The Achilles' Heel of LLMs: How Altering a Handful of Neurons Can Cripple Language Abilities

Zixuan Qin[1] Qingchen Yu[2,3] Kunlin Lyu[1] Zhaoxin Fan[2,3*]
Yifan Sun[1*]

[1]School of Statistics, Renmin University of China
[2]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing
[3]School of Artificial Intelligence, Beihang University


Background & Motivation

From Human Brains to Large Language Models (LLMs)

- **Neuroscience insight:** Human cognitive functions heavily rely on a small subset of "bottleneck neurons"; damage to these causes immediate capability loss.
- **The fundamental question:** Despite their massive, distributed parameter scale (tens of billions), do LLMs also possess a similarly sparse set of *critical artificial neurons* indispensable for their capabilities?
- **Previous gap:** Most research focuses on weight/activation outliers, lacking neuron-level causal verification of capability collapse.

Methodology: Perturbation-based Causal Identification

Stage 1: Neuron Importance Evaluation

Inject controlled Gaussian noise into the input embedding. Generating a ranked list of "potential critical neurons."

Stage 2: Critical Neuron Identification (Greedy Causal Verification)

Sequentially mask the top-ranked neurons. Monitor for a catastrophic surge in perplexity.

Algorithm 1 Stage 1: Neuron Importance Evaluation

Require: Model f_θ, input text p, noise scale α, samples K

$\mathbf{x} \leftarrow \text{Context}(p)$
$A^{\text{clean}} \leftarrow f_\theta(\mathbf{x})$
for $i = 1$ **to** K **do**
  $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x} + \alpha \cdot \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $A_i^{\text{noisy}} \leftarrow f_\theta(\tilde{\mathbf{x}}_i)$
end for
for each neuron s **do**
  $\text{Imp}(s) \leftarrow \frac{1}{K} \sum_{i=1}^{K} |A_s^{\text{clean}} - A_{i,s}^{\text{noisy}}|$
end for
$S \leftarrow \text{SortByDescending}(\{s_1, s_2, \ldots, s_N\}, \text{Imp})$
return Importance-ranked neuron list S

Algorithm 2 Stage 2: Critical Neuron Identification

Require: Model f_θ, sorted neurons S, input text p, threshold ϵ

$\text{PPL}_{\text{original}} \leftarrow \text{Perplexity}(f_\theta, p)$
$M \leftarrow \emptyset$
for $n = \Delta n, 2\Delta n, \ldots, |S|$ **do**
  $M \leftarrow \text{Top-}n(S)$
  Apply masking: $\tilde{n}_l^{(i)}(\mathbf{x}) = 0$ for all $(l, i) \in M$
  $\text{PPL}_{\text{masked}} \leftarrow \text{Perplexity}(f_\theta^{-M}, p)$
  $\Delta = \log_{10}(\text{PPL}_{\text{masked}}) - \log_{10}(\text{PPL}_{\text{original}})$
  if $\Delta \geq \epsilon$ **then**
    break
  end if
end for
return Critical neuron set M

Finding 1: Ultra-Sparse Vulnerability

Catastrophic Collapse: Masking merely **3–10 neurons** ($\sim 10^{-8}$ of total parameters) completely destroys language ability.

