Unveiling Linguistic and Mathematical Knowledge: Interpreting Grammar and Arithmetic Embeddings in Small-Scale LLMs

Pratim Chowdhary*
Department of Computer Science
Dartmouth College
cpratim.25@dartmouth.edu

Peter Chin
Department of Engineering
Thayer School of Engineering
pc@dartmouth.edu

Deepernab Chakrabarty
Department of Computer Science
Dartmouth College
deepernab@dartmouth.edu

Abstract

Small transformer language models ($\leq$10 B parameters) already solve a surprising range of grammatical and numerical tasks. But *which* internal components drive each capability—and how much circuitry is reused across domains—remains unclear. We study three task families—synthetic arithmetic verification, arithmetic word-problems, and grammatical acceptability—and trace responsibility down to the level of individual attention heads. Using a causal ablation-and-pruning procedure that extracts a *Minimum-Sufficient Head Circuit* (MSHC) for each task, we show that only 10-20 heads (0.4 % of parameters) are needed to recover 90 % of full-model accuracy in the models we analyse (Gemma-9B, Llama-8B, Qwen-8B). The MSHCs for arithmetic verification and word-problems overlap by 40-60 %, revealing a reusable numerical sub-network, whereas grammar circuits are largely disjoint. These findings suggest that small LLMs learn a dedicated "number sense" circuit that generalises from bare arithmetic to text-framed problems, while syntactic competence is carried by a separate set of heads. Our results offer new levers for parameter-efficient fine-tuning and mechanistic interpretability of compact language models.

1 Introduction

Large language models (LLMs) have transformed natural language processing and mathematical reasoning, demonstrating unprecedented capabilities across a spectrum of tasks—from simple question answering to complex linguistic analysis, arithmetic problem solving, and formal reasoning [Brown et al., 2020, Chowdhery et al., 2022, Touvron et al., 2023b, Jiang et al., 2023]. These models, trained through self-supervised learning on vast corpora of text and mathematical data, have increasingly approached human-like performance in generating both grammatically well-formed text and mathematically coherent solutions, despite having no explicit grammatical rules or arithmetic algorithms programmed into their architecture. This emergent dual competence—arising purely from statistical patterns in training data—represents a fascinating case study in how both linguistic and mathematical knowledge can be acquired implicitly through exposure rather than explicit instruction.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Submitted to the Mathematics of Modern Machine Learning Workshop at NeurIPS 2024. Do not distribute.

In this work we move beyond aggregate performance metrics and zoom in on the level of individual attention heads. Inspired by circuit-based interpretability analyses [**??**], we introduce the *Minimum Sufficient Head Circuit* (MSHC)—the smallest set of heads whose activations suffice to solve a task within a user-specified tolerance. Extracting MSHCs for arithmetic verification, mathematical word-problems, and grammatical acceptability across three open-weight models (Gemma-9B, Llama-8B, Qwen-8B), we uncover a substantial overlap between the arithmetic and word-problem circuits and only minimal intersection with the grammar circuit. These results indicate that small LLMs reuse a shared numerical sub-circuit across superficially different tasks while maintaining a distinct pathway for syntactic reasoning. We quantify this overlap, analyse how it scales with model size, and discuss implications for parameter-efficient fine-tuning and model editing.

The field has witnessed exponential growth in model size, from early transformer models with hundreds of millions of parameters [Vaswani et al., 2017] to modern giants like GPT-4 [OpenAI, 2023] that likely contain trillions of parameters. While these massive models have captured headlines with their impressive capabilities in both language and mathematics, a parallel revolution has been unfolding in the development of smaller, more efficient models in the 1-8 billion parameter range. Models like Llama 2 [Touvron et al., 2023b], Gemma 7B [Jiang et al., 2023], and Qwen [Bai et al., 2023] have demonstrated remarkable performance in both linguistic and numerical tasks despite their relatively modest size, making them particularly valuable for practical applications where computational efficiency and deployment costs are significant concerns.

