MultiMax: Sparse and Multi-Modal Attention Learning

Yuxuan Zhou [1 2] **Mario Fritz** [2] **Margret Keuper** [1 3]

Abstract

SoftMax is a ubiquitous ingredient of modern machine learning algorithms. It maps an input vector onto a probability simplex and reweights the input by concentrating the probability mass at large entries. Yet, as a smooth approximation to the Argmax function, a significant amount of probability mass is distributed to other, residual entries, leading to poor interpretability and noise. Although sparsity can be achieved by a family of SoftMax variants, they often require an alternative loss function and do not preserve multi-modality. We show that this trade-off between multi-modality and sparsity limits the expressivity of SoftMax as well as its variants. We provide a solution to this tension between objectives by proposing a piece-wise differentiable function, termed MultiMax, which adaptively modulates the output distribution according to input entry range. Through comprehensive analysis and evaluation, we show that Multi-Max successfully produces a distribution that supresses irrelevant entries while preserving multi-modality, with benefits in image classification, language modeling and machine translation. The code is available at https://github.com/ZhouYuxuanYX/MultiMax.

1. Introduction

The SoftMax has remained in wide use in modern machine learning methods and finds its application in a variety of algorithms such as multi-class classification (LeCun et al., 2015; Goodfellow et al., 2016; Bishop & Nasrabadi, 2006), attention mechanisms (Vaswani et al., 2017; Veličković et al., 2017; Bahdanau et al., 2014; Gehring et al., 2016) and

[1]University of Mannheim, Germany [2]CISPA Helmholz Center for Information Security, Germany [3]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany. Correspondence to: Mario Fritz <fritz@cispa.de>, Margret Keuper <keuper@mpi-inf.mpg.de>.

reinforcement learning (Sutton & Barto, 2018; Rummery & Niranjan, 1994; Williams, 1992). It can be regarded as a differentiable approximation of the Argmax operation and projects the input onto the probability simplex, which allocates most of the probability mass to large entries. From the perspective of optimization, the SoftMax function allows for a reasonable trade-off between exploitation and exploration (White & Sofge, 1992), i.e., important positions are emphasized while every position has a chance of being explored. This trade-off can be controlled by a scale factor, which is often referred to as temperature.

However, the expressivity of SoftMax is severely limited by the following dilemma: a high temperature leads to over-smoothing and reduces the efficiency of the optimization, whereas a small temperature collapses multi-modality and makes training unstable. In attention layers for example, a small temperature will cause relevant positions except the peak to be overlooked, whereas a large temperature will "waste" a non-negligible portion of attention on irrelevant keys. Therefore, temperature is often set to one by default in attention layer. As shown later, such a compromise also results in the recently observed over-smoothing issue in both vision (Gong et al., 2021a; Wang et al., 2022c) and language (Shi et al., 2022) transformers. Moreover, transformer-based Large Language Models are shown to be prone to the interference of irrelevant context (Shi et al., 2023; Jia & Liang, 2017), which is also highly related to the portion of attention on irrelevant tokens (Weston & Sukhbaatar, 2023).

To overcome the issue, previous works have proposed sparse SoftMax alternatives, which allow to completely ignore small entries below a threshold. These sparse SoftMax variants have been studied in diverse contexts, e.g., generative modeling (Chen et al., 2021), output activations of multi-class classifiers, and/or attention mechanisms (Peters et al., 2019; Martins & Astudillo, 2016; Gupta et al., 2021).

However, such methods often suffer from poor gradient signal, which leads to instability during training. Moreover, the number of non-sparse dimensions is often treated as empirically selected hyperparameter.

