`softmax` is not enough (for sharp out-of-distribution)

Petar Veličković Google DeepMind
Christos Perivolaropoulos Google DeepMind
Federico Barbero* University of Oxford
Razvan Pascanu Google DeepMind

Abstract

A key property of reasoning systems is the ability to make *sharp* decisions on their input data. For contemporary AI systems, a key carrier of sharp behaviour is the `softmax` function, with its capability to perform differentiable query-key lookups. It is a common belief that the predictive power of networks leveraging `softmax` arises from "circuits" which sharply perform certain kinds of computations consistently across many diverse inputs. However, for these circuits to be robust, they would need to generalise well to *arbitrary* valid inputs. In this paper, we dispel this myth: even for tasks as simple as finding the maximum key, any learned circuitry *must disperse* as the number of items grows at test time. We attribute this to a fundamental limitation of the `softmax` function to robustly approximate sharp functions, prove this phenomenon theoretically, and propose *adaptive temperature* as an ad-hoc technique for improving the sharpness of `softmax` at inference time.

1 Motivation

It is no understatement to say that the $\texttt{softmax}_\theta : \mathbb{R}^n \to [0,1]^n$ function[1]:

$$\texttt{softmax}_\theta(\mathbf{e}) = \left[\frac{\exp(e_1/\theta)}{\sum_k \exp(e_k/\theta)} \quad \cdots \quad \frac{\exp(e_n/\theta)}{\sum_k \exp(e_k/\theta)}\right] \tag{1}$$

is one of the most fundamental functions in contemporary artificial intelligence systems.

The role of `softmax` in deep learning is to convert any vector of *logits*, $\mathbf{e} \in \mathbb{R}^n$, into a *probability distribution*, in a form that is part of the *exponential* family. Further, `softmax` allows for application of a *temperature* parameter, $\theta \in \mathbb{R}$, to adjust the amount of probability mass attached to the highest logit—a concept borrowed from the Boltzmann distribution in statistical mechanics.

Initially, the primary utilisation of `softmax` in deep learning was within the final layer of *classifiers*. Its influence in this domain vastly expanded after it saw use in the *internal* layers—as a differentiable key-value store [GWD14] or a mechanism for *attending* over the most relevant parts of the input [BCB15]. This *attentional* framing of `softmax` was critical in defining important models for sequences [VSP+17, Transformers], images [DBK+21, ViTs] and graphs [VCC+18, GATs].

Several efforts attribute the success of `softmax` to its capability of modelling computations relevant to reasoning. This can be related to the concept of *circuits* in theoretical computer science [AB09]. Several interpretable pieces of "circuitry" [OCS+20] have already been discovered in large Transformers, primarily under the umbrella of *mechanistic interpretability* [ENO+21, OEN+22, WVC+22].

Here we study the robustness of such circuitry, especially when going beyond the distribution the models are trained on—a critical regime for *reasoning engines*. We find that, in spite of its many successes, `softmax` *does not have a chance* to robustly generalise such circuits out of distribution, especially because it provably cannot approximate **sharpness** with increasing problem size (Figure 1). Here we call a function *sharp* if its output only depends on a *constant* number of its inputs (e.g. `max`).

*Work performed while the author was at Google DeepMind.

[1]Strictly speaking, the proper name for this function should be `soft`**`arg`**`max`. We choose to retain the terminology introduced by [Bri89], primarily for reasons of alignment with modern deep learning frameworks.

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

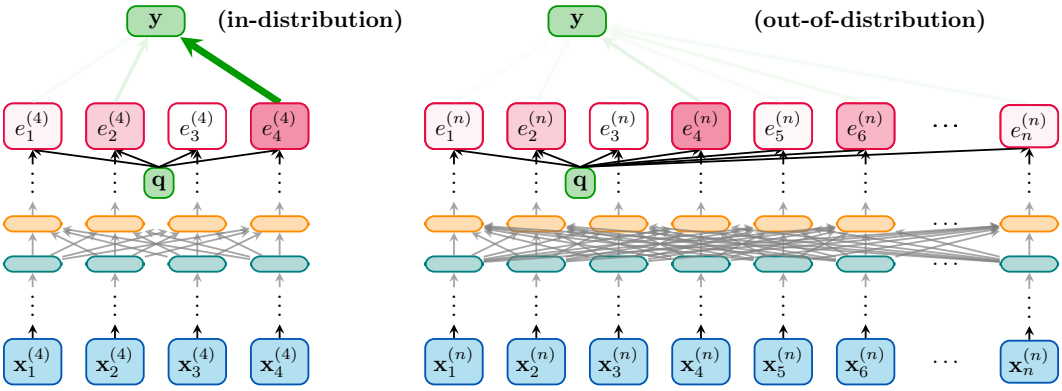

Figure 1: Illustration of Theorem 2.2, one of our key results. Assuming a tokenised input from a fixed vocabulary and a non-zero temperature, for every `softmax` attention head inside an architecture comprising only **MLPs** and **`softmax` self-attention layers**, it must hold that, given sufficiently many **tokens**, its **attention coefficients** will *disperse*, even if they were sharp for in-distribution instances.

We hope that this result encourages future study of alternative attentional functions, in light of the problems we identify, especially for building reasoning engines of the future. That being said, we also believe that our findings indicate ways to modify the `softmax` function to support sharpness for longer—as one simple example of this, we propose an *adaptive temperature* mechanism for `softmax`.

Background The analysis of attentional coefficients and attempting to attribute interpretable operations to them dates back to the earliest deployments of internal `softmax` layers at scale; examples include [GWD14, Figure 6], [BCB15, Figure 3], [VSP+17, Figures 3–5] and [QTM+18, Figure 5]. A strong current in this space analyses the self-attentional heads of Transformers [VTM+19, JW19].

With the rise of large language models, mechanistic interpretability has taken charge in detecting and elucidating various circuits in Transformers [ENO+21]. Some prominent discoveries include induction heads [OEN+22], indirect object identification [WVC+22], multiple-choice heads [LRK+23], successor heads [GOOC23], attentional sinks [DOMB23], comparator heads [HLV24] and retrieval heads [WWX+24]. Most recently, these efforts have relied on sparse autoencoders [KKB+24].