These smaller LLMs offer a compelling balance between capability and efficiency, enabling deployment on consumer hardware, edge devices, and resource-constrained environments. Their reduced inference costs make them attractive for commercial applications, while their smaller memory footprint allows for fine-tuning and adaptation with more modest computational resources. Understanding how these models encode and represent both grammatical and numerical knowledge is therefore not merely an academic exercise but has significant practical implications for developing more capable, efficient, and cognitively robust language technologies that can handle both linguistic and mathematical reasoning.

2 Related Work and Background

2.1 Linguistic and Mathematical Evaluation of Language Models

Research on evaluating neural language models has evolved from early work on LSTM-based architectures [Linzen et al., 2016] to comprehensive assessment of transformer-based models across both linguistic and mathematical domains [Goldberg, 2019, Devlin et al., 2019, **?**]. For grammatical evaluation, frameworks have progressed from simple agreement tests to comprehensive benchmarks like CoLA [Warstadt et al., 2019] and BLiMP [Warstadt et al., 2020]. For mathematical reasoning, benchmarks include GSM8K [**?**], SVAMP [**?**], and MathQA [**?**], which test arithmetic computation, word problem understanding, and quantitative reasoning. Recent studies have revealed that while performance in both domains scales with size, challenges remain with complex hierarchical structures in grammar and multi-step reasoning in mathematics [Thrush et al., 2022, Qian et al., 2022, **?**].

2.2 Scale and Cognitive Competence

The relationship between model scale and cognitive abilities follows complex patterns beyond the general power-law scaling observed in language modeling [Kaplan et al., 2020]. Research suggests that rare grammatical constructions and complex arithmetic operations require disproportionately more training data [Wei et al., 2021, **?**]. Certain phenomena in both domains show nonlinear improvements at specific parameter thresholds [Zhang et al., 2023], with mathematical reasoning often exhibiting steeper scaling curves than linguistic tasks [**?**]. Studies on compositional generalization indicate that scaling alone may not capture human-like cognitive productivity without architectural innovations that support both symbolic and statistical processing [Hu and Daumé III, 2020, **?**].

2.3 Probing Language Models for Cognitive Knowledge

Researchers have developed various probing techniques to understand how linguistic and mathematical knowledge is represented within model parameters. Structural probes have revealed that models implicitly encode both syntactic hierarchies and numerical relationships [Hewitt and Manning, 2019, **?**], with different types of knowledge appearing at different network depths [Tenney et al., 2019, **?**]. Studies show that transformer-based models exhibit emergent capabilities resembling both discrete linguistic rules and arithmetic algorithms [Manning et al., 2020, **?**], with individual neurons specializing in specific linguistic features or numerical operations [Geva et al., 2021, Patel and Pavlick, 2022]. Attention patterns often correspond to both syntactic dependencies and mathematical relations, with different attention heads specializing in distinct cognitive phenomena [Clark et al., 2019, **?**].

2.4 Small LLMs and Comparative Studies

The proliferation of open-weight models like OPT [Zhang et al., 2022], Llama [Touvron et al., 2023a], Gemma [**?**], and Qwen [Bai et al., 2023] has enabled more systematic analyses of how capabilities in both linguistic and mathematical domains scale and how architectural choices affect performance. Studies show that smaller models with innovative architectures can outperform larger ones in specific aspects of language and mathematics [**?**], and data quality may be as important as scale for robust cognitive representations [**?**]. Comparative analyses across architectures remain limited but suggest significant variations in performance even at similar parameter scales, with some models showing domain-specific strengths [Talmor et al., 2020, Zhao et al., 2023]. Recent frameworks for quantifying evaluation uncertainties [Roberts et al., 2023] and structured evaluation approaches [Xia et al., 2023] have revealed that architectural design choices impact both grammatical and mathematical phenomena differently, suggesting that model architectures encode cognitive knowledge in fundamentally different ways.