In contrast to sparsity, multi-modality has been less discussed in the previous studies. Since attention is not supposed to be exclusive in most cases, the vanilla SoftMax, as an approximation of Argmax, does not easily comply
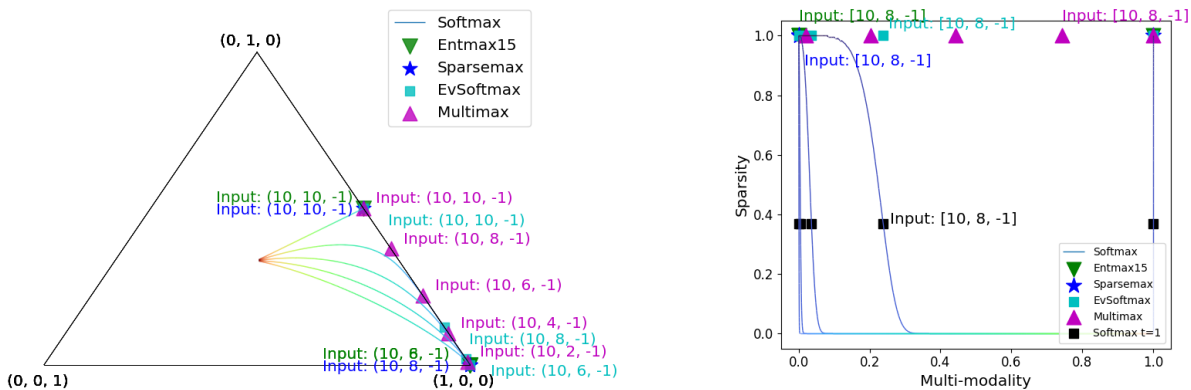
(a) SoftMax output depends on the temperature, which we show by the color coding from dark blue (low temperature) to red (high temperature). Sparse SoftMax variants collapse multi-modality, while MultiMax successfully produces approximately **sparse** and **multi-modal** distributions.

(b) SoftMax and its sparse extensions are limited by the trade-off between sparsity and multi-modality, which is improved by our MultiMax.

Figure 1. We evaluate SoftMax, SparseMax, EntMax, EvSoftMax and MultiMax (using the parameters of a hidden layer MultiMax trained on ImageNet directly) functions on a series of example input points $v \in \mathbb{R}^3$ and project the resulting distribution on a simplex Δ^2. Informally, the interior of the simplex stands for trimodal distributions, the edges constitute the set of bimodal distributions, and the vertices are unimodal distributions. Notably, the above figures highlight the advantage of MultiMax's multi-modality. EntMax, Sparsemax and SoftMax with small temperature (blue colored line) yield a (quasi) uni-modal distribution, which ignore the second largest entry. In contrary, SoftMax with higher temperatures (green and orange colored line) fails to ignore the negative entry.

with multi-modality. The sparse alternatives (Martins & Astudillo, 2016; Peters et al., 2019; Laha et al., 2018) to SoftMax have even a larger tendency to not preserve the multi-modality of distributions (Itkina et al., 2020).

In this paper, we propose MultiMax as an alternative to SoftMax. MultiMax allows for learning when to emphasize sparsity and when to emphasize multi-modality, offering a flexible trade-off between both. At the same time, it remains piecewise differentiable such as to allow for stable gradient-based optimization.

Specifically, MultiMax extends the traditional SoftMax by a preceding parameterized function that enables to learn distinct temperature values for particular input value ranges separately. Used within a self-attention mechanism, this facilitates for example to learn particularly low temperatures that induce sparsity for low input value ranges, i.e. unrelated tokens can be ignored, while learning high temperatures for higher input value ranges, i.e. several related tokens can share the attention in a multi-modal way. The improved multi-modality and sparsity brought by MultiMax is demonstrated in Figure 1. MultiMax is able to serve as a drop-in replacement of SoftMax in any applications and adapt to an appropriate form via training.

After a theoretic analysis, we show empirically that MultiMax can improve the attention mechanism and is an effective classifier output activation as well. MultiMax consistently improves over SoftMax baselines in a wide range of tasks, with an increase of 0.6% classification accuracy on ImageNet, an improve of 0.7 in perplexity for language

modeling on WikiText-103, and a gain of 0.3 in BLEU score for English to German translation on WISLT-2014.