While the skills above are quite impressive and span many rules one might hope a robust reasoning system would have, and the discovered heads always appear sharp when inspected, it is also known that many *easy* tasks requiring sharp attention—such as finding minima—are hard to do reliably with LLMs out-of-distribution [MMI+24, Figure 6]. More challenging sharp order statistic tasks, such as finding the second minimum [OV22] may even be hard to learn in-distribution. The discrepancy of such results with the previous paragraph motivate our study, and formalisation of `softmax` dispersion.

Certain dispersion effects in `softmax`—e.g. as an effect of increasing temperature—are already well-understood in thermodynamics. A core contribution of our work is understanding dispersion in a setting where the **amount of logits can vary**, which is relevant for generalisation in Transformers. We are not the first to observe dispersion in this setting empirically; prior works studying the capability of Transformers to execute algorithms [YSK+20] and perform random-access lookups [EPM24] also note dispersion patterns. Our work is the first to rigorously prove these effects, directly attribute them to the `softmax` operator, as well as propose ways to improve sharpness empirically within `softmax`. The proof technique we will use to demonstrate this is inspired by [BBK+24], though unlike their work, our results apply regardless of whether the computational graph is bottlenecked or not.

2 Demonstrating and proving the dispersion in `softmax` and Transformers

To motivate our theory, we train a simple architecture including a single dot-product attention head to predict a feature of the *maximum* item in a set. Each item's features are processed with a deep MLP before attending, and the output vector of the attention is passed to a deep MLP predictor (see Appendix A for experimental details). We train this model using sets of ≤ 16 items, and in Figure 2 we visualise the head's attentional coefficients, computed over sets of varying size at inference time.

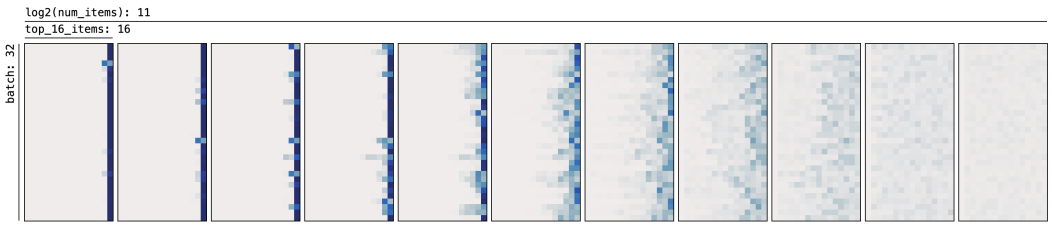

Figure 2: Visualising the attentional head for the `max` retrieval task for a batch of 32 sets, over the 16 items with largest key. If the head operates correctly, it must allocate sharp attention to the rightmost item. From left to right, in each frame we *double* the number of items the head has to process.

While the model indeed attributes focus sharply and cleanly on the maximum item, this only holds true on the problem sizes that the model was trained on. As we simulate an out-of-distribution setting where the problem size increases (without changing the value distribution), the attentional coefficients eventually disperse towards the uniform distribution.

This effects manifests in the attention heads of Transformers as well—we visualise the *entropy* (a proxy for sharpness) of Gemma 2B [GMH+24]'s heads when answering a similar maximisation task in Figure 3.

In fact, we can show that this effect is *inevitable* in `softmax` using the following Lemma (proved in Appendix B):

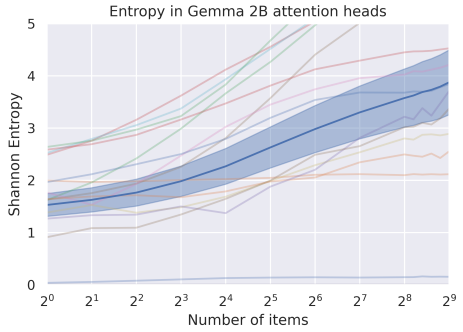

Figure 3: Entropy of attention heads in the first block of Gemma 2B with prompt "`What is the maximum in the following sequence: {seq}? The maximum is:`" and varying the number of elements in `seq`.

Lemma 2.1 (`softmax` must disperse)**.** *Let $\mathbf{e}^{(n)} \in \mathbb{R}^n$ be a collection of n logits going into the $\texttt{softmax}_\theta$ function with temperature $\theta > 0$, bounded above and below s.t. $m \leq e_k^{(n)} \leq M$ for some $m, M \in \mathbb{R}$. Then, as more items are added ($n \to +\infty$), it must hold that, for each item $1 \leq k \leq n$, $\texttt{softmax}_\theta(\mathbf{e}^{(n)})_k = \Theta(\frac{1}{n})$. That is, the computed attention coefficients **disperse** for all items.*

Lemma 2.1 relies on being able to bound the logit values with specific constants. In modern Transformer architectures operating over a vocabulary of possible token values, we can actually bound the logits in every single attentional layer—implying that dispersion *must* happen everywhere in a Transformer for sufficient problem sizes. We prove this important result in Appendix C.

Theorem 2.2 (`softmax` in Transformers over vocabularies must disperse)**.** *Let $\mathcal{X} \subset \mathbb{R}^m$ be an m-dimensional input feature space, and let $\mathbf{X}^{(n)} \in \mathcal{X}^n$ be a matrix of input features for n items. Further, assume that input features come from a **finite** set of possible values, i.e. $|\mathcal{X}| < |\mathbb{N}|$. Let $e_j^{(n)} = (\mathbf{q}^{(n)})^\top \mathbf{k}_j^{(n)}$ where $\mathbf{q}^{(n)} = \phi(\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_n^{(n)})$ and $\mathbf{K}^{(n)} = \kappa(\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_n^{(n)})$, where $\phi : \mathcal{X}^n \to \mathbb{R}^k$ and $\kappa : \mathcal{X}^n \to \mathbb{R}^{n \times k}$ are continuous functions, each expressible as a composition of L layers $g_L \circ f_L \circ \cdots \circ g_1 \circ f_1$ where each layer contains a feedforward component $f_i(\mathbf{z}_1, \ldots, \mathbf{z}_n)_k = f_i(\mathbf{z}_k)$ or a self-attentional component $g_i(\mathbf{z}_1, \ldots, \mathbf{z}_n)_k = \sum_{1 \leq l \leq n} \alpha_{lk} v_i(\mathbf{z}_l)$ where $\alpha_{lk} \in [0, 1]$ are $\texttt{softmax}$-normalised attention coefficients and v_i is a feedforward network. Then, for any $\theta > 0$ and $\epsilon > 0$, there must exist an $n \in \mathbb{N}$ such that $\texttt{softmax}_\theta(\mathbf{e}^{(n)})_k < \epsilon$ for all $1 \leq k \leq n$. That is, attention coefficients must **disperse** in all Transformer heads if the input vocabulary is finite.*

