For neurons in adjacent layers ($|l_A - l_B| = 1$), we define the distance as the reciprocal of their connection weight:

$$d_{AB} = |1/w_{AB}|, \quad \text{if } w_{AB} \neq 0 \tag{4}$$

This captures the intuition that stronger connections (larger weights) correspond to shorter distances.

(2) *Multi-layer Distance*: For neurons spanning multiple layers ($|l_A - l_B| > 1$), we employ a dynamic programming algorithm to find the shortest path between knowledge neurons:

$$d_{AB} = \min_{P \in \text{Paths}(\mathcal{N})} \sum_{(i,j) \in P} d_{ij} \tag{5}$$

where $\text{Paths}(\mathcal{N})$ encompasses all possible paths from A to B through the set of knowledge neurons $\mathcal{N}$. For any two knowledge neurons i and j, let $d_{ij}^{(k)}$ denote the minimum distance between them using at most k intermediate KNs. The optimal distance can be computed recursively:

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, \min_{m \in \mathcal{N}}\{d_{im}^{(k-1)} + d_{mj}^{(k-1)}\}\} \tag{6}$$

(3) *Same-layer Distance*: For neurons within the same layer ($l_A = l_B$), we set:

$$d_{AB} = \infty \tag{7}$$

This follows from LLMs' architectural constraint where information flows between layers rather than within a layer (Meng et al., 2022).

Based on these distance metrics, we construct an adjacency matrix $\mathcal{A}$ where each entry represents the distance d_{AB} between neurons A and B. We use this adjacency matrix $\mathcal{A}$ to structurally define a set of degenerate knowledge neurons $\mathcal{D}$, where

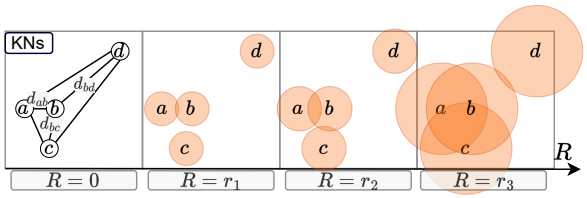

Figure 2: The clustering step of NTC method. The x-axis (R) represents the increasing distance threshold starting from 0. Circles with radius R are drawn around neurons, and intersecting circles indicate that the KNs are clustered together.

$\mathcal{A} \in \mathbb{R}^{k \times k}$, and k represents the number of knowledge neurons in $\mathcal{D}$. Intuitively, every element (base degenerate components, $\mathcal{B}$) within DKNs should demonstrate strong internal connections. Notably, this distance measurement approach applies to any set of neurons. As a pioneering structural exploration in KNs, we employ necessary simplifications, detailed in Appendix D.

4 Neuronal Topology Clustering

4.1 Algorithm of Neuronal Topology Clustering

Enumerating all possible knowledge neuron subsets to find optimal DKNs that satisfy Equation 2 and 3 incurs $O(2^n)$ complexity. Instead, we propose a two-stage approach using **structural** clustering followed by **functional** filtering. Inspired by topology data analysis (Edelsbrunner et al., 2008; Edelsbrunner, 2013; Chazal and Michel, 2021), we propose the Neuronal Topology Clustering (NTC) method for identifying DKNs (See Appendix E for topological background details).

Figure 2 informally illustrates clustering step. Given four knowledge neurons $\{a, b, c, d\}$ with fixed connection weights and an increasing distance threshold R starting from 0, we observe whether the KNs can cluster together as R changes. At $R = 0$, the KNs are isolated points. When $R = r_1 > d_{ab}$, $\{a, b\}$ form a cluster; at $R = r_2 > d_{bc}$, $\{a, b, c\}$ cluster together; and at $R = r_3 > d_{bd}$, $\{a, b, c, d\}$ form a single cluster. Notably, a wide range of R values maintains the $\{a, b, c\}$ cluster (from r_2 to r_3), indicating its stable existence. This stable cluster, suggesting a strong knowledge expression ability (Zhu and Li, 2023), is identified as a base degenerate component (BDC).

Formally, we structure our method in three steps. (1) In the initialization step, we begin with a distance threshold R starting from 0, and initialize an

empty set for degenerate knowledge neurons ($\mathcal{D}$):

$$R = 0, \mathcal{D} = \emptyset \tag{8}$$

(2) In the clustering step, as R increases, we track the evolution of clusters. For each cluster, we define R_1 as its initial formation radius and R_2 as the radius where it merges with either a new neuron or another cluster. The persistence of a cluster is calculated as: $R_p = R_2 - R_1$. During this process, we continuously record all knowledge neuron clusters along with their corresponding persistence values R_p. We then filter these clusters based on their persistence to obtain potential BDCs ($\mathcal{B}$):

$$\mathcal{D}_{\text{potential}} = \{\mathcal{B}_i | R_p(\mathcal{B}_i) > \tau_1\} \tag{9}$$

(3) In the final filtering step, we apply a threshold τ_2 for prediction probability. A potential BDC is added to $\mathcal{D}$ only if it satisfies Equation 2. The final set of $\mathcal{D}$ is thus defined as:

$$\mathcal{D} = \{\mathcal{B}_i | R_p(\mathcal{B}_i) > \tau_1 \text{ and } Prob(\mathcal{B}_i) \geq \tau_2\} \tag{10}$$

where $Prob(\mathcal{B}_i)$ is the prediction probability when $\mathcal{B}_i$ is activated. This dual-threshold filtering ensures that we identify BDCs that are both *structurally* stable (high persistence) and *functionally* significant (high prediction probability).

4.2 Experiments of DKNs Acquisition

Experimental settings Now, we verify the effectiveness of the neuronal topology clustering method through experiments. First, we identify degenerate knowledge neurons and plot their neuron distribution across model layers, as shown in Figure 3.

Then, to measure the degeneracy of $\mathcal{D} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$, we progressively suppress $\mathcal{B}_i$, varying the number from 1 to n. Based on Equations 2 and 3, we expect that suppressing any subset of $\mathcal{D}$ (i.e., 1 to n-1 $\mathcal{B}$) results in a small decrease in the LLMs' answer probability, while suppressing all $\mathcal{B}$ leads to a big decrease, exhibiting a "sudden change" pattern. To quantify this effect, we first calculate the relative drop in answer probability before (b) and after (a) suppression:

$$\Delta Prob = \frac{Prob_b - Prob_a}{Prob_b} \tag{11}$$

Then, to quantitatively measure this "sudden change" pattern, inspired by the concept of curvature in calculus, we propose a metric called Generalized Curvature Ratio (GCR):

$$GCR = \frac{|\Delta Prob_n - \mu(\Delta Prob_{[1,n-1]})|}{\sigma(\Delta Prob_{[1,n-1]})} \tag{12}$$

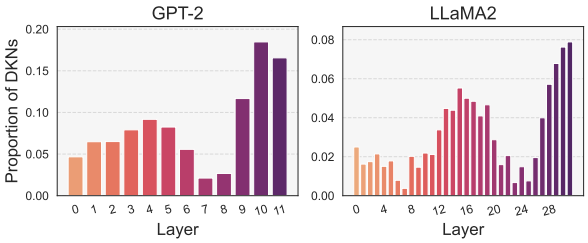

Figure 3: Distribution of DKNs across different layers in GPT-2 and LLaMA2-7b models (under NTC method).

where $\Delta Prob_n$ represents the probability drop when suppressing all BDCs, $\mu(\Delta Prob_{[1,n-1]})$ denotes the mean of probability drops when suppressing partial BDCs (1 to n-1), and $\sigma(\Delta Prob_{[1,n-1]})$ is their standard deviation. This metric captures the relative intensity of the final change compared to the overall variation pattern, with a higher GCR indicating better degeneracy. When $n = 2$, since only $\Delta Prob_1$ exists in the partial suppression set, we simply define GCR as $|\Delta Prob_2 - \Delta Prob_1|$. The overall GCR results are reported in Table 1. Figure 4 presents more fine-grained results, showing the changes in $\Delta Prob$ as the number of suppressed BDCs increases from 1 to n, which also aims to capture this "sudden change" pattern.