Model Family	Model	Critical Neurons Number	Critical Neurons Rate (10^{-8})	WikiText-103 Original → Masked	C4 Original → Masked
Llama-3	Llama-3.2-1B-Instruct	4	6.43	17.74 → **1.58×10^6**	21.15 → **1.41×10^6**
	Llama-3.2-3B-Instruct	4	3.19	13.36 → **3.79×10^5**	16.47 → **4.69×10^5**
	Llama-3-8B-Instruct	5	2.21	10.88 → **1.48×10^6**	14.04 → **1.01×10^6**
	Llama-3.3-70B-Instruct	7	0.63	5.41 → **3.86×10^6**	8.65 → **2.18×10^6**
Gemma	Gemma-2B	9	7.79	12.58 → **2.36×10^16**	13.59 → **6.66×10^15**
	Gemma-7B	3	1.01	9.98 → **6.25×10^21**	11.41 → **1.32×10^20**
DeepSeek-R1	Deepseek-R1-Distill-Qwen-1.5B	23	21.12	61.05 → **1.28×10^3**	49.09 → **1.19×10^3**
	DeepSeek-R1-Distill-Qwen-7B	3	1.28	34.57 → **5.59×10^3**	35.62 → **3.70×10^3**
	DeepSeek-R1-Distill-Llama-8B	3	1.33	17.56 → **2.81×10^5**	23.43 → **2.55×10^5**
	DeepSeek-R1-Distill-Qwen-14B	4	1.12	12.85 → **6.85×10^3**	19.26 → **5.00×10^3**
	DeepSeek-R1-Distill-Qwen-32B	28	3.58	9.36 → **8.07×10^2**	15.14 → **7.40×10^2**
	DeepSeek-R1-Distill-Llama-70B	3	0.27	7.86 → **2.21×10^4**	11.34 → **1.84×10^4**
Phi-3	Phi-3-mini-4k-Instruct	13	6.83	8.24 → **9.51×10^4**	10.57 → **4.69×10^4**
	Phi-3.5-mini-Instruct	13	6.83	8.46 → **2.67×10^6**	10.71 → **1.02×10^6**
Qwen2.5	Qwen2.5-0.5B-Instruct	3	5.90	18.18 → **1.76×10^6**	22.34 → **1.08×10^6**
	Qwen2.5-1.5B-Instruct	11	10.19	12.37 → **4.01×10^2**	15.95 → **3.28×10^2**
	Qwen2.5-3B-Instruct	5	2.89	11.24 → **3.34×10^3**	14.28 → **2.52×10^3**
	Qwen2.5-7B-Instruct	10	4.30	9.75 → **9.64×10^4**	13.01 → **7.61×10^4**
	Qwen2.5-14B-Instruct	4	1.13	7.69 → **2.39×10^3**	11.12 → **1.59×10^3**
	Qwen2.5-32B-Instruct	45	5.80	7.40 → **1.01×10^2**	10.98 → **0.94×10^2**
	Qwen2.5-72B-Instruct	3	0.26	10.23 → **2.24×10^4**	10.72 → **1.40×10^4**