3 Adaptive temperature

Since we now know dispersion is inevitable, are there any ways we can leverage our theory's findings to make `softmax` sharper? One obvious constraint our theory rests on is the assumption that $\theta > 0$, i.e. that our temperature is nonzero. While zero temperature—a special case of *hard attention*

Table 1: Improvements observed when applying adaptive temperature on the max retrieval task (without changing the parameters), averaged over ten seeds. p-values computed using a paired t-test.

Model	ID size	Out-of-distribution sizes								
	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384
Baseline	**98.6%**	**97.1%**	94.3%	89.7%	81.3%	70.1%	53.8%	35.7%	22.6%	15.7%	12.4%
Adaptive θ	**98.6%**	**97.1%**	**94.5%**	**89.9%**	**82.1%**	**72.5%**	**57.7%**	**39.4%**	**24.9%**	**17.5%**	**14.0%**
p-value	0.4	0.4	0.002	$2 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	$3 \cdot 10^{-5}$	10^{-4}	$6 \cdot 10^{-4}$	0.02	10^{-3}	$4 \cdot 10^{-3}$

[DBLdF12, Ran14, MHG+14, XBK+15, MA16, CNM19, PNM19]—guarantees sharpness, training large-scale Transformers with it tends to not work well in practice [BIB+24].

What about applying zero temperature to an *already-trained* Transformer? We can show that this is also problematic since, for any attention head where the Transformer has learnt to induce sharpness, it *necessarily* did so by increasing magnitude of its weights (see Appendix D for a proof):

Proposition 3.1 (Sharpness in Transformers necessitates large weights)**.** *Let $\mathbf{e}^{(n)} \in \mathbb{R}^n$ be a collection of n logits, computed using a dot product attention mechanism; i.e. $e_k^{(n)} = \langle \mathbf{Q}\mathbf{y}, \mathbf{K}\mathbf{x}_k \rangle$, where $\mathbf{y} \in \mathbb{R}^m$ is a query vector and $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{m' \times m}$ are parameters. Let $\delta = \max_{1 \le i \le n} e_i^{(n)} - \min_{1 \le j \le n} e_j^{(n)}$ be their maximum difference. Then δ is upper bounded as $\delta \le 2\sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \max_{1 \le i \le n} \|\mathbf{x}_i\|$, where $\sigma_{\max}^{(Q)}, \sigma_{\max}^{(K)} \in \mathbb{R}$ are the largest singular values of $\mathbf{Q}$ and $\mathbf{K}$. That is, the sharpness of the softmax in Transformers depends on the norm of its parameters.*

However, forcing parameters to be of large magnitude promotes overfitting, and the likelihood that the *incorrect* item gets the largest logit—as can be observed in several cases in Figure 2. As such, setting temperature to zero will *degrade* accuracy—we might prefer a solution that makes the coefficients sharper while making sure that the chosen item is not left behind.

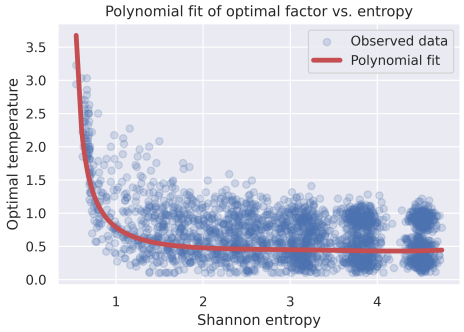

Figure 4: The polynomial fit used to derive our adaptive formula for θ as a function of the Shannon entropy, H. The fit degree-4 function was $\theta \approx 1/(-1.791 + 4.917H - 2.3H^2 + 0.481H^3 - 0.037H^4)$.

This motivates our use of **adaptive temperature**, where we vary θ depending on the *entropy* in the input coefficients. Adaptive temperature can be elegantly motivated by the fact that decreasing the temperature must monotonically decrease the entropy, which is well-known in thermodynamics. We demonstrate it both theoretically (Proposition E.1) and empirically (Figures 5–6) in Appendix E. Note we are not the first to propose dynamically adapting temperature—[NZV18, RKH+21] do this in the classification layer (and hence do not have to handle an ever-increasing amount of items), whereas [CC22, CRF24] perform it over intermediate attentional heads, but in a way that only depends on problem size (e.g. multiplying logits by $\log n$), hence not taking into account initial logit sharpness. It is important to also call out Entropix [xd24], a notable library for (var)entropy-based LLM sampling.

To compute the approximate temperature value as a function of entropy, we generate a dataset of inputs to our model where the maximal items do not obtain the highest logit. For each such input, we find the "optimal" value of θ that would maximise its probability. Then we fit an inverse degree-4 polynomial to this data—see Figure 4—and use it to predict temperatures to use at inference time. Note we do not wish to increase entropy; as such, we do not apply the correction to θ if it's predicted to be greater than 1. Our proposed temperature adaptation indeed leads to sharper coefficients (Appendix E, Figure 6) and improved out-of-distribution accuracy on the max retrieval task (Table 1).

We conclude by remarking, once again, that adaptive temperature is an *ad-hoc* method and it does not escape the conclusions of our theory! Any kind of unnormalised attention, such as *linear* [Sch92] or *sigmoid* attention [RDD+24] does not have such issues. Similarly, forcing *hard* or *local* attention, or inserting *discontinuities* in the feedforward layers [DvGPV24] breaks the assumptions of our theory. While such approaches haven't seen as much success at scale as the "vanilla" Transformer, we hope our results inspire future work into making them stable, especially for constructing reasoning systems.