The contributions of this paper are as follows:

- We generate insights in the trade-off between sparsity and multi-modality in SoftMax.

- We propose MultiMax – an alternative to SoftMax with better and learnable tradeoffs between both, multi-modality and sparsity.

- We show advantageous properties of MultiMax theoretically and demonstrate performance improvements on diverse tasks ranging from image classification over language modeling to machine translation.

2. Related Work

We organize the related work by first discussing related SoftMax alternatives afterwards more broadly approaches that have aimed to improve attention mechanism as well as prevent oversmoothing.

SoftMax alternatives. In previous work, huge efforts have been made to pursue sparsity. Sparsemax (Martins & Astudillo, 2016) and its generalization EntMax-α (Peters et al., 2019) are sparse SoftMax variants through thresholding the output probability. Although the hyperparameter α is supposed to control the degree of sparsity, the functions lack full support for $\alpha > 1$. Another variant, in principle similar to EntMax-1.5, with control of the sparsity is

Sparsehourglass (Laha et al., 2018). As output activation of a classifier, these approaches require alternative losses to enable gradient-based optimization. Yet, this can cause slow convergence and training instability as well as an additional approximation error. Ev-SoftMax (Chen et al., 2021) additionally reveals that these sparse SoftMax variants could harm multi-modality. It achieves sparsification by zeroing out input entries smaller than average and provides a training-time modification strategy to enable gradient-based training. This is indeed similar to the broadly adopted top-k selection of SoftMax output, e.g., in attention layers of vision (Wang et al., 2022b; Zhao et al., 2019) and language (Gupta et al., 2021) transformers. In contrast, our MultiMax achieves sparsity and improved multi-modality at the same time without extra hyperparameters. It has also full support and thus is a drop-in replacement of SoftMax in any context.

Anti-oversmoothing approaches. Over-smoothing refers to the issue that the representations of different tokens tend to become more similar as layer depth increases. This problem is observed in both vision (Wang et al., 2022c; Gong et al., 2021a) and language transformers (Shi et al., 2022). Patch Diversification (Wang et al., 2022c) combines three regularization losses to explicitly encourage diversity in patch representations. AttnScale (Wang et al., 2022c) decomposes a self-attention block into low-pass and high-pass components, and rescales the high-pass component of the self-attention matrix. While these remedies have been proposed, the reason behind lacks in-depth discussion. Notably, (Shi et al., 2022) has attempted an analysis by relating self-attention matrix to adjacent matrix of a graph. Their claim of post-normalization being the root cause has led to further discussion, as they stick to post-normalization in the end and pre-normalization empirically performs no better than post-normalization (He et al., 2020). We find that the over-smoothing problem is indeed is comparable to over-smoothing problem in GCNs (Chen et al., 2020; Oono & Suzuki, 2019), and strongly related to the inevitable amount of attention assigned to irrelevant tokens. The identity of each token degrades rapidly due to the repetitive attention operations. As shown in the studies of GCNs, sparsification (Rong et al., 2019; Hasanzadeh et al., 2020; Zheng et al., 2020) is a direct and effective solution.

Attention mechanism A vast amount of efforts have been invested in proposing new or improving the existing attention mechanisms (Vaswani et al., 2017; Veličković et al., 2017; Bahdanau et al., 2014; Gehring et al., 2016). (Kim et al., 2017) successfully incorporated richer structural distributions into attention networks via graph encodings. (Niculae & Blondel, 2017) introduced a new framework for sparse and structured attention with a smoothed max operator, which can be regarded as a generalization of softmax and sparsemax. (Deng et al., 2018) considered variational attention networks as alternatives to soft and hard atten-

Table 1. Classification accuracy on ImageNet1K using Deit-small baseline with Global Avarege Pooling (GAP) and classification token (CLS) respectively.