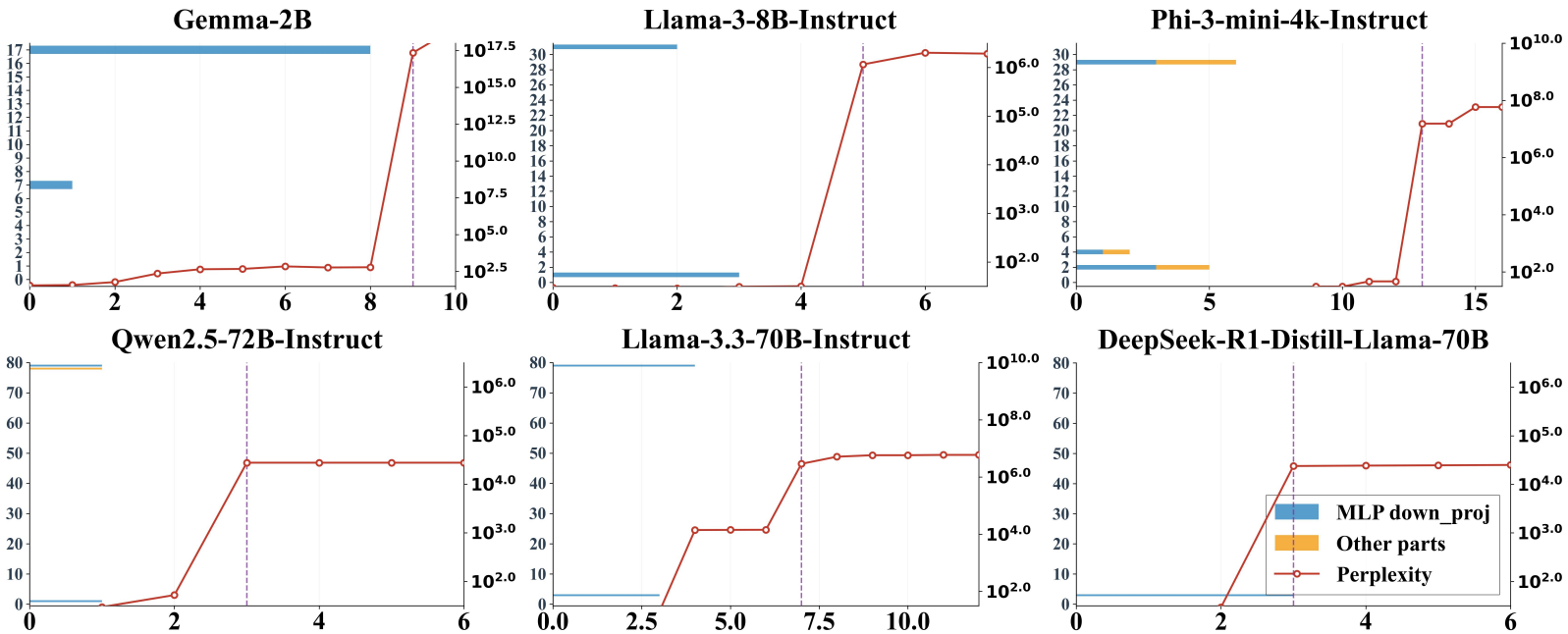
Finding 2: Architectural Concentration

Where are the critical neurons located?

- **Outer Layers:** Critical neurons are not uniformly distributed but exhibit a strong tendency to cluster in the outer layers.
- **Component Specificity:** They are almost exclusively located within the **MLP** `down_proj` components.
- **Interpretation:** This aligns with the "Super Weight" phenomenon. Outer `down_proj` layers act as extreme information bottlenecks, compressing high-dimensional representations back into the embedding space before final output generation.

Finding 3: Phase Transition Behavior

- **Threshold Effect:** Masking partial subsets yields minimal impact; masking the *complete* set triggers a sudden, catastrophic perplexity surge.
- **Collective Circuitry:** These neurons operate as a highly interdependent computational unit rather than independent switches.



Impact on Downstream Tasks

Universal Capability Zeroing

We evaluated 70B-class models across 7 diverse benchmarks (MMLU-Pro, HumanEval, MATH, GPQA Diamond, IFEval, MGSM, SimpleQA).

- **Result:** After masking the identified critical neurons, accuracy/success rates dropped to **exactly 0.0000** across **all** tasks.
- This confirms that these sparse neurons control fundamental language processing capabilities, not just isolated or task-specific knowledge.

Model	Condition	MMLU-Pro	IFEval	GPQA-Diamond	HumanEval	MATH	MGSM	SimpleQA
Llama-3.3-70B-Instruct	Original	0.4441	0.4658	0.2879	0.2988	0.4420	0.9590	0.3704
	Masked	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DeepSeek-R1-Distill-Llama-70B	Original	0.1631	0.2458	0.1869	0.1951	0.1420	0.9809	0.3104
	Masked	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Qwen2.5-72B-Instruct	Original	0.2442	0.5268	0.3687	0.2866	0.3140	0.9080	0.2443
	Masked	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Conclusion & Implications

- **Core Takeaway:** LLM capabilities are not as robustly distributed as assumed; they rely on an "Achilles' Heel" of ultra-sparse critical neurons.
- **Security Risks:** This extreme sparsity presents significant vulnerabilities, as minute, targeted interventions can trigger complete system failure.
- **Future Architecture:** Highlights the urgent need to rethink Transformer designs to enforce more uniform, redundant, and robust computational distribution across the network.

Code available at: github.com/qqqqqqqzx/The-Achilles-Heel-of-LLMs