4 Debunking Challenge Submission

4.1 What commonly-held position or belief are you challenging?

Provide a short summary of the body of work challenged by your results. Good summaries should outline the state of the literature and be reasonable, e.g. the people working in this area will agree with your overview. You can cite sources beside published work (e.g., blogs, talks, etc).

It is a commonly-held belief that deep learning architectures relying on `softmax` attentional aggregators (such as Transformers) are capable of robust reasoning because their attentional heads specialise to *sharp* values and learn to simulate specific circuits [AB09] over the inputs. There is a wide body of work, in mechanistic interpretability and beyond, uncovering such candidate circuits.

Some prominent discoveries (restated from the main paper) include induction heads [OEN+22], indirect object identification [WVC+22], multiple-choice heads [LRK+23], successor heads [GOOC23], attentional sinks [DOMB23], comparator heads [HLV24] and retrieval heads [WWX+24].

4.2 How are your results in tension with this commonly-held position?

Detail how your submission challenges the belief described in (1). You may cite or synthesize results (e.g. figures, derivations, etc) from the main body of your submission and/or the literature.

Our work does not deny that Transformer attention heads are capable of circuit-like behaviours *over specific inputs*—likely ones the Transformer was prepared for in some manner at training time. We challenge the belief that these observed behaviours are truly robust across all relevant, valid inputs.

We prove (Theorem 2.2) that no `softmax`-based attention head in Transformer architectures can ever remain sharp over all possible inputs, and that it *must disperse*. We also empirically demonstrate this dispersion effect, over carefully controlled synthetic tasks (Figure 1) and within LLMs (Figure 3).

4.3 How do you expect your submission to affect future work?

Perhaps the new understanding you are proposing calls for new experiments or theory in the area, or maybe it casts doubt on a line of research.

We expect that our submission will call upon greater resources to be committed to investigating variants of attentional architecture which do not feature solely the `softmax` operator.

Such architectural proposals that escape the confines of our theoretical results already exist—examples include linear attention [Sch92], sigmoidal attention [RDD+24], hard attention [DBLdF12, Ran14, MHG+14, XBK+15] or introducing other kinds of discontinuities in the feedforward layers [DvGPV24]. These have not seen as much broad initial success at scale compared to "vanilla" Transformer architectures, leading to a lack of careful tuning efforts for making them stable.

Our hope is that our results indicate clear motivation for investigating such systems and related ones in greater depth, as well as investing greater efforts into stabilising them and making them performant.

Acknowledgments and Disclosure of Funding

We would like to deeply thank Daniel Johnson, the author of Penzai [Joh24]—without this library, our exploratory analyses would not be nearly as fruitful. Further, we thank Alex Matthews, Andrew Dudzik and João Araùjo for helping us with the proof of one of our propositions, and Arthur Conmy, Neel Nanda and Csaba Szepesvári for reviewing the paper prior to submission and having a plethora of highly useful comments and references. Lastly, we show deep appreciation to Alex Vitvitskyi, Olga Kozlova and Larisa Markeeva, for their endless engineering support with CLRS-Text.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[Axl15] Sheldon Axler. *Linear algebra done right*. Springer, 2015.

[BBK+24] Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João GM Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. Transformers need glasses! information over-squashing in language tasks. *arXiv preprint arXiv:2406.04267*, 2024.

[BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[BFH+18] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[BIB+24] Ioana Bica, Anastasija Ilic, Matthias Bauer, Goker Erdogan, Matko Bošnjak, Christos Kaplanis, Alexey A. Gritsenko, Matthias Minderer, Charles Blundell, Razvan Pascanu, and Jovana Mitrovic. Improving fine-grained understanding in image-text pre-training. In *Forty-first International Conference on Machine Learning*, 2024.

[Bri89] John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.

[CC22] David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7654–7664, 2022.

[CNM19] Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics.

[CRF24] Chenjie Cao, Xinlin Ren, and Yanwei Fu. MVSFormer++: Revealing the devil in transformer's details for multi-view stereo. In *The Twelfth International Conference on Learning Representations*, 2024.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[DBLdF12] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012.

[DOMB23] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[DvGPV24] Andrew Joseph Dudzik, Tamara von Glehn, Razvan Pascanu, and Petar Veličković. Asynchronous algorithmic alignment with cocycles. In *Learning on Graphs Conference*, pages 3–1. PMLR, 2024.

[ENO+21] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[EPM24] MohammadReza Ebrahimi, Sunny Panchal, and Roland Memisevic. Your context is not an array: Unveiling random access limitations in transformers. *arXiv preprint arXiv:2408.05506*, 2024.

[GMH+24] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[GOOC23] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. *arXiv preprint arXiv:2312.09230*, 2023.

[GWD14] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[HG16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[HLO+24] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024.

[HLV24] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36, 2024.

[Joh24] Daniel D. Johnson. Penzai + Treescope: A toolkit for interpreting, visualizing, and editing models as data. *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

[JW19] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[KKB+24] Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. Interpreting attention layer outputs with sparse autoencoders. *arXiv preprint arXiv:2406.17759*, 2024.

[LRK+23] Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

[MA16] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

[MHG+14] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

[MMI+24] Larisa Markeeva, Sean McLeish, Borja Ibarz, Wilfried Bounsi, Olga Kozlova, Alex Vitvitskyi, Charles Blundell, Tom Goldstein, Avi Schwarzschild, and Petar Veličković. The clrs-text algorithmic reasoning language benchmark. *arXiv preprint arXiv:2406.04229*, 2024.

[NZV18] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. 2018.

[OCS+20] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

[OEN+22] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[OV22] Euan Ong and Petar Veličković. Learnable commutative monoids for graph neural networks. In *Learning on Graphs Conference*, pages 43–1. PMLR, 2022.

[PNM19] Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics.

[QTM+18] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2110–2119, 2018.

[Ran14] Marc'Aurelio Ranzato. On learning where to look. *arXiv preprint arXiv:1405.5488*, 2014.