Baselines We select four clustering methods as baselines: K-Means (Ahmed et al., 2020), DBSCAN (Ester et al., 1996), Hierarchical Clustering (Murtagh and Contreras, 2012) and AMIG (Chen et al., 2024a). With the previously defined adjacency matrix $\mathcal{A}$ providing pairwise distances between neurons, these general-purpose clustering algorithms can be directly applied. In contrast, our NTC method is specifically designed for identifying DKNs, taking into account both their structural and functional properties. Additionally, we conduct significance tests to verify the statistical significance of the performance gap between our neuronal topology clustering method and the baselines (see Appendix F.1, Table 6).

Findings DKNs identified by NTC exhibit strongest degeneracy. (1) As shown in Table 1, NTC (our method) yields the largest average GCR of **26.82** compared to other baselines on GPT-2, notably $\sim \times 3$ higher than the strongest baseline (Hierarchical, 8.71). Statistical significance tests confirm that these differences are significant (Table 6). Across different DKN set cardinalities, NTC also generally achieves the highest GCR values.

(2) The fine-grained results in Figure 4 further validate this finding by demonstrating a clear "sud-

Model	Method	2	3	4	5	6	7	8	Average
GPT-2	DBSCAN	14.17	17.37	2.72	2.65	3.77	2.73	1.63	6.43
	Hierarchical	1.37	7.36	10.78	**13.78**	10.25	/	/	8.71
	K-Means	14.60	17.56	6.30	3.66	1.09	1.74	**1.80**	6.68
	AMIG	1.68	/	/	/	/	/	/	1.68
	NTC (Ours)	**45.10**	**58.58**	**14.57**	10.06	**19.28**	**13.31**	/	**26.82**
LLaMA2-7b	DBSCAN	8.41	6.00	4.30	2.79	3.92	4.65	4.00	4.87
	Hierarchical	**15.66**	5.33	/	/	/	/	/	10.50
	K-Means	13.30	8.73	5.19	2.76	5.15	2.60	5.55	6.18
	AMIG	2.98	/	/	/	/	/	/	2.98
	NTC (Ours)	12.80	**17.31**	**12.18**	**12.49**	**7.36**	**6.41**	**12.85**	**11.63**

Table 1: Average GCR values for different methods across varying cardinalities of DKN set (numbers 2 to 8 in table headings). Higher GCR values indicate stronger degeneracy. Best values are bold, second-best underlined, and "/" indicates that the method failed to generate a $\mathcal{D}$ with the specified cardinality. For significance tests, see Table 6.

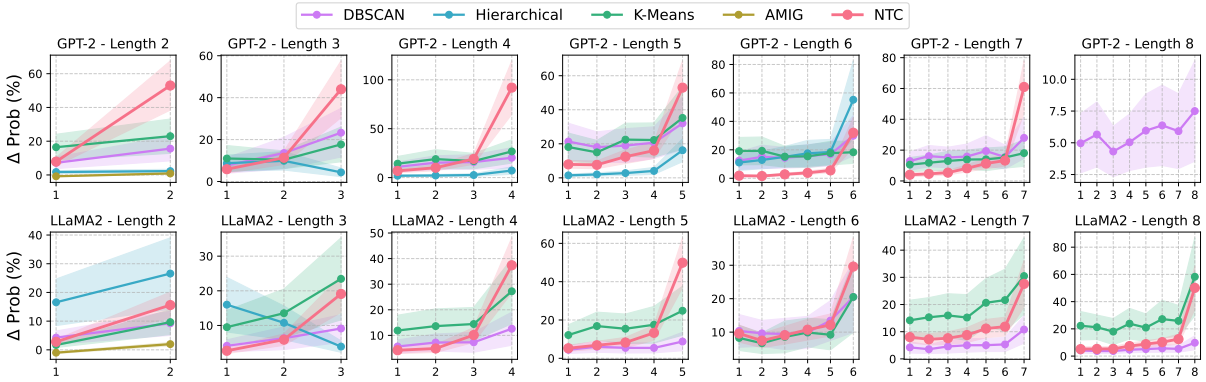

Figure 4: Relationship between $\Delta Prob$ and number of suppressed BDCs. Lower $\Delta Prob$ for partial suppression and higher $\Delta Prob$ for full suppression indicate stronger degeneracy. The red lines represent our NTC method.

den change" pattern. When using NTC (shown in red lines), $\Delta Prob$ remains low ($< 20\%$) as long as one $\mathcal{B}$ exists (points 1 to n-1), showing a significant increase ($\sim 40\%$ to 60%) only when all $\mathcal{B}$ are suppressed (final point n), rather than gradually increasing with the number of suppressed components. In other words, it exhibits the most pronounced "sudden change". Other methods lack this desirable degeneracy property, either showing higher $\Delta Prob$ when suppressing partial $\mathcal{B}$ or lower $\Delta Prob$ when suppressing all $\mathcal{B}$, failing to achieve both conditions simultaneously.

5 DKNs Can Guide LLMs to Learn New Knowledge

Motivation and Dataset Setup. In real-world scenarios, it is meaningful for LLMs to continuously learn or update new knowledge without forgetting old knowledge. In cognitive science, degeneracy is considered to be related to evolvability (Whitacre and Bender, 2010; Edelman and Gally, 2001; Whitacre, 2010; Mason, 2015). Since degenerate knowledge neurons (DKNs) effectively capture the degeneracy property in LLMs, and this property is linked to learning potential in biological systems, this biological insight inspires us to explore: can we use DKNs to study LLMs' ability to learn new knowledge?

To investigate this question systematically, we leverage the TempLama dataset (Dhingra et al., 2022), which incorporates timestamps that allow us to track knowledge evolution over time. Based on Table 5, we identify 3,334 facts that appear in both 2018 and 2019 timestamps. We utilize the 2018-timestamp facts to obtain DKNs and employ the 2019-timestamp facts as fine-tuning data. While timestamps may change without affecting answers, or LLMs may already possess knowledge across multiple timestamps, we follow the idea from CounterFact (Meng et al., 2022) and replace the answers for 2019-facts with incorrect ones to ensure that the new facts represents knowledge not yet mastered by LLMs.

To better simulate real-world learning scenarios, where LLMs typically learn from free-form text rather than triple-form factual knowledge, we use GPT-4 to convert the triple-form facts into natural language. This modified dataset, which we call

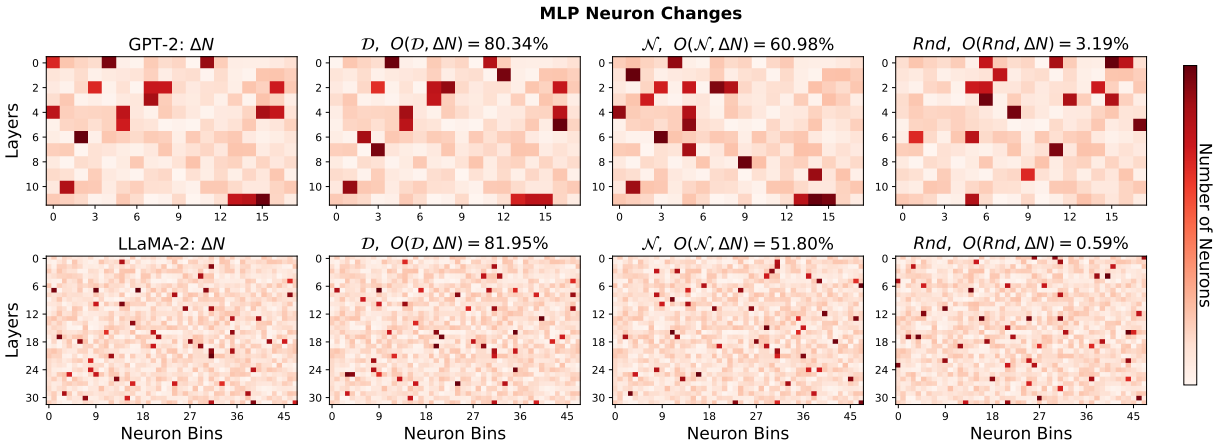

Figure 5: Results of the parameter changes experiment. The plots from left to right show the distributions of four types of neurons: ΔN, $\mathcal{D}$, $\mathcal{N}$, and Rnd. We group the neurons into bins for better visualization. For each row, we compare the distributions of $\mathcal{D}$, $\mathcal{N}$, and Rnd against ΔN, where higher similarity indicates greater overlap.

TempNarrativeLAMA, keeps the same relations and dates but transforms queries into free-form text (details in Appendix G).