Model	Head	Temperature $\frac{1}{t}$					
		0.1	0.5	1	2	10	trainable
Deit-small	CLS	5.1	79.9	79.9	**80.0**	79.5	79.7
	GAP	4.7	80.3	**80.4**	80.0	79.9	80.2

tion for better learning latent variable alignment models. (Maruf et al., 2019) suggested to adopt sparse attention to selectively focus on relevant sentences in the document context for improved neural machine translation. (Zhang et al., 2020) explored the feasibility of specifying rule-based patterns to sparsify encoder outputs for improved decoding efficiency. While these approaches mainly focus on improving sparsity, our MultiMax improves both multi-modality and sparsity at the same time. Moreover, MultiMax is a universal alternative to the SoftMax function, which is not limited to the application in the attention mechanism.

3. Background, Metrics, and Analysis

In this section, we state the challenge of sparsity-multi-modality trade offs in reweighting functions such as softmax. Based on metrics to measure these quantities, we provide a theoretical analysis that shows the tension between those two goals in previous formulations.

3.1. Background

SoftMax is the most widely adopted **reweighting function** in machine learning and is formulated as follows:

Definition 3.1. Let $\Delta^{K-1} = \{\boldsymbol{p} \in \mathbb{R}^K_{\geq 0} | \mathbb{1}^T \boldsymbol{p} = 1\}$ be the $K-1$ dimensional simplex. SoftMax maps a vector $\boldsymbol{x} \in \mathbb{R}^K$ with $K \in \mathbb{Z}_+$ to a proper distribution in Δ^{K-1}:

$$\phi_{SoftMax}(\boldsymbol{x})_i = \frac{e^{tx_i}}{\sum_{k=1}^{K} e^{tx_k}}, \tag{1}$$

where $\frac{1}{t}$ controls the entropy of the generated distribution and is often referred to as "temperature". The exponential term makes the distribution concentrated on the largest entries, which reflects the selective nature of for example the attention mechanism or multi-class classification.

3.2. Sparsity and Multi-Modality Trade-off

Although sparsity seems to be easily acquired by decreasing the temperature, we find that the gain of increased sparsity comes at a cost in practice. We exemplify such an issue by comparing the classification performance of a transformer on ImageNet1K with different SoftMax temperatures in Table 1. As shown in the table, tuning temperature is tedious

and brings no obvious advantage. Moreover, a small temperature typically provides poor learning signal and can hamper training stability, as suggested by the low accuracy for temperature 0.1. For a better understanding of the inefficacy of temperature tuning, we follow-up with a brief theoretical study to show that the temperature tuning of SoftMax function is indeed limited by an inherent trade-off between sparsity and multi-modality.

To enable a precise analysis on the trade-off between multi-modality and sparsity, we need to define appropriate quantitative metrics for these two properties of reweighting functions.

3.2.1. QUANTIFYING MULTI-MODALITY AND SPARSITY OF REWEIGHTING FUNCTIONS

For multi-modality and sparsity, the probabilities close to peak and zero are with no doubt the most relevant, respectively. And such relevance equivalently transfers to the largest and smallest input entries, since the studied reweighting (activation) functions should be monotonically non-decreasing (Ganea et al., 2019; Gao & Pavel, 2017). For simplification, we omit the trivial case when two entries are equal, since they remain equal after any valid function.

To quantitatively compare the **multi-modality** of the distributions generated by different reweighting functions ϕ w.r.t. a given input $\boldsymbol{x}$, we propose the following metric $\mathcal{M}(\boldsymbol{x})$:

Definition 3.2. Without loss of generality, let x_{max} be the largest entry and $x_{max} > x_n > \epsilon$, where ϵ could be any reasonable threshold for a entry to be considered relevant and N is the counts of such entries. The **Multi-Modality Metric** is given by:

$$\mathcal{M}(\boldsymbol{x}) = 1 - \frac{1}{N} \sum_{\epsilon < x_n < x_{max}}^{N} (\phi(\boldsymbol{x})_{max} - \phi(\boldsymbol{x})_n), \quad (2)$$

Intuitively, this metric captures the average difference between the reweighted relevant entries $\phi(\boldsymbol{x})_n \ \forall x_n > \epsilon$ and the maximum $\phi(\boldsymbol{x})_{max}$. The average distance would be close to 0, if all output entries are about the same (maximum multi-modality). In order to make it a large=better metric, we subtract it from 1.