[RDD+24] Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention. *arXiv preprint arXiv:2409.04431*, 2024.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Sch92] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[VCC+18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[VSP+17] 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.

[VTM+19] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[WVC+22] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[WWX+24] Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024.

[XBK+15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR.

[xd24] xjdr and doomslide. Entropix: Entropy Based Sampling and Parallel CoT Decoding, 2024.

[XLZ+20] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020.

[YSK+20] Yujun Yan, Kevin Swersky, Danai Koutra, Parthasarathy Ranganathan, and Milad Hashemi. Neural execution engines: Learning to execute subroutines. *Advances in Neural Information Processing Systems*, 33:17298–17308, 2020.

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

A Experimental details for the maximum entry retrieval task

As briefly described in the main paper, we leverage the `max` retrieval task over a single attention head as a way to empirically validate our theory, as well as assess the benefits of adaptive temperature in a controlled setting. In this section, we describe the various aspects of our experimental setup, for the purposes of clarity and reproducibility.

A.1 Motivation

We deliberately focus on a *single attention head* environment and a *simple* selection function (`max`) to remove any confounders from our observations.

Since we are using exactly one attention head, whatever coefficients it outputs can be directly related to the network's belief in which items are most important for the downstream prediction. This allows us to, e.g., correlate the coefficients with the ground-truth magnitude of the items.

The fact that we are looking for the maximal element's property means we are not requiring any complicated behaviour from the coefficients: when our target task is to approximate `max`, the `softmax` coefficients need to approximate `argmax`—which is exactly what they are designed to be a smooth approximation for. As such, this choice of target task exhibits high algorithmic alignment [XLZ+20].

A.2 Data generation

Let n be the number of items in the set that we wish to classify. For each item, $1 \leq i \leq n$, we need to define a *priority value*, which is used to select the maximal entry. We sample these values from a uniform distribution; $\rho_i \sim \mathcal{U}(0, 1)$.

We would also wish our task to be a *classification* rather than *regression* task, in order to leverage a more robust accuracy metric. As such, let C be the desired number of classes. We can now attach to each item a class, $\kappa_i \sim \mathcal{U}\{1, \ldots, C\}$, sampled uniformly at random. In all our experiments, $C = 10$.

Then, for each input item, $1 \leq i \leq n$, we consider its features to be $\mathbf{x}_i \in \mathbb{R}^{C+1}$ to be defined as $\mathbf{x}_i = \rho_i \| \mathrm{onehot}(\kappa_i, C)$, i.e. the concatenation of these two sampled pieces of data where κ_i is represented as a one-hot vector.

Lastly, since we will leverage dot-product attention, we also need a *query* vector. In this particular task, the query is irrelevant, and we initialise it to a random uniformly-sampled value, $q \sim \mathcal{U}(0, 1)$.

Our task is to predict, given $\{\mathbf{x}_i\}_{1 \leq i \leq n}$ and q, the class of the maximal item, i.e., $\kappa_{\mathrm{argmax}_i \rho_i}$.

A.3 Neural network architecture

The neural network model is designed to be a simple set aggregation model (in the style of Deep Sets [ZKR+17]), with a single-head dot product attention as the aggregation function.

Its equations can be summarised as follows:

$$\mathbf{h}_i = \psi_x(\mathbf{x}_i) \tag{2}$$
$$\mathbf{q} = \psi_q(q) \tag{3}$$
$$e_i = (\mathbf{Q}\mathbf{q})^\top (\mathbf{K}\mathbf{h}_i) \tag{4}$$
$$\alpha_i = \frac{\exp(e_i/\theta)}{\sum_{1\le j\le n}\exp(e_j/\theta)} \tag{5}$$
$$\mathbf{z} = \sum_{1\le i\le n} \alpha_i \mathbf{V}\mathbf{h}_i \tag{6}$$
$$\mathbf{y} = \phi(\mathbf{z}) \tag{7}$$

Equations 2–3 prepare the embeddings of the items and query, using two-layer MLPs ψ_x and ψ_q using the GeLU activation function [HG16] and an embedding size of 128 dimensions. Then, a single-head dot-product attention (with query, key and value matrices $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$) is executed in equations 4–6. Lastly, the output class logits are predicted from the attended vector using a two-layer GeLU MLP, ϕ. Each component is a two-layer MLP to ensure it has universal approximation properties.

A concise implementation of our network using JAX [BFH+18] and Flax [HLO+24] is as follows:

```
import jax.numpy as jnp
from flax import linen as nn
from typing import Callable

class Model(nn.Module):
  n_classes: int = 10
  n_feats: int = 128
  activation: Callable = nn.gelu

  @nn.compact
  def __call__(self, x, q):
    x = nn.Dense(features=self.n_feats)(x)
    x = self.activation(x)
    x = nn.Dense(features=self.n_feats)(x)
    x = self.activation(x)
    q = nn.Dense(features=self.n_feats)(q)
    q = self.activation(q)
    q = nn.Dense(features=self.n_feats)(q)
    x = nn.MultiHeadDotProductAttention(
        num_heads=1,
        qkv_features=self.n_feats)(
        inputs_q=q,
        inputs_kv=x)
    x = nn.Dense(features=self.n_feats)(jnp.squeeze(x, -2))
    x = self.activation(x)
    x = nn.Dense(features=self.n_classes)(x)
    return x
```

A.4 Experimental hyperparameters

We train our model for $100,000$ gradient steps using the Adam SGD optimiser [KB15] with initial learning rate of $\eta = 0.001$. At each step, we present to the model a batch of 128 input sets. All sets within a batch have the same size, sampled uniformly from $n \sim \mathcal{U}\{5, \ldots, 16\}$. The model is trained using a cross-entropy loss, along with L_2 regularisation with hyperparameter $\lambda = 0.001$.

The mixed-size training is a known tactic, designed to better prepare the model for distribution shifts on larger sets at inference time. Similarly, the weight decay follows the recommendation in Proposition 3.1, as an attempt to mitigate overfitting out-of-distribution as a byproduct of sharpening the `softmax` coefficients.