3 Methodology

Our analysis proceeds in three strands:

(1) Low-dimensional linear separability (LS). We measure how well a *D-dimensional* projection of the hidden state at each layer separates correct from incorrect examples (D $\leq$ 3).

(2) Minimum Sufficient Head Circuit (MSHC). Guided by the LS curves we isolate the smallest set of attention heads whose activations suffice for the task.

(3) Controlled datasets. All tasks are cast as *minimal pairs* so that a single factual change flips the label, letting us attribute separability (or its absence) to that fact alone.

Throughout, let $\mathcal{V}$ denote the vocabulary and $m : \mathcal{V}^* \to \mathbb{R}^d$ a causal transformer with L layers and H heads per layer. For a sequence $x = (x_1, \ldots, x_n)$ we write $\mathbf{h}_{x,\ell} \in \mathbb{R}^d$ for the activation at the end-of-sequence (EOS) token at layer ℓ, and $\mathbf{a}_{x,\ell,h} \in \mathbb{R}^d$ for the contribution of head $h \in \{1, \ldots, H\}$ in that layer.

Activation collection. For every item we construct the full prompt by concatenating a single one-shot demonstration, the query sentence or equation, and the end-of-sequence marker `</s>` (or its model-specific analogue). The model is executed in teacher-forcing mode. At each layer $\ell \in \{0, \ldots, L\}$ we record the hidden state $\mathbf{h}_{x,\ell}$ at the position of that final EOS token—thus collecting a compact $L+1$-vector trace that summarises the entire computation leading to the model's output logit. We omit intermediate attention projections and all key/value tensors to minimise I/O overhead; subsequent probes operate solely on this last-token trace. Because all prompts fit within the context window, no padding is required, and teacher forcing guarantees determinism across repeated runs.

3.1 Task families and evaluation sets

We consider three evaluation corpora, each built from *minimal pairs* (x_c, x_i) in which the two members differ in exactly one fact that determines correctness. Every item appears in three prompt variants: the raw text, a two-choice question ("A" or "B"), and a single-candidate acceptability probe, each preceded by a one-shot demonstration so that prompting remains uniform across models.

Grammar (G). The 67k sentence pairs from BLiMP [Warstadt et al., 2020] cover twelve syntactic phenomena.

Good Sentence: *Who should Derek hug after shocking Richard?*
Bad Sentence: *Who should Derek hug Richard after shocking?*

Arithmetic verification (A). We generate 10^3 addition pairs with addends $n_1, n_2 \in [1, 10^3]$; incorrect results satisfy the deviation constraint in Eq. (1).

Good Equation: 1338+ 88 = 1426
Bad Equation: 1338 + 88 = 2139

$$.5 * (n1 \pm n2) \leq \text{noisy}(n1 \pm n2) \leq 1.5 * (n1 \pm n2) \tag{1}$$

Word-problem arithmetic (W). The 100 story problems are produced by numerically perturbing the ending of a template narrative:

Good Expression: Tim has 5 apples and eats 2, leaving him with 3 apples.
Bad Expression: Tim has 5 apples and eats 2, leaving him with 10 apples.

where the perturbation is applied by taking a sentence generated with a template and replacing the numbers with perturbed numbers from the 10000-item dataset of addition/subtraction pairs generated as in (1).

3.2 Low-dimensional linear separability metric

A well-known pathology of very high–dimensional spaces is that almost *any* two finite clouds are linearly separable with overwhelming probability [see **?**]. Raw accuracy of a linear probe in the full residual space $\mathbb{R}^d$ therefore over-states how "easy" a task is. To obtain a more conservative—and hence more informative—measure of representational structure, we evaluate separability after first collapsing the hidden states onto just *two* principal directions. We call the resulting statistic the *low-dimensional linear separability* score, $\mathrm{LS}_{t,\ell}$.