We aim to verify two questions. Q1, whether LLMs primarily utilize DKNs during full fine-tuning, and Q2, whether unfreezing only DKNs for efficient fine-tuning can achieve better results. These two questions can mutually corroborate that DKNs can help LLMs learn new knowledge.

5.1 Overlap of DKNs and Parameter Changes

Experimental settings To address Q1, following the above setup, we first use the 2018-timestamp facts to obtain the corresponding set of DKNs for these facts. Then, we select data from TempNarrativeLAMA with the 2019 timestamp for full fine-tuning. We record the positions of neurons where significant parameter changes occur, denoted as ΔN:

$$\Delta N = \{n \,|\, \Delta P(n) > \tau_{\Delta N}\} \tag{13}$$

$$\Delta P(n) = \sqrt{\left(\frac{\|w_2^{\text{fc}}(n) - w_1^{\text{fc}}(n)\|}{\|w_1^{\text{fc}}(n)\|}\right)^2 + \left(\frac{\|w_2^{\text{proj}}(n) - w_1^{\text{proj}}(n)\|}{\|w_1^{\text{proj}}(n)\|}\right)^2} \tag{14}$$

where $\tau_{\Delta N}$ is a dynamic threshold, $\Delta P(n)$ indicates parameter change. $w_1^{\text{fc}}(n)$ and $w_1^{\text{proj}}(n)$ signify the feed-forward and projection weights of neuron n before fine-tuning, respectively, while $w_2^{\text{fc}}(n)$ and $w_2^{\text{proj}}(n)$ are their post-fine-tuning counterparts. Then, we calculate the overlap between ($\mathcal{D}$) and ΔN:

$$O(\mathcal{D}, \Delta N) = \frac{|\mathcal{D} \cap \Delta N|}{|\mathcal{D}|} \tag{15}$$

Our objective is to determine whether the neurons with significant parameter changes indeed show high overlap with $\mathcal{D}$ (degenerate knowledge neurons). For comparison, we choose the knowledge neurons ($\mathcal{N}$) and randomly chosen neurons (Rnd, the same number as $\mathcal{D}$) as baselines.

Findings Figure 5 illustrates the distribution of neurons across four distinct sets: ΔN, $\mathcal{D}$, $\mathcal{N}$, Rnd. Compared to the baselines ($\mathcal{N}$ and Rnd), $\mathcal{D}$'s distribution pattern shows the closest resemblance to ΔN. Quantitatively, $O(\mathcal{D}, \Delta N)$ achieves the highest value ($> 80\%$), surpassing $O(\mathcal{N}, \Delta N)$ by 20% (GPT-2) and 30% (LLaMA2), and significantly exceeding random neurons (Rnd). This indicates that even during full fine-tuning, LLMs tend to utilize DKNs to learn new knowledge.

5.2 DKNs Guide LLMs to Learn Knowledge

Experimental settings Since LLMs primarily utilize DKNs during fine-tuning, we naturally propose Q2 to consider leveraging them for efficient fine-tuning. Following our previous setup, we first identify DKNs from facts at the 2018 timestamp, then use facts from TempNarrativeLAMA at the 2019 timestamp as fine-tuning data.

For evaluation, we use triple-form queries from the TempLama dataset (2019 timestamp), which enables direct question-answering assessment. We design three evaluation sets: (1) Q_{new}: 2019-facts for assessing new knowledge learning. Notably, the ground truth answers corresponding to the counterfactual data in TempNarrativeLAMA. (2) Q_{old}: 2018-facts for evaluating knowledge preservation. To ensure meaningful evaluation, we first conduct incremental fine-tuning to guarantee the model has mastered these facts. (3) Q_{au}: Paraphrased queries of Q_{new} to verify that models learn knowledge

Method	GPT-2				LLaMA2			
	Q_{new}	Q_{old}	Q_{au}	Average	Q_{new}	Q_{old}	Q_{au}	Average
$\Theta(\mathcal{N})$	54.71	50.08	31.91	45.87	64.71	60.77	51.86	59.07
$\Theta(Rnd)$	13.24	40.92	7.90	21.09	47.98	48.04	41.04	46.19
$\Theta(All)$	**62.91**	18.98	**56.03**	46.25	**69.18**	35.71	**59.92**	55.95
$\boldsymbol{\Theta}(\mathcal{D})$	57.81	**53.19**	51.06	**54.07**	68.21	**62.88**	59.85	**63.75**

Table 2: Model accuracy (%) comparison across different fine-tuning methods. The best and second-best results in each column are marked in bold and underlined, respectively. For significance tests, see Table 7.

rather than just semantic associations.

We propose unfreezing DKNs ($\Theta(\mathcal{D})$) as our method and compare it with three baselines: non-DKNs knowledge neurons[2] ($\Theta(\mathcal{N})$), an equal number of random neurons ($\Theta(Rnd)$), and all neurons ($\Theta(All)$, i.e., full fine-tuning).

Findings (1) Table 2 shows that unfreezing DKNs ($\Theta(\mathcal{D})$) achieves the highest average accuracy for both LLaMA2 (**63.75**% vs. 59.07%) and GPT-2 (**54.07**% vs. 45.87%) compared to the strongest baselines. Beyond average accuracy, our method also consistently achieves either the best or second-best performance across different test sets. These results demonstrate that **DKNs can guide LLMs to learn new knowledge**.

(2) Comparing $\Theta(\mathcal{D})$ with $\Theta(All)$, while $\Theta(All)$ achieves slightly better performance on Q_{new} (e.g., **69.18**% vs. 68.21% in LLaMA2), $\Theta(\mathcal{D})$ maintains substantially better performance on Q_{old} (e.g., **62.88**% vs. 35.71% in LLaMA2). This indicates that while full fine-tuning can learn new knowledge, it tends to encounter catastrophic forgetting, which $\Theta(\mathcal{D})$ effectively mitigates.

(3) The decrease in accuracy on Q_{au} demonstrates that for some facts, LLMs might not have truly learned the knowledge but rather learned superficial semantic associations, a challenge present across all methods.

The statistical significance of these findings is further validated through significance tests (Appendix F.2, Table 7).

6 DKNs relate to LLMs' robustness against input perturbations

Motivation and Dataset Setup In cognitive science, degeneracy is considered to be related to robustness (Whitacre and Bender, 2010; Edelman and Gally, 2001; Whitacre, 2010; Mason, 2015). In practical scenarios, AI chat bots often encounter user input errors (e.g., typos, omissions). Robust LLMs should maintain strong performance despite these perturbations. We thus investigate whether DKNs correlate with LLMs' robustness to such perturbations.

We first use the triple-format TempLama dataset (2018 timestamp) and perform incremental fine-tuning to ensure LLMs master these facts. Then, we simulate input error scenarios by applying random perturbations to the inputs. For a fact with its corresponding query $Q = \{q_1, q_2, \ldots, q_n\}$, we generate its perturbed counterpart. Here, "[replace]" and "[add]" are special characters.

$$Q^* = \begin{cases} \{q_1, \ldots, q_{i-1}, [\text{replace}], q_{i+1}, \ldots, q_n\} & \text{if replace,} \\ \{q_1, \ldots, q_{i-1}, [\text{add}], q_i, \ldots, q_n\} & \text{if add,} \\ \{q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_n\} & \text{if delete.} \end{cases} \tag{16}$$

Experimental Settings To examine the role of DKNs ($\mathcal{D}$), we employ both suppression and enhancement methods. For suppression, we either zero out neuron values or nullify connection weights. For enhancement, we either double neuron values or connection weights.

For evaluation metrics, we adopt two settings: (1) After suppression, we calculate prediction probabilities for both Q and Q^*, and compute the probability decrease: $\Delta Prob(\%) = \frac{Prob(Q) - Prob(Q^*)}{Prob(Q)}$. (2) After enhancement, $\Delta Prob$ becomes a suboptimal metric, as its ideal outcome would be $\Delta Prob = 0$ (i.e., no change in probability when the model is already correct), which limits its sensitivity in measuring improvements. Therefore, we introduce a more informative metric: we first identify cases (Q^*) where LLMs initially give incorrect answers, and then measure the accuracy improvement (ΔAcc) after neuron enhancement. Higher $\Delta Prob$ and ΔAcc indicate stronger correlation between DKNs and LLMs' robustness against input perturbations.