Both methods prove to be effective in deriving a stable baseline model.

B Proof of Lemma 2.1

Lemma 2.1 (softmax must disperse)**.** *Let $\mathbf{e}^{(n)} \in \mathbb{R}^n$ be a collection of n logits going into the $\texttt{softmax}_\theta$ function with temperature $\theta > 0$, bounded above and below s.t. $m \leq e_k^{(n)} \leq M$ for some $m, M \in \mathbb{R}$. Then, as more items are added ($n \to +\infty$), it must hold that, for each item $1 \leq k \leq n$, $\texttt{softmax}_\theta(\mathbf{e}^{(n)})_k = \Theta(\frac{1}{n})$. That is, the computed attention coefficients **disperse** for all items.*

Proof. Let us denote the attentional coefficient assigned to k by $\alpha_k^{(n)} = \texttt{softmax}_\theta(\mathbf{e}^{(n)})_k \in [0, 1]$. Then we can bound $\alpha_k^{(n)}$ above as:

$$\alpha_k^{(n)} = \frac{\exp(e_k^{(n)}/\theta)}{\sum_l \exp(e_l^{(n)}/\theta)} \leq \frac{\exp(M/\theta)}{n \exp(m/\theta)} = \frac{1}{n} \exp\left(\frac{M-m}{\theta}\right) \tag{8}$$

Similarly, we can bound $\alpha_k^{(n)}$ below as:

$$\alpha_k^{(n)} = \frac{\exp(e_k^{(n)}/\theta)}{\sum_l \exp(e_l^{(n)}/\theta)} \geq \frac{\exp(m/\theta)}{n \exp(M/\theta)} = \frac{1}{n} \exp\left(\frac{m-M}{\theta}\right) \tag{9}$$

Hence, if we let $\delta = (M - m)$

$$\frac{1}{n} \exp -\frac{\delta}{\theta} \leq \alpha_k^{(n)} \leq \frac{1}{n} \exp \frac{\delta}{\theta} \tag{10}$$

Which implies $\alpha_k^{(n)} = \Theta(\frac{1}{n})$ as δ and θ are both constants. □

C Proof of Theorem 2.2

Theorem 2.2 (softmax in Transformers over vocabularies must disperse)**.** *Let $\mathcal{X} \subset \mathbb{R}^m$ be an m-dimensional input feature space, and let $\mathbf{X}^{(n)} \in \mathcal{X}^n$ be a matrix of input features for n items. Further, assume that input features come from a **finite** set of possible values, i.e. $|\mathcal{X}| < |\mathbb{N}|$. Let $e_j^{(n)} = (\mathbf{q}^{(n)})^\top \mathbf{k}_j^{(n)}$ where $\mathbf{q}^{(n)} = \phi(\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_n^{(n)})$ and $\mathbf{K}^{(n)} = \kappa(\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_n^{(n)})$, where $\phi : \mathcal{X}^n \to \mathbb{R}^k$ and $\kappa : \mathcal{X}^n \to \mathbb{R}^{n \times k}$ are continuous functions, each expressible as a composition of L layers $g_L \circ f_L \circ \cdots \circ g_1 \circ f_1$ where each layer contains a feedforward component $f_i(\mathbf{z}_1, \ldots, \mathbf{z}_n)_k = f_i(\mathbf{z}_k)$ or a self-attentional component $g_i(\mathbf{z}_1, \ldots, \mathbf{z}_n)_k = \sum_{1 \leq l \leq n} \alpha_{lk} v_i(\mathbf{z}_l)$ where $\alpha_{lk} \in [0, 1]$ are softmax-normalised attention coefficients and v_i is a feedforward network. Then, for any $\theta > 0$ and $\epsilon > 0$, there must exist an $n \in \mathbb{N}$ such that $\texttt{softmax}_\theta(\mathbf{e}^{(n)})_k < \epsilon$ for all $1 \leq k \leq n$. That is, attention coefficients must **disperse** in all Transformer heads if the input vocabulary is finite.*

Proof. Firstly, note that since $\mathcal{X}$ is a finite set of m-dimensional vectors, then it is also part of a *compact* space spanning all convex combinations of those vectors. Then, all feedforward layers, f_i and v_i, being continuous functions, move inputs from a compact set to another compact set. Similarly, every self-attentional layer, g_i, computes a convex combination of the outputs of v_i, and as such, if outputs of v_i are on a compact space, the outputs of g_i remain on the same compact space. Therefore, if the input space of ϕ and κ is compact, then the output space of ϕ and (each row of) κ on $\mathbb{R}^k$ must be compact as well, regardless of the choice of n. Further, the dot product of two vectors $(\mathbf{q}^{(n)})^\top \mathbf{k}_j^{(n)}$ coming from compact spaces must be compact as well. Hence, by definition, the logits must be bounded by $m \leq e_k^{(n)} \leq M$ for constant m and M. Then, letting $\delta = M - m$, we know (Lemma 2.1) that $\texttt{softmax}_\theta(\mathbf{e}^{(n)})_k \leq \frac{1}{n} \exp(\delta/\theta)$, so for all $n > \dfrac{\exp(\delta/\theta)}{\epsilon}$ this value will be below ϵ. □

D Proof of Proposition 3.1

Proposition 3.1 (Sharpness in Transformers necessitates large weights)**.** *Let $\mathbf{e}^{(n)} \in \mathbb{R}^n$ be a collection of n logits, computed using a dot product attention mechanism; i.e. $e_k^{(n)} = \langle \mathbf{Q}\mathbf{y}, \mathbf{K}\mathbf{x}_k \rangle$, where*

$\mathbf{y} \in \mathbb{R}^m$ *is a query vector and* $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{m' \times m}$ *are parameters. Let* $\delta = \max_{1 \le i \le n} e_i^{(n)} - \min_{1 \le j \le n} e_j^{(n)}$ *be their maximum difference. Then* δ *is upper bounded as:*

$$\delta \le 2\sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \max_{1 \le i \le n} \|\mathbf{x}_i\|$$

where $\sigma_{\max}^{(Q)}, \sigma_{\max}^{(K)} \in \mathbb{R}$ *are the largest singular values of* $\mathbf{Q}$ *and* $\mathbf{K}$*. That is, the sharpness of the softmax in Transformers depends on the norm of its parameters.*