Notation. Fix a task t and a transformer layer ℓ. For every prompt x we write $\mathbf{h}_{x,\ell} \in \mathbb{R}^d$ for the centred hidden state at the EOS position: $\mathbf{h}_{x,\ell} := \mathbf{h}_{x,\ell}^{(n)} - \bar{\mathbf{h}}_\ell$, where $\bar{\mathbf{h}}_\ell = \frac{1}{|\mathcal{D}_t|} \sum_{x \in \mathcal{D}_t} \mathbf{h}_{x,\ell}^{(n)}$ is the layer mean computed over the complete training split $\mathcal{D}_t$.

Step A: variance-maximising projection. Let $\Sigma_\ell = \frac{1}{|\mathcal{D}_t|} \sum_{x \in \mathcal{D}_t} \mathbf{h}_{x,\ell} \mathbf{h}_{x,\ell}^\top$ be the empirical covariance matrix. We seek an orthonormal matrix $\mathbf{W}_\ell \in \mathbb{R}^{d \times 2}$ that captures the greatest possible variance under a rank-2 constraint:

$$\max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}_2} \operatorname{Tr}\big(\mathbf{W}^\top \Sigma_\ell \mathbf{W}\big). \tag{2}$$

The optimal columns are the two leading eigenvectors of Σ_ℓ. The corresponding 2-D coordinates are $\tilde{\mathbf{h}}_{x,\ell} = \mathbf{W}_\ell^\top \mathbf{h}_{x,\ell} \in \mathbb{R}^2$.

Step B: linear decision boundary. Assign labels $y_i = +1$ for *correct* members of a minimal pair and $y_i = -1$ for *incorrect* ones. We train a soft-margin support-vector machine in the projected space:

$$\min_{\mathbf{w},b} \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N_t} \max\big(0,\, 1 - y_i(\mathbf{w}^\top \tilde{\mathbf{h}}_{x_i,\ell} + b)\big), \qquad C = 10. \tag{3}$$

Because $\tilde{\mathbf{h}}_{x,\ell} \in \mathbb{R}^2$, the resulting classifier depends on at most three parameters, precluding the kind of pathological over-fitting that haunts full-dimensional probes.

Definition of the metric. Let $\hat{y}_i = \operatorname{sign}\big(\mathbf{w}^\top \tilde{\mathbf{h}}_{x_i,\ell} + b\big)$ denote the SVM's prediction on a held-out validation example. We define

$$\mathrm{LS}_{t,\ell} = \frac{1}{N_t^{\text{val}}} \sum_{i=1}^{N_t^{\text{val}}} \mathbf{1}[\hat{y}_i = y_i], \qquad \mathrm{LS}_{t,\ell} \in [0, 1]. \tag{4}$$

Interpretation. Random guessing gives $\mathrm{LS}_{t,\ell} = 0.5$. A score close to 1 implies that the first two principal axes already support a linear decision boundary, i.e. the class-conditional embeddings differ in a *low-codimension* direction. Conversely, scores near 0.5 indicate either geometric entanglement or dispersion of the signal across many dimensions.

In what follows we plot $\ell \mapsto \mathrm{LS}_{t,\ell}$ for each task. Peaks in these curves spotlight layers where the model's hidden states make the good/bad distinction in the simplest possible way: by shifting along just two orthogonal directions.

3.3 The Minimum Sufficient Head Circuit

A transformer's computation is frequently dominated by a surprisingly small subset of its attention heads. We formalise this with the *Minimum Sufficient Head Circuit* (MSHC). Fix an accuracy tolerance $\epsilon \in (0, 0.5)$. Let $\mathcal{H} = \{(\ell, h) : 1 \le \ell \le L,\ 1 \le h \le H\}$ be the set of all heads. A subset $\mathcal{C} \subseteq \mathcal{H}$ is an **MSHC**$_\epsilon$ if, *with high probability (WHP)*, enabling any single head from $\mathcal{C}$ is already sufficient to lift accuracy above chance:

$$(\forall h \in \mathcal{C}) \qquad \Pr_{x\sim\mathcal{D}}\big[\mathrm{Acc}\big(m; \{h\}\big) > 0.5 + \epsilon\big] \ \ge\ 1 - \delta,$$

where δ is a user-specified failure probability (we use $\delta = 0.05$ in all experiments), and $\mathcal{C}$ is inclusion-minimal with this property.