For comparison, we select three baselines: non-DKNs knowledge neurons[3] ($\mathcal{N}$), randomly chosen

[2] If non-DKNs knowledge neurons are fewer than DKNs, we randomly add DKNs to match the size, which actually strengthens this baseline.

[3] Same size-matching strategy as in footnote 2.

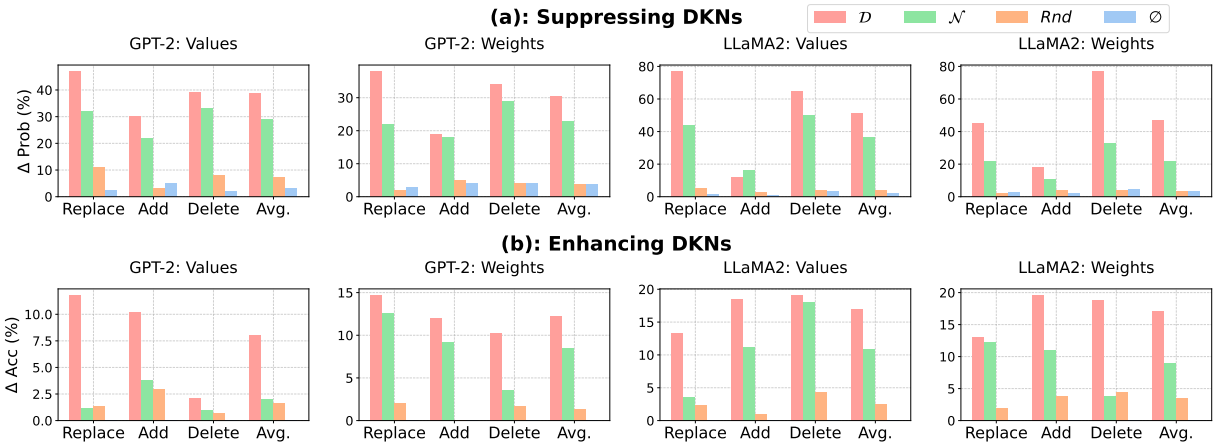

Figure 6: $\Delta Prob$ (a) or ΔAcc (b) of LLMs corresponding to the suppression or enhancement of DKNs and other baselines. For significance tests, see Table 8. "Avg." denotes the average value.

neurons (Rnd, the same number as $\mathcal{D}$), and no operation ($\emptyset$). Since perturbation itself may affect LLMs, we include $\emptyset$ as a baseline to measure LLMs' inherent ability to handle Q^*. Figure 6 presents our results. Additional suppression results using ΔAcc are in Appendix H (Figure 7), corroborating the results in Figure 6.

Findings (1) Figure 6 (a) shows that without manipulation ($\emptyset$), $\Delta Prob$ remains low, indicating that $Prob(Q^*)$ is only slightly lower than $Prob(Q)$. This demonstrates that LLMs can still correctly answer queries despite perturbations. However, after suppressing $\mathcal{D}$, $\Delta Prob$ increases significantly ($\sim$ 30% to 50%). Moreover, this $\Delta Prob$ exceeds that of the strongest baseline ($\mathcal{N}$). This indicates that *suppressing DKNs compromises LLMs' robustness against input perturbation.*

(2) Figure 6 (b) shows that for queries where LLMs initially answer incorrectly ($Acc = 0$), enhancing $\mathcal{D}$ can lead to correct answers ($\Delta Acc > 0$). Compared to baselines, enhancing $\mathcal{D}$ achieves the highest accuracy improvement, with $\Delta Acc \sim 2\times$ that of the strongest baseline ($\mathcal{N}$). This demonstrates that *enhancing DKNs strengthens LLMs' robustness against input perturbation.*

Significance tests further validate these findings, as detailed in Appendix F.3, Table 8. Combining (1) and (2), we can conclude that **DKNs relate to LLMs' robustness against input perturbations**.

7 Related Work

Petroni et al. (2019b) argue that numerous factual knowledge exists within LLMs and suggest using "fill-in-the-blank" cloze tasks to determine if the models have stored specific facts. Meanwhile, Geva et al. (2021) suggest that MLP neurons within transformer models operate as key-value memories. Building on this, Dai et al. (2022) uncover that some MLP neurons are capable of storing factual knowledge, termed as knowledge neurons (KNs). Lundstrom et al. (2022) confirm the reliability of their knowledge localization method, while subsequent knowledge editing experiments by Meng et al. (2022) and Meng et al. (2023) reinforce that MLP neurons indeed store factual knowledge. Moreover, Geva et al. (2023) investigate KN dynamics. Additionally, Chen et al. (2024a) discover multiple distinct KN sets storing identical facts, termed degenerate knowledge neurons (DKNs). Other researchers have also identified various KNs with unique properties (Wang et al., 2022; Tang et al., 2024). It is worth noting that while some research has critiqued the knowledge neuron theory, these studies also acknowledge that, in many cases, KN-based analysis yields meaningful conclusions (Niu et al., 2024; Bricken et al., 2023; Chen et al., 2024b). In summary, despite limitations, the KN-based analysis approach remains valuable for further research.

8 Conclusion

This paper presents a comprehensive analysis of degenerate knowledge neurons in LLMs. First, we provide a comprehensive definition of DKNs that covers both structural and functional aspects, pioneering the study of the internal structures of LLMs. Based on this, we introduce the neuronal topology clustering method for more precise DKN identification. Finally, we explore the applications of DKNs, confirming that DKNs can guide LLMs to learn new knowledge and relate to the robustness of LLMs against input perturbation.

9 Limitations

The primary limitation lies in our simplification of neuronal weight relationships during exploration, as this represents preliminary research. We provide detailed descriptions and potential future work in Appendix D. Additionally, recent works increasingly challenge the knowledge neurons theory (Niu et al., 2024; Bricken et al., 2023; Chen et al., 2024b). While this theory proves valid and applicable under many conditions, it is not perfect and has several limitations. Investigating the degenerate properties of knowledge storage units in LLMs by integrating these new mechanistic interpretability theories represents a promising research direction. A minor limitation stems from the neuron enhancement experiments in Section 6. The requirement to first identify Q^* cases where LLMs initially answer incorrectly results in a relatively small dataset size. While our significance testing partially addresses this concern, creating larger datasets constitutes valuable future work.

References

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. 2020. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *ArXiv preprint*, abs/2309.14402.

Serguei Barannikov. 1994. The framed morse complex and its invariants. *Advances in Soviet Mathematics*, 21:93–116.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Gunnar Carlsson. 2009. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308.

Frédéric Chazal and Bertrand Michel. 2021. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:108.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024a. 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*.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024b. Knowledge localization: Mission not accomplished? enter query localization! *Preprint*, arXiv:2405.14117.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. 2005. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271.

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, Dublin, Ireland. Association for Computational Linguistics.

Tamal K Dey, Dayu Shi, and Yusu Wang. 2019. Simba: An efficient tool for approximating rips-filtration persistence via sim plicial ba tch collapse. *Journal of Experimental Algorithmics (JEA)*, 24:1–16.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Gerald M Edelman and Joseph A Gally. 2001. Degeneracy and complexity in biological systems. *Proceedings of the National Academy of Sciences*, 98(24):13763–13768.

Herbert Edelsbrunner. 2013. Persistent homology: theory and practice.

Herbert Edelsbrunner, John Harer, et al. 2008. Persistent homology-a survey. *Contemporary mathematics*, 453(26):257–282.

Herbert Edelsbrunner and John L Harer. 2022. *Computational topology: an introduction*. American Mathematical Society.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *Preprint*, arXiv:2304.14767.

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, Online and Punta Cana,

Dominican Republic. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *Preprint*, arXiv:2106.07139.

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, Online. Association for Computational Linguistics.

Michael Kerber and Raghvendra Sharathkumar. 2013. Approximate čech complex in low and high dimensions. In *International Symposium on Algorithms and Computation*, pages 666–676. Springer.

Daniel D Lundstrom, Tianjian Huang, and Meisam Razaviyayn. 2022. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pages 14485–14508. PMLR.

Paul H Mason. 2015. Degeneracy: Demystifying and destigmatizing a core concept in systems biology. *Complexity*, 20(3):12–21.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Pre-

ston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. 2017. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019a. 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, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019b. 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, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *Preprint*, arXiv:2402.16438.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alessandro Verri, Claudio Uras, Patrizio Frosini, and Massimo Ferri. 1993. On the use of size functions for shape analysis. *Biological cybernetics*, 70(2):99–107.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding skill neurons in pre-trained transformer-based language models. In *Conference on Empirical Methods in Natural Language Processing*.