Proof. We start by showing that the largest singular values of $\mathbf{Q}$ and $\mathbf{K}$ determine the maximum stretch due to that matrix acting on $\mathbf{x} \in \mathbb{R}^m$. More precisely, we wish to show:

$$\|\mathbf{Q}\mathbf{x}\| \le \sigma_{\max}^{(Q)} \|\mathbf{x}\| \qquad\qquad \|\mathbf{K}\mathbf{x}\| \le \sigma_{\max}^{(K)} \|\mathbf{x}\|$$

where $\|\cdot\|$ is the Euclidean norm. Since both inequalities have the same form, we focus on $\mathbf{Q}$ w.l.o.g. Many of these statements can be derived from linear algebra textbooks [Axl15]. However, the proofs are short enough that we re-derive them here for clarity.

Consider the singular value decomposition (SVD) $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{\Sigma}$ is a rectangular diagonal matrix of singular values $\sigma_i^{(Q)} \in \mathbb{R}$. As $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, $\|\mathbf{U}\mathbf{x}\| = \|\mathbf{V}\mathbf{x}\| = \|\mathbf{x}\|$. Therefore, $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{x}\| = \|\mathbf{\Sigma}\mathbf{v}\|$, where $\mathbf{v} = \mathbf{V}^\top\mathbf{x}$, meaning that $\|\mathbf{v}\| = \|\mathbf{x}\|$. Then we derive:

$$\|\mathbf{\Sigma}\mathbf{v}\| = \|\mathbf{Q}\mathbf{x}\| = \sqrt{\sum_i \left(\sigma_i^{(Q)} v_i\right)^2} \le \sigma_{\max}^{(Q)} \sqrt{\sum_i v_i^2} = \sigma_{\max}^{(Q)} \|\mathbf{x}\|$$

We now note that

$$e_k^{(n)} = \langle \mathbf{Q}\mathbf{y}, \mathbf{K}\mathbf{x}_k \rangle = \|\mathbf{Q}\mathbf{y}\| \|\mathbf{K}\mathbf{x}_k\| \cos\theta$$

with θ the angle between the arguments of the inner product. We can now bound $e_k^{(n)}$ from above:

$$e_k^{(n)} \le \|\mathbf{Q}\mathbf{y}\| \|\mathbf{K}\mathbf{x}_k\| \le \sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \|\mathbf{x}_k\|$$

with $\sigma_{\max}^{(Q)}, \sigma_{\max}^{(K)}$ being the maximum singular value of $\mathbf{Q}$ and $\mathbf{K}$, respectively, and where the last step comes from the inequality shown above. Similarly, we obtain a lower bound, yielding:

$$-\sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \|\mathbf{x}_k\| \le e_k^{(n)} \le \sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \|\mathbf{x}_k\|$$

This gives us the desired upper bound for δ:

$$\begin{aligned}
\delta &= \max_{1 \le i \le n} e_i^{(n)} - \min_{1 \le j \le n} e_j^{(n)} \\
&\le \max_{1 \le i \le n} \sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \|\mathbf{x}_i\| - \min_{1 \le j \le n} -\sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \|\mathbf{x}_j\| \\
&= \sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \max_{1 \le i \le n} \|\mathbf{x}_i\| + \sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \max_{1 \le j \le n} \|\mathbf{x}_j\| \\
&= 2\sigma_{\max}^{(Q)} \sigma_{\max}^{(K)} \|\mathbf{y}\| \max_{1 \le i \le n} \|\mathbf{x}_i\|
\end{aligned}$$

completing the proof. □

E On the relationship between temperature and entropy

There exists a close connection between the *temperature*, θ, leveraged within the $\texttt{softmax}_\theta$ function, and the resulting *Shannon entropy* of the output coefficients. In this section, we explore this relationship through various angles.

First, in Proposition E.1, we show this connection theoretically. This is done by adapting a standard result from thermodynamics (via the *Boltzmann distribution*) into the domain of `softmax`:

Proposition E.1 (Decreasing temperature decreases entropy)**.** *Let* $\mathbf{e}^{(n)} \in \mathbb{R}^n$ *be a collection of* n *logits. Consider the Boltzmann distribution over these* n *items,* $p_i \propto \exp(-\beta e_i^{(n)})$ *for* $\beta \in \mathbb{R}$*, and let* $H = -\sum_i p_i \log p_i$ *be its Shannon entropy. Then, as* β*'s magnitude increases,* H *must monotonically decrease. Thus, since* $\beta \propto \frac{1}{\theta}$ *where* θ *is the temperature in* $\texttt{softmax}_\theta$*, decreasing the temperature must monotonically decrease the entropy.*

Proof. We start by briefly acknowledging the extremal values of β: at $\beta = 0$ (i.e., $\theta \to \infty$), all logits are weighed equally, hence $p_i = \mathcal{U}(n)$ are uniform, and entropy is maximised. Similarly, at $\beta \to \pm\infty$ (i.e., $\theta = 0$), either the minimum or the maximum logit is given a probability of 1, leading to a distribution with minimal (zero) entropy.