Hunting the circuit Our search still employs a sliding window but now ranks layers by the *size* of the accuracy loss they incur. We slide a window of width $w = \lfloor xL \rfloor$ layers from the bottom to the top of the network, disable *all* heads in that window, and measure the resulting accuracy $\mathrm{Acc}_{\mathrm{off}}$. For every layer ℓ covered by the current window we store $\textsc{AccLayer}[\ell] = \max\big(\textsc{AccLayer}[\ell], \mathrm{Acc}_{\mathrm{off}}\big)$—the *best* accuracy ever observed when *any* window that contains ℓ is switched off. After scanning all windows we compute a per-layer drop score $\Delta_\ell = \mathrm{Acc}_{\mathrm{full}} - \textsc{AccLayer}[\ell]$ and keep those layers whose Δ_ℓ lies in the top 25th percentile. These high-impact layers seed the head-level search:

Step 1. Disable every head *in the selected top-quartile layers*, establishing a "dark-start" baseline.

Step 2. Run a stochastic pruning loop (Alg. 1) that begins with bundle size $k_0 = \lceil 2\sqrt{|\mathcal{C}|}\rceil$. For each bundle size k, we draw N random bundles of k heads, measure their accuracies, and remove the bundle that attains the *lowest* accuracy whenever that value is $\le 0.5 + \epsilon$. When the worst of the N bundles exceeds the threshold, we tighten the constraint by updating $k \leftarrow \max(1, \lfloor k/2 \rfloor)$ and continue.

Step 3. Stop when $k = 1$ and every random bundle succeeds—by definition, the remaining heads form an MSHC.

The procedure terminates because the candidate set shrinks monotonically and can be pruned at most $|\mathcal{H}|$ times.

The narrative interpretation is simple: we first find *where* knowledge lives, then keep only those filaments that reliably relight the lamp.

Empirically, the circuit coalesces in fewer than 300 iterations on all three models studied—Gemma-9B, Llama-8B, and Qwen-8B.

3.4 Experimental protocol

Our analysis focuses on three open-weight checkpoints that occupy a comparable size class yet differ architecturally: *Gemma-9B*, *Llama-8B* and *Qwen-8B*. Each model is evaluated on exactly the same train/validation/ test splits (60 : 20 : 20) of the G, A, and W corpora described above. For a given model we first trace the accuracy-vs-layer curves (§**??**), then feed the validation set to the sliding-window search with $(\epsilon, k) = (0.1, 3)$ and a window fraction $x = 0.2$. The resulting MSHC is frozen before we inspect test performance, and uncertainty bands are computed via 1 000-sample bootstrap over minimal pairs. All experiments run on a single A100 80 GB GPU; wall-clock times are listed in Appendix C.

Algorithm 1 Sliding-window percentile localisation followed by stochastic discovery of an MSHC_ϵ