James Whitacre and Axel Bender. 2010. Degeneracy: a design principle for achieving robustness and evolvability. *Journal of theoretical biology*, 263(1):143–153.

James M Whitacre. 2010. Degeneracy: a link between evolvability, robustness and complexity in biological systems. *Theoretical Biology and Medical Modelling*, 7:1–17.

Zeyuan Allen Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *ArXiv preprint*, abs/2309.14316.

Afra Zomorodian and Gunnar Carlsson. 2004. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356.

A Experimental Hyperparameters

Our experimental hyperparameters are mainly set based on experience and fine-tuned with reference to experimental results, similar to general experimental papers.

A.1 Hardware spcification and environment.

We ran our experiments on the machine equipped with the following specifications:

- CPU: Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz, Total CPUs: 56
- GPU: NVIDIA GeForce RTX 3090, 24576 MiB (10 units)

- Software:
  - Python Version: 3.10.10
  - PyTorch Version: 2.0.0+cu117

A.2 Experimental Hyperparameters of Neurological Topology Clustering

In Section 4, where we obtain degenerate knowledge neurons, the primary hyperparameters are τ_1 and τ_2. First, τ_1 is a dynamic threshold, set as

$$\tau_1 = 0.5 \times \max\left(R_{\text{persist}}(\mathcal{B}_1), \ldots, R_{\text{persist}}(\mathcal{B}_n)\right) \tag{17}$$

We experimented with different multipliers for τ_1, ranging from 0.3 to 0.7, before settling on 0.5 as it provided the best balance between sensitivity and specificity in identifying degenerate knowledge neurons.

Then, τ_2 is a fixed value.

$$\tau_2 = 0.3 \tag{18}$$

For τ_2, we tested values between 0.1 and 0.5, with 0.05 increments. The value of 0.3 was chosen as it yielded the most consistent results across different datasets.

In this experiment, some data led to excessively large changes in predictive probability, indicating that the LLMs had not originally mastered this factual knowledge, resulting in a very low initial predictive probability. To study the storage mechanism of factual knowledge, it's essential to investigate facts already grasped by the model. Therefore, we set a threshold to exclude data that caused extreme changes in predictive probability. If the change in predictive probability $\Delta Prob$ satisfies:

$$\Delta Prob > 900 \tag{19}$$

then that data is excluded. We experimented with thresholds ranging from 500 to 1500, and found that 900 effectively filtered out outliers without significantly reducing the dataset.

A.3 Experimental Hyperparameters of Knowledge Learning

In Section 5, the experimental hyperparameter $\tau_{\Delta N}$ for the Overlap of DKN and Parameter Changes experiment is set as a dynamic threshold. The process involves calculating the maximum value of $\Delta P(n)$ according to Equation 11. Once this value is determined, $\tau_{\Delta N}$ is set differently based on the model in use. For the GPT-2 model, the threshold $\tau_{\Delta N}$ is calculated as:

$$\tau_{\Delta N} = 0.04 \times \max(\Delta P(n_1), \Delta P(n_2), \ldots, \Delta P(n_k)) \tag{20}$$

In contrast, for the Llama2 model, the calculation of the threshold $\tau_{\Delta N}$ is slightly adjusted:

$$\tau_{\Delta N} = 0.05 \times \max(\Delta P(n_1), \Delta P(n_2), \ldots, \Delta P(n_k)) \tag{21}$$

For both models, we tested multipliers ranging from 0.01 to 0.10, with 0.01 increments. The values of 0.04 for GPT-2 and 0.05 for Llama2 were selected as they provided the most consistent results in identifying significant parameter changes across different datasets and model sizes.

This distinction in the calculation of $\tau_{\Delta N}$ reflects the specific characteristics and performance considerations of each model.

A.4 Experimental Hyperparameters of Disturbance Analysis

In Section 6, the first experiment under Query-Perturbation, namely the Suppressing DKN experiment, similar to A.2, excludes data that satisfies the condition:

$$Prob_{\text{sup}} > 900 \tag{22}$$

We tested threshold values between 700 and 1100, with 900 providing the best balance between data retention and outlier exclusion.

B Details of TempLAMA Dataset

The TempLama dataset (Dhingra et al., 2022) comprises temporal knowledge facts spanning from 2010 to 2020, containing 4,050 unique facts across 9 different relations. The dataset captures temporal evolution of various relationships, including political positions (P39), sports team memberships (P54), employment relationships (P108), and educational backgrounds (P69). Each fact may appear across multiple timestamps, reflecting the temporal persistence of knowledge. For instance, some facts like Tom Brady's team membership span the entire dataset period (2010-2020), while others like David Beckham's team affiliations only appear in specific time windows (2010-2013).

The dataset shows varying temporal characteristics across different years (Table 5). The number of unique queries per year ranges from 2,834 (2010) to 3,348 (2018), with a general trend of increasing coverage in more recent years. Table 3 presents the distribution of relations and illustrative examples

Relation	Description	#Facts	Example Query
P39	Position held	700	Silvio Berlusconi holds the position of *X*. → Prime Minister of Italy
P54	Member of sports team	691	Tom Brady plays for *X*. → New England Patriots
P108	Employer	628	Edward Snowden works for *X*. → Dell Inc.
P286	Head coach	594	*X* is the head coach of San Francisco 49ers. → Chip Kelly
P102	Member of political party	519	Donald Trump is a member of the *X*. → Republican Party
P488	Chairperson	307	*X* is the chair of Conservative Party. → Theresa May
P6	Head of government	287	*X* is the head of the government of Mexico. → Felipe Calderón
P127	Owned by	170	Houston Rockets is owned by *X*. → Leslie Alexander
P69	Educated at	154	Lamar Jackson attended *X*. → Boynton Beach Community High School

Table 3: Statistics and examples of relations in TempLama dataset.

Query Example	Time Span	#Years
Tom Brady plays for X	2010-2020	11
Cristiano Ronaldo plays for X	2010-2020	11
Zlatan Pepemovic plays for X	2010-2020	11
Wayne Rooney plays for X	2010-2020	11
Peyton Manning plays for X	2010-2015	6
Sachin Tendulkar plays for X	2010-2014	5
David Beckham plays for X	2010-2013	4
Ronaldo plays for X	2010-2011	2

Table 4: Query examples with different temporal characteristics.

for each relation type, demonstrating the diverse nature of temporal knowledge captured in our dataset. The examples show how the dataset covers various domains including politics (P39, P6), sports (P54, P286), employment (P108), and education (P69). To further illustrate the temporal nature of our dataset, Table 4 showcases several example queries with their temporal spans, demonstrating how different facts persist over varying time periods, from short-term associations (e.g., Ronaldo's 2-year span) to long-term relationships (e.g., Tom Brady's 11-year span).

C Knowldege Localization

This section introduces the method we use to acquire knowledge neurons. We employ the approach proposed by Chen et al.(2024a), which we will detail below.

Given a query q, we can define the probability of the correct answer predicted by a LLMs as follows:

$$\mathrm{F}(\hat{w}_j^{(l)}) = p(y^*|q, w_j^{(l)} = \hat{w}_j^{(l)}) \tag{23}$$

Here, y^* represents the correct answer, $w_j^{(l)}$ denotes the j-th neuron in the l-th layer, and $\hat{w}_j^{(l)}$ is the specific value assigned to $w_j^{(l)}$. To calculate the attribution score for each neuron, we employ the technique of integrated gradients.

To compute the attribution score of a neuron $w_j^{(l)}$, we consider the following formulation:

$$\mathrm{Attr}(w_j^{(l)}) = (\overline{w}_j^{(l)} - w'^{(l)}_j) \int_0^1 \frac{\partial \mathrm{F}(w'^{(l)}_j + \alpha(\overline{w}_j^{(l)} - w'^{(l)}_j))}{\partial w_j^{(l)}} d\alpha \tag{24}$$

Year	#Queries	Year	#Queries
2020	3,216	2015	3,208
2019	3,334	2014	3,197
2018	3,348	2013	3,188
2017	3,284	2012	3,083
2016	3,249	2011	2,967
		2010	2,834

Table 5: Temporal distribution of facts. As mentioned in Section 5, we select 2018 and 2019 as timestamps since they contain the largest number of facts. Since 2018 has slightly more facts than 2019, we simply exclude the facts without 2019 timestamps.