Now, consider the partition function $Z = \sum_i \exp(-\beta e_i^{(n)})$, such that $p_i = \frac{\exp(-\beta e_i^{(n)})}{Z}$. We will take derivatives of $\log Z$ with respect to β. Starting with the first derivative:

$$\frac{d}{d\beta} \log Z = \frac{1}{Z} \sum_i -e_i^{(n)} \exp(-\beta e_i^{(n)}) = -\sum_i e_i^{(n)} p_i = -\mathbb{E}_{i\sim p_i}(e_i^{(n)})$$

we recover the expected logit value sampled under the distribution. Now we differentiate again:

$$\begin{aligned}
\frac{d^2}{d\beta^2} \log Z &= -\frac{d}{d\beta} \sum_i e_i^{(n)} p_i \\
&= -\sum_i e_i^{(n)} \frac{d}{d\beta} \frac{\exp(-\beta e_i^{(n)})}{Z} \\
&= -\sum_i e_i^{(n)} \frac{-e_i^{(n)} \exp(-\beta e_i^{(n)}) Z - \exp(-\beta e_i^{(n)}) \sum_j -e_j^{(n)} \exp(-\beta e_j^{(n)})}{Z^2} \\
&= \sum_i (e_i^{(n)})^2 \frac{\exp(-\beta e_i^{(n)})}{Z} - \sum_j e_j^{(n)} \frac{\exp(-\beta e_j^{(n)})}{Z} \frac{\sum_k e_k^{(n)} \exp(-\beta e_k^{(n)})}{Z} \\
&= \sum_i (e_i^{(n)})^2 p_i - \sum_j e_j^{(n)} p_j \sum_k e_k^{(n)} p_k \\
&= \mathbb{E}_{i\sim p_i}((e_i^{(n)})^2) - \mathbb{E}_{i\sim p_i}(e_i^{(n)})^2 = \mathrm{Var}_{i\sim p_i}(e_i^{(n)})
\end{aligned}$$

and we recover the variance of the expected logit value.

Now we turn our attention to the entropy formula:

$$\begin{aligned}
H = -\sum_i p_i \log p_i &= -\sum_i p_i (\log \exp(-\beta e_i^{(n)}) - \log Z) \\
&= \sum_i p_i \log Z - \sum_j -\beta e_j^{(n)} p_j \\
&= \log Z + \beta \mathbb{E}_{i\sim p_i}(e_i^{(n)}) = \log Z - \beta \frac{d}{d\beta} \log Z
\end{aligned}$$

To check the monotonicity of H as β varies, we now take the derivative of this expression w.r.t. β:

$$\frac{dH}{d\beta} = \frac{d}{d\beta} \log Z - \frac{d}{d\beta} \log Z - \beta \frac{d^2}{d\beta^2} \log Z = -\beta \frac{d^2}{d\beta^2} \log Z = -\beta \mathrm{Var}_{i\sim p_i}(e_i^{(n)})$$

Since variance can never be negative, we find that $\frac{dH}{d\beta} \leq 0$ when $\beta \geq 0$, and $-\frac{dH}{d\beta} \leq 0$ when $\beta \leq 0$. As such, as the magnitude $|\beta|$ grows, the value of H must monotonically decrease. □

To supplement our proof in a way that clearly indicates the trends between the two quantites, we also provide—in Figure 5—a visualisation of how the Shannon entropy varies with temperature, for a 10-logit input with varying spread between the logits. While this figure clearly illustrates the expected trends, it is worth reflecting on its asymmetry.

Lastly, another effect of the temperature on entropy can be directly observed in our experimental framework—in Figure 6 we demonstrate the sharpening effect that applying adaptive temperature can have on the `softmax` coefficients of the sole attention head.

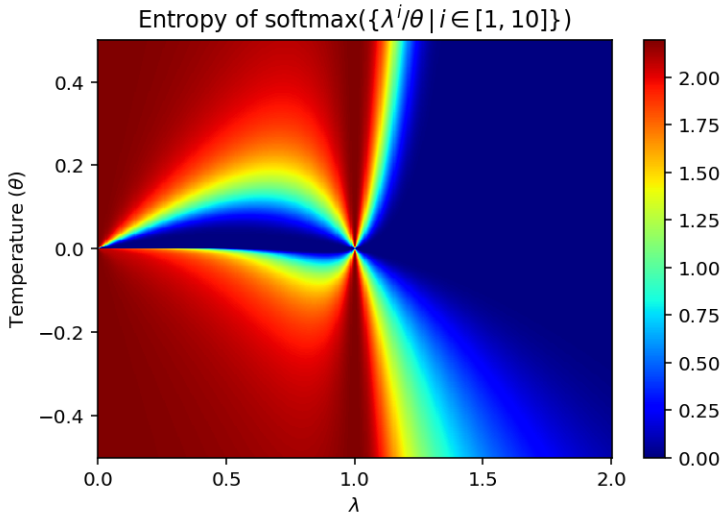

Figure 5: Entropy of the $\texttt{softmax}_\theta$ function for 10 elements of a power series. Entropy increases with temperature but the rate at which it increases is heavily dependent on the attention logit distribution. For the degenerate cases near the axes $\lambda = 0$ and $\lambda = 1$ all logits are the same an we have maximum entropy.

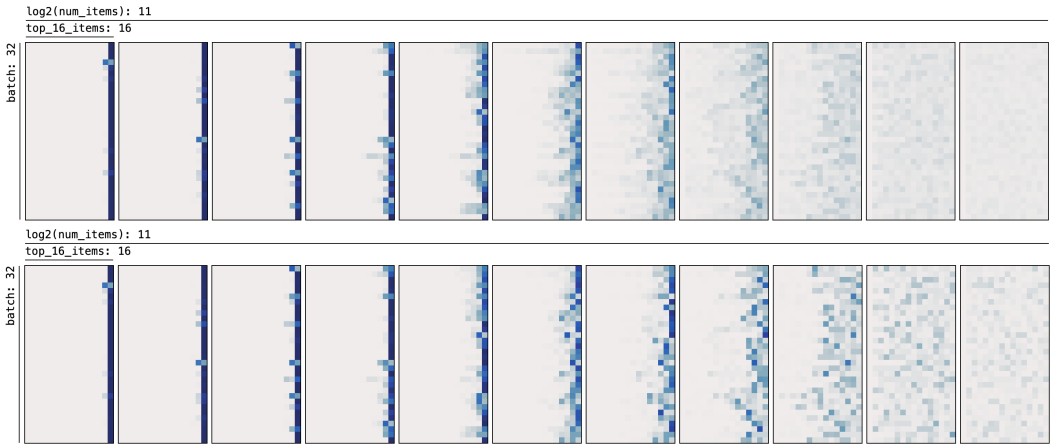

Figure 6: Visualising the attentional head for the `max` retrieval task with (**below**) and without (**above**) adaptive temperature applied, for the same batch and parameters as in Figure 2. Note the increased sharpness in the coefficients, especially as the amount of items increases.