Require: Window fraction x, tolerance ϵ, sample count N

$\mathrm{Acc}_{\mathrm{full}} \leftarrow \mathrm{Acc}(m)$
$w \leftarrow \lfloor xL \rfloor$
initialise array $\textsc{AccLayer}[1\!:\!L] \leftarrow 0$ ▷ best accuracy seen with each layer disabled
for $s = 1$ **to** $L - w + 1$ **do**
  disable all heads in layers s to $s + w - 1$
  $\mathrm{Acc}_{\mathrm{off}} \leftarrow \mathrm{Acc}(m)$
  for $\ell = s$ **to** $s + w - 1$ **do**
    $\textsc{AccLayer}[\ell] \leftarrow \max\big(\textsc{AccLayer}[\ell],\, \mathrm{Acc}_{\mathrm{off}}\big)$
  re-enable the disabled layers
▷ compute per-layer drop scores
for $\ell = 1$ **to** L **do**
  $\textsc{Drop}[\ell] \leftarrow \mathrm{Acc}_{\mathrm{full}} - \textsc{AccLayer}[\ell]$
$\tau \leftarrow$ 75th percentile of $\textsc{Drop}$
$\mathcal{L} \leftarrow \{\ell \mid \textsc{Drop}[\ell] \geq \tau\}$ ▷ top-quartile layers
$\mathcal{C} \leftarrow$ all heads in $\mathcal{L}$ ▷ initial candidate circuit
$k \leftarrow \left\lceil 2\sqrt{|\mathcal{C}|} \right\rceil$
while $k \geq 1$ **do**
  repeat
    minAcc $\leftarrow 1$; $\mathcal{K}_{\min} \leftarrow \emptyset$
    for $i = 1$ **to** N **do**
      draw $\mathcal{K} \sim \mathrm{Unif}\{\mathcal{S} \subseteq \mathcal{C} : |\mathcal{S}| = k\}$
      acc $\leftarrow \mathrm{Acc}(m; \mathcal{K})$
      if acc $<$ minAcc **then**
        minAcc $\leftarrow$ acc; $\mathcal{K}_{\min} \leftarrow \mathcal{K}$
    if minAcc $\leq 0.5 + \epsilon$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{K}_{\min}$ ▷ prune the worst bundle
  until minAcc $> 0.5 + \epsilon$
  $k \leftarrow \max(1, \lfloor k/2 \rfloor)$
return $\mathcal{C}$

4 Experiments

5 Discussion

6 Conclusion

References

Jinze Bai, Shuai Lv, Sheng Peng, Yida Wang, Xingjian Zhang, Ziyue Yang, Beilei Yang, Haotian Gong, Zhiyu Fu, Kongming Liu, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

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

Yoav Goldberg. Assessing bert's syntactic abilities. In *arXiv preprint arXiv:1901.05287*, 2019.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word 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 4129–4138, 2019.

Weihs Hu and Hal Daumé III. Systematic evaluation of causal discovery in visual model based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12578–12590, 2020.

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. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Krishna Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. *arXiv preprint arXiv:2210.02539*, 2022.

Peng Qian, Tahsina Huang, Reza Firoozi, Zhiyuan Wang, Qiyang Zhou, Eric Wong, Kevin Chen, Shaopeng Pan, Zhou Yu, Yang Xiang, et al. Limitations of language models in arithmetic and symbolic reasoning. *arXiv preprint arXiv:2208.05051*, 2022.

Adam Roberts, Albert Webson, Colin Larson, Leo Gao, Niket Tandon, Kai-Wei Tai, Hyung Won Chung, Colin Raffel, and Gaurav Mishra. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. olmpics-on what language model pre-training captures. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 743–758. MIT Press, 2020.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, 2019.

Tristan Thrush, Sanjay Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. In *Transactions of the Association for Computational Linguistics*, volume 7, pages 625–641. MIT Press, 2019.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020.

Jerry Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Frequency effects on syntactic rule learning in transformers. *arXiv preprint arXiv:2109.07020*, 2021.

Jiacheng Xia, Songbo Li, Haozhao Xu, Danny Chen, Yang Liu, Bill Cohen, and Leyang Zhang. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2023.

Hugh Zhang, Amy Webb, Saujas Petryk, Yiheng Han, Jason Lei, and Chelsea Finn. Language modeling with reduced spurious correlations. *arXiv preprint arXiv:2306.01708*, 2023.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in appendix.