Here, $\overline{w}_j^{(l)}$ represents the actual value of $w_j^{(l)}$, $w_j'^{(l)}$ serves as the baseline vector for $w_j^{(l)}$. The term $\frac{\partial \mathrm{F}(w_j'^{(l)}+\alpha(w_j^{(l)}-w_j'^{(l)}))}{\partial w_j^{(l)}}$ computes the gradient with respect to $w_j^{(l)}$.

Next, we aim to obtain $w'^{(l)}_j$. Starting from the sentence q, we acquire a baseline sentence and then encode this sentence as a vector.

Let the baseline sentence corresponding to q_i be q_i', and q_i' consists of m words, maintaining a length consistent with q, denoted as $q_i' = (q_{i1}' \dots q_{ik}' \dots q_{im}')$. Since we are using auto-regressive models, according to Chen et al.(2024a)' method, $q_{ik}' = \langle \text{eos} \rangle$, where $\langle \text{eos} \rangle$ represents "end of sequence" in auto-regressive models.

The attribution score $Attr_i(w_j^{(l)})$ for each neuron, given the input q_i, can be determined using Equation (24). For the computation of the integral, the Riemann approximation method is employed:

$$Attr_i(w_j^l) \approx \frac{\overline{w}_j^{(l)}}{N} \sum_{k=1}^{N} \frac{\partial F(w'^{(l)}_j + \frac{k}{N} \times (\overline{w}_j^{(l)} - w'^{(l)}_j))}{\partial w_j^{(l)}} \tag{25}$$

where N is the number of approximation steps.

Then, the attribution scores for each word q_i are aggregated and subsequently normalized:

$$Attr(w_j^l) = \frac{\sum_{i=1}^{m} Attr_i(w_j^l)}{\sum_{j=1}^{n} \sum_{i=1}^{m} Attr_i(w_j^l)}, \tag{26}$$

Let $\mathcal{N}$ be the set of neurons classified as knowledge neurons based on their attribution scores exceeding a predetermined threshold τ, for a given input q. This can be formally defined as:

$$\mathcal{N} = \left\{ w_j^{(l)} \,\middle|\, Attr(w_j^{(l)}) > \tau \right\} \tag{27}$$

where l encompassing all layers and j including all neurons within each layer.

D Simplifications and Future Work

Our approach to analyzing the structural properties of Degenerate Knowledge Neurons (DKNs) employs several simplifications to make the problem tractable. Here, we detail these simplifications and discuss potential avenues for future research.

Choice of Distance Metric We define the distance between neurons as the inverse of the weight connecting them. This simplification allows for an intuitive interpretation where stronger connections (higher weights) result in shorter distances. However, this approach has limitations:

- It doesn't account for the sign of the weight, which could be significant in neural information processing.
- It assumes a linear relationship between weight and distance, which may not always hold true.

Future work could explore alternative distance metrics, such as:

- Incorporating both magnitude and sign of weights.
- Using non-linear transformations of weights to better reflect neural dynamics.
- Developing context-dependent distance metrics that consider the activation patterns of neurons.

Distance Aggregation Method For neurons separated by multiple layers, we calculate the total distance by summing the individual distances along the path. This additive approach simplifies calculations but may not fully capture the complexities of information flow in neural networks. Alternative methods to consider in future research include:

- Multiplicative aggregation, which could better represent the compounding effects of multiple connections.
- Non-linear aggregation functions that account for potential synergistic or antagonistic effects between layers.
- Weighted aggregation methods that consider the relative importance of different paths or layers.

Exclusion of Non-Knowledge Neurons Our current model focuses solely on DKNs, excluding non-knowledge neurons from the analysis. While this simplifies the model, it may overlook important interactions. Future enhancements could:

- Incorporate non-knowledge neurons with different weighting schemes.
- Develop a hierarchical model that considers interactions between knowledge and non-knowledge neurons at different scales.
- Investigate the role of non-knowledge neurons in facilitating or modulating information flow between DKNs.

Unidirectional Information Flow We assume that information in LLMs flows only between layers and not within them. This simplification aligns with current understanding but may not capture all aspects of neural network dynamics. Future research could:

- Explore potential intra-layer interactions and their impact on knowledge representation.
- Investigate feedback mechanisms that might allow information to flow backwards through the network.

Static Network Analysis Our current approach analyzes the network structure statically. However, neural networks are dynamic systems. Future work might:

- Develop time-dependent models that capture how DKN structures evolve during training or inference.
- Investigate how different input patterns activate and modulate DKN structures.

Scalability Considerations The current method may face computational challenges with very large networks. Future research could focus on:

- Developing more efficient algorithms for distance calculation in large-scale networks.
- Exploring sampling or approximation techniques for analyzing subsets of the network.
- Leveraging graph theory and network science techniques for analyzing DKN structures at scale.

Validation and Empirical Testing While our model provides a theoretical framework, extensive empirical validation is needed. Future work should:

- Conduct comprehensive experiments across various LLM architectures and tasks.
- Correlate structural properties of DKNs with measurable performance metrics.
- Develop benchmarks specifically designed to test the predictive power of DKN structural analysis.

By addressing these simplifications and exploring these future directions, researchers can build upon our foundational work to develop more sophisticated and accurate models of knowledge representation in large language models. This could lead to improved understanding of how LLMs store and process information, potentially informing the development of more efficient and interpretable AI systems.

E Topology Data Analysis

Persistent homology is a method for computing topological features of a space at different spatial resolutions. More persistent features are detected over a wide range of spatial scales and are deemed more likely to represent true features of the underlying space rather than artifacts of sampling, noise, or particular choice of parameters (Carlsson, 2009).

To find the persistent homology of a space, the space must first be represented as a simplicial complex. A distance function on the underlying space corresponds to a filtration of the simplicial complex, that is a nested sequence of increasing subsets. One common method of doing this is via taking the sublevel filtration of the distance to a point cloud, or equivalently, the offset filtration on the point cloud and taking its nerve in order to get the simplicial filtration known as Čech filtration (Kerber and Sharathkumar, 2013). A similar construction uses a nested sequence of Vietoris–Rips complexes known as the Vietoris–Rips filtration (Dey et al., 2019).

E.1 Definition

In persistent homology, formally, we consider a real-valued function defined on a simplicial complex, denoted as $f : K \to \mathbb{R}$. This function is required to be non-decreasing on increasing sequences of faces, meaning that for any two faces σ and τ in K, if σ is a face of τ, then $f(\sigma) \leq f(\tau)$.

For every real number a, the sublevel set $K_a = f^{-1}((-\infty, a])$ forms a subcomplex of K. The values of f on the simplices in K create an ordering of these sublevel complexes, which leads to a filtration:

$$\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K \tag{28}$$

Within this filtration, for $0 \leq i \leq j \leq n$, the inclusion $K_i \hookrightarrow K_j$ induces a homomorphism on the simplicial homology groups for each dimension p, noted as $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$. The p^{th} persistent homology groups are the images of these homomorphisms, and the p^{th} persistent Betti numbers $\beta_p^{i,j}$ are defined as the ranks of these groups (Edelsbrunner and Harer, 2022). Persistent Betti numbers for $p = 0$ coincide with the size function, an earlier concept related to persistent homology (Verri et al., 1993).

The concept extends further to any filtered complex over a field F. Such a complex can be transformed into its canonical form, which is a direct sum of filtered complexes of two types: one-dimensional complexes with trivial differential (expressed as $d(e_{t_i}) = 0$) and two-dimensional complexes with trivial homology (expressed as $d(e_{s_j+r_j}) = e_{r_j}$) (Barannikov, 1994).

A persistence module over a partially ordered set P consists of a collection of vector spaces U_t, indexed by P, along with linear maps $u_t^s : U_s \to U_t$ for $s \leq t$. This module can be viewed as a functor from P to the category of vector spaces or R-modules. Persistence modules over a field F indexed by $\mathbb{N}$ can be expressed as:

$$U \simeq \bigoplus_i x^{t_i} \cdot F[x] \oplus \left(\bigoplus_j x^{r_j} \cdot (F[x]/(x^{s_j} \cdot F[x])) \right) \tag{29}$$

Here, multiplication by x represents a forward step in the persistence module. The free parts correspond to homology generators that appear at a certain filtration level and persist indefinitely, whereas torsion parts correspond to those that appear at a filtration level and last for a finite number of steps (Barannikov, 1994; Zomorodian and Carlsson, 2004).

This framework allows the unique representation of the persistent homology of a filtered simplicial complex using either a persistence barcode or a persistence diagram. In the barcode, each persistent generator is represented by a line segment starting and ending at specific filtration levels, while in the diagram, each generator is represented as a point with coordinates indicating its birth and death times. Barannikov's canonical form offers an equivalent representation.

E.2 Stability

The stability of persistent homology is a key attribute, particularly in its application to data analysis, as it ensures robustness against small perturbations or noise in the data (Cohen-Steiner et al., 2005). This stability is quantitatively defined in terms of the **bottleneck distance**, a metric for comparing persistence diagrams.

The bottleneck distance between two persistence diagrams X and Y is defined as:

$$W_\infty(X, Y) := \inf_{\varphi: X \to Y} \sup_{x \in X} \|x - \varphi(x)\|_\infty \tag{30}$$

where the infimum is taken over all bijections φ from X to Y. This metric essentially measures the greatest distance between matched points (or generators) in two persistence diagrams, considering the optimal matching.

A fundamental result in the theory of persistent homology is that small changes in the input data (such as a filtration of a space) result in small changes in the corresponding persistence diagram, as measured by the bottleneck distance. This is formalized by considering a space X, homeomorphic to a simplicial complex, with a filtration determined by the sublevel sets of a continuous tame function $f : X \to \mathbb{R}$. The map D that takes the function f to the persistence diagram of its kth homology is 1-Lipschitz with respect to the supremum norm on functions and the bottleneck distance on persistence diagrams. Formally, this is expressed as (Cohen-Steiner et al., 2005):

$$W_\infty(D(f), D(g)) \leq \|f - g\|_\infty \tag{31}$$

This Lipschitz condition implies that a small change in the function f, as measured by the supremum norm, will not cause a disproportionately large change in the persistence diagram. Consequently, persistent homology is particularly useful in applications where data may be subject to noise or small variations, as the essential topological features (captured by the persistence diagrams) are not overly sensitive to such perturbations.

E.3 Computation

There are various software packages for computing persistence intervals of a finite filtration (Otter

et al., 2017). The principal algorithm is based on the bringing of the filtered complex to its canonical form by upper-triangular matrices (Barannikov, 1994).

F Results of Statistical Significance Tests

To establish the robustness of our findings, we conduct comprehensive statistical analyses across our three main experimental directions: DKN acquisition, knowledge learning, and disturbance analysis. For each experiment, we employ independent t-tests to assess statistical significance, complemented by Cohen's d measurements to quantify effect sizes. The t-tests evaluate whether the observed differences between methods are statistically significant, while Cohen's d provides a standardized measure of the practical significance of these differences. As shown in Tables 6, 7 and 8, across all experiments, we consistently observe statistically significant results ($p < 10^{-5}$) with moderate to large effect sizes (Cohen's d ranging from 0.13 to 2.15), strongly supporting the effectiveness of our proposed methods. The following subsections present detailed analyses for each experimental direction.

F.1 Significance Tests for Subsection 4.2

Our dataset contains 34,963 facts, each corresponding to a Generalized Curvature Ratio (GCR) value. While Table 1 presents their mean values, we conduct rigorous statistical analyses on the raw data to establish the significance of our findings (Table 6). We perform independent t-tests comparing NTC with each baseline method, complemented by Cohen's d effect size measurements to quantify the practical significance of these differences. The large t-statistics and extremely small p-values ($< 10^{-5}$) across all comparisons indicate that the performance differences are highly significant. Furthermore, the substantial Cohen's d values (ranging from 0.13 to 1.63) suggest moderate to large practical effects. These results strongly support the superior performance of our NTC method over the baseline approaches.

F.2 Significance Tests for Subsection 5.1

The statistical analyses in Table 7 reveal several key findings. For GPT-2, comparing $\Theta(\mathcal{D})$ vs. $\Theta(\mathcal{N})$ shows significant improvements across all query types ($p < 10^{-5}$), with particularly strong effects for Q_{au} (Cohen's d = 0.79). The comparison between $\Theta(\mathcal{D})$ vs. $\Theta(Rnd)$ demonstrates substantial improvements ($p < 10^{-5}$) with large effect sizes for Q_{new} (d = 2.01) and Q_{au} (d = 2.14). Meanwhile, $\Theta(\mathcal{D})$ vs. $\Theta(All)$ shows mixed results with negative effects for Q_{new} and Q_{au} but strong positive effects for Q_{old} (d = 1.57).

For LLaMA2-7b, $\Theta(\mathcal{D})$ vs. $\Theta(\mathcal{N})$ exhibits significant improvements across all query types ($p < 10^{-5}$). The comparison between $\Theta(\mathcal{D})$ vs. $\Theta(Rnd)$ shows consistent strong improvements with moderate to large effect sizes (d ranging from 0.51 to 0.70). Furthermore, $\Theta(\mathcal{D})$ vs. $\Theta(All)$ demonstrates mixed results with particularly strong positive effects for Q_{old} (d = 1.02).

F.3 Significance Tests for Subsection 5.2

The statistical analyses in Table 8 reveal significant findings for both suppression and enhancement experiments.

(1) Suppressing DKNs: The comparison between $\mathcal{D}$ and $\mathcal{N}$ shows significant effects across all settings ($p < 10^{-5}$), with effect sizes ranging from moderate (d = 0.46) to large (d = 1.11). The comparison between $\mathcal{D}$ and Rnd demonstrates even larger effects (d ranging from 1.92 to 2.15), indicating the specificity of DKN impact.

(2) Enhancing DKNs: We observe significant improvements across all settings ($p < 10^{-5}$), with particularly large effects in GPT-2 Values (d = 1.73) and consistently strong effects across other settings. Comparisons with random neurons show larger effect sizes (d ranging from 1.83 to 2.11), supporting the targeted nature of DKN enhancement.

G Construction of TempNarrativeLAMA Dataset

TempNarrativeLAMA is our newly constructed dataset for evaluating LLMs' temporal knowledge learning capabilities. We derive it from TempLAMA by converting structured knowledge into natural language narratives while introducing counterfactual instances. The dataset spans 9 relations with 10,693 facts, each associated with 2-6 timestamps. Consistent with the original TempLAMA, answers may either remain constant or vary across different timestamps.

To construct this dataset, we first create counterfactual instances by replacing correct answers in TempLAMA with incorrect ones sampled from the same relation type, maintaining temporal and semantic coherence. To better simulate real-world fine-tuning scenarios where models learn from nat-

Method	GPT-2			LLaMA2-7b		
	t-statistic	p-value	Cohen's d	t-statistic	p-value	Cohen's d
NTC vs. DBSCAN	151.85	$< 10^{-5}$	1.29	110.99	$< 10^{-5}$	0.94
NTC vs. Hierarchical	131.91	$< 10^{-5}$	1.12	14.84	$< 10^{-5}$	0.13
NTC vs. K-Means	149.65	$< 10^{-5}$	1.27	85.37	$< 10^{-5}$	0.73
NTC vs. AMIG	192.06	$< 10^{-5}$	1.63	149.02	$< 10^{-5}$	1.27

Table 6: Statistical significance results comparing NTC with other methods.

Method	GPT-2								
	$\mathbf{Q_{new}}$			$\mathbf{Q_{old}}$			$\mathbf{Q_{au}}$		
	t-stat	p-value	Cod	t-stat	p-value	Cod	t-stat	p-value	Cod
$\Theta(\mathcal{D})$ vs. $\Theta(\mathcal{N})$	6.58	$< 10^{-5}$	0.09	7.01	$< 10^{-5}$	0.10	58.05	$< 10^{-5}$	0.79
$\Theta(\mathcal{D})$ vs. $\Theta(Rnd)$	147.18	$< 10^{-5}$	2.01	33.26	$< 10^{-5}$	0.45	156.46	$< 10^{-5}$	2.14
$\Theta(\mathcal{D})$ vs. $\Theta(All)$	-11.80	$< 10^{-5}$	-0.16	114.91	$< 10^{-5}$	1.57	-12.65	$< 10^{-5}$	-0.17
Method	**LLaMA2-7b**								
	$\mathbf{Q_{new}}$			$\mathbf{Q_{old}}$			$\mathbf{Q_{au}}$		
	t-stat	p-value	Cod	t-stat	p-value	Cod	t-stat	p-value	Cod
$\Theta(\mathcal{D})$ vs. $\Theta(\mathcal{N})$	7.00	$< 10^{-5}$	0.10	4.81	$< 10^{-5}$	0.07	19.50	$< 10^{-5}$	0.27
$\Theta(\mathcal{D})$ vs. $\Theta(Rnd)$	48.07	$< 10^{-5}$	0.66	37.60	$< 10^{-5}$	0.51	51.22	$< 10^{-5}$	0.70
$\Theta(\mathcal{D})$ vs. $\Theta(All)$	-2.12	0.034	-0.03	74.31	$< 10^{-5}$	1.02	-0.03	0.978	-0.00

Table 7: Statistical significance results comparing $\mathbf{\Theta}(\mathcal{D})$ with other methods across different query types. t-stat: t-statistic from independent t-test; p-value: statistical significance level; Cod: Cohen's d effect size.

Setting	$\mathcal{D}$ vs. $\mathcal{N}$			$\mathcal{D}$ vs. Rnd		
	t-statistic	p-value	Cohen's d	t-statistic	p-value	Cohen's d
	Suppressing DKNs					
GPT-2 Values	19.39	$< 10^{-5}$	0.47	78.55	$< 10^{-5}$	1.92
GPT-2 Weights	18.64	$< 10^{-5}$	0.46	84.79	$< 10^{-5}$	2.07
LLaMA2 Values	22.57	$< 10^{-5}$	0.55	87.62	$< 10^{-5}$	2.14
LLaMA2 Weights	45.21	$< 10^{-5}$	1.11	87.95	$< 10^{-5}$	2.15
	Enhancing DKNs					
GPT-2 Values	22.01	$< 10^{-5}$	1.31	23.16	$< 10^{-5}$	1.38
GPT-2 Weights	9.01	$< 10^{-5}$	0.54	26.30	$< 10^{-5}$	1.57
LLaMA2 Values	3.95	$< 10^{-3}$	0.60	9.43	$< 10^{-5}$	1.43
LLaMA2 Weights	5.21	$< 10^{-5}$	0.79	8.78	$< 10^{-5}$	1.33

Table 8: Statistical significance results comparing $\mathcal{D}$ with $\mathcal{N}$ and Rnd methods across different settings. empty set, not used in comparison.

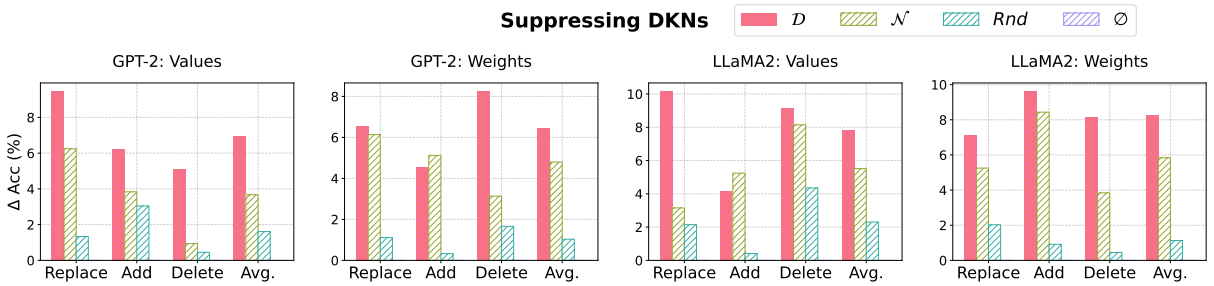

Figure 7: Accuracy changes after suppressing different types of neurons (DKNs ($\mathcal{D}$), knowledge neurons ($\mathcal{N}$), and random neurons (Rnd)) in GPT-2 and LLaMA2. Results are shown for both neuron values and connection weights.

ural language rather than structured triples, we then convert each instance into free-form text using carefully designed templates.

The templates are designed to provide rich contextual information while maintaining consistency across different relation types. Table 9 presents the complete set of templates used for this conversion. Each template incorporates three key elements: temporal context (specified by date), the subject entity, and the target information (either correct or incorrect answer). The templates reflect the formal style of encyclopedic or news articles, with sufficient length and detail to provide meaningful context for learning.

For counterfactual answer selection, we implement a controlled sampling strategy where incorrect answers are randomly selected from the pool of all possible answers within the same relation type, excluding the correct answer. This ensures that the incorrect information maintains semantic validity while being factually wrong. For example, when dealing with political party membership (P102), an incorrect answer would be another political party rather than an arbitrary entity.

H Additional Suppression Results

To complement our main results using $\Delta Prob$, we present additional suppression experiments using accuracy changes (ΔAcc) as the evaluation metric. While our enhancement experiments measure accuracy improvements on initially incorrect predictions (Q^*), these suppression experiments focus on accuracy degradation for initially correct predictions (also for Q^*). This provides a different perspective on the importance of identified neurons for model robustness.

Specifically, we: (1) Identify cases where the model initially predicts correctly; (2) Apply suppression to different neuron sets ($\mathcal{D}$, $\mathcal{N}$, and Rnd); (3) Calculate the decrease in accuracy: $\Delta Acc = Acc_{after} - Acc_{before}$.

As shown in Figure 7, DKNs ($\mathcal{D}$) consistently lead to larger accuracy drops compared to knowledge neurons ($\mathcal{N}$) and random neurons (Rnd) across both models and both suppression methods (values and weights). This aligns with our $\Delta Prob$ findings in the main text, further supporting our conclusion that DKNs play a crucial role in maintaining model robustness against interference. The more substantial accuracy degradation when suppressing DKNs indicates that these neurons are particularly important for preserving the model's correct predictions in the presence of potential interference.

Relation	Template
P39 (Position held)	Official government and institutional records from {date} document that {subject} holds the position of {answer}. This role involves significant responsibilities in policy-making, leadership, and institutional governance. Their appointment to this position reflects their expertise and experience in the relevant field, as well as their commitment to public service.
P54 (Plays for team)	Sports records and team rosters from {date} confirm that {subject} is an active player for {answer}. Their role within the team encompasses both competitive performance and contribution to team dynamics. This professional affiliation represents a significant phase in their athletic career and the team's competitive strategy.
P108 (Employer)	Based on professional records and organizational documentation from {date}, {subject} holds a position at {answer}. Their professional role involves significant contributions to the organization's objectives and ongoing projects. This appointment demonstrates the organization's commitment to bringing in experienced professionals to strengthen its capabilities and advance its mission.
P286 (Head coach)	Sports management records from {date} confirm that {answer} serves as the head coach for {subject}. In this role, they are responsible for team strategy, player development, and overall performance improvement. Their coaching philosophy and leadership approach have become integral to the team's competitive strategy and organizational culture.
P102 (Member of)	According to recent political developments and official party records from {date}, {subject} is an active member of {answer}. Their involvement in the party includes participating in policy discussions, representing party interests in various forums, and contributing to the party's legislative agenda. This membership reflects their commitment to the party's core values and political platform.
P488 (Chairperson)	Organizational documents and board records from {date} establish that {answer} serves as the chairperson of {subject}. In this leadership capacity, they oversee strategic planning, governance, and major organizational initiatives. Their appointment to this position brings valuable experience and vision to guide the organization's development and future direction.
P6 (Head of government)	Official government records and administrative documentation from {date} confirm that {answer} serves as the head of government for {subject}. In this executive leadership role, they are responsible for policy implementation, administrative oversight, and strategic governance. Their administration has focused on addressing key challenges and implementing initiatives for regional development.
P127 (Owned by)	According to corporate ownership records and financial documentation dated {date}, {subject} operates under the ownership of {answer}. This ownership structure influences the strategic direction and operational decisions of the entity. The acquisition represents a significant component of the owner's portfolio and reflects their long-term investment strategy in this sector.
P69 (Educated at)	Academic records and institutional documentation from {date} indicate that {subject} pursued their education at {answer}. Their academic journey at this institution has contributed significantly to their professional development and expertise in their field. This educational background represents an important foundation for their subsequent career achievements and professional contributions.

Table 9: Templates used for converting structured knowledge into natural language text. Each template is designed to provide rich context while maintaining a consistent style across different relation types.