Analogously, we build a **Sparsity Metric** for the reweighting functions upon the common $-L_{\epsilon}^1$ sparsity metric for vectors (Hurley & Rickard, 2009), which calculates the negative sum of entries smaller than ϵ. Although sparse or non-sparse is a binary status, a smooth metric is desired to additionally consider values close to zero (i.e. approximately sparse). Moreover, we would like to take the non-linear nature of such sparsity into account, i.e., above a reasonably small threshold, a large portion of the range from 0 to 1 is supposed to be non-sparse. In this case, a non-linear scaling

(especially an approximation of a step function) helps to better reflect the actual degree of sparsity. Thus, we define the sparsity metric as follows:

Definition 3.3.

$$\mathcal{S}(\boldsymbol{x}) = \frac{1}{L} \sum_{x_l < \epsilon}^{L} \exp\left(\frac{s - \phi(\boldsymbol{x})_l}{s} - 1\right), \quad (3)$$

where $s \in [0, 1]$ can be any reference value for a non-linear scaling of the sparsity score and L is the counts of entries smaller than ϵ. For example, the probability of the smallest entry x_{min} after SoftMax ($\text{SoftMax}(\boldsymbol{x})_{min}$) can be chosen as a reasonable reference value. Together with the exponential term, $\mathcal{S}(\boldsymbol{x})$ results in a smooth approximation of a step function, with the output range normalized to $[0, 1]$, where larger values indicate stronger degrees of sparsity. Having defined the two metrics, we are able to prove there exists a trade-off between them.

3.2.2. PROOFING THE TRADE-OFF

Lemma 3.4. $\mathcal{S}(\boldsymbol{x})$ *is monotonically increasing w.r.t.* $\phi(\boldsymbol{x})_l$. *(See Appendix B.1 for the proof.)*

This can be easily proved by checking the partial derivative. Similar proof can be done for the following:

Proposition 3.5. *For a given input* $\boldsymbol{x}$, *the following statements hold w.r.t. temperature* t.

1. *Multi-modality of SoftMax is monotonically increasing.*

2. *Sparsity of SoftMax is monotonically decreasing for* $\epsilon \leq \frac{\|\boldsymbol{x}\|_1}{K}$.

(See Appendix B.2 for the proof.)

It is clear that we could increase either multi-modality or sparsity by simply varying temperature, but at the cost of decreasing the other. As a remedy, we suggest a piecewise modulation scheme, which modulates small and large entries via two corresponding temperatures independently.

4. MultiMax

Based on our insights in the trade-off between sparsity and multi-modality in SoftMax, we propose MultiMax that reconciles those two objectives in a learnable formulation. We start by defining MultiMax that introduces two temperature terms that control for sparsity and multi-modality respectively. We analyze improved properties that are achieved by this formulation and finally extend the concept to higher order polynomials and beyond attention mechanisms.

The following sections will provide a theoretic analysis of MultiMax, starting with its first-order form.

4.1. First-order MultiMax

Definition 4.1. Let b and d be two control parameters. We apply two corresponding temperatures t_b and t_d only to the entries smaller than b and larger than d, respectively. We construct a piece-wise linear function σ to modulate the SoftMax input x, which defines the proposed MultiMax:

$$\phi_{MultiMax}(\boldsymbol{x})_i = \frac{\exp\left(\sigma(x_i)\right)}{\sum_{k=1}^{K}\exp\left(\sigma(x_k)\right)}, \quad \text{where}$$

$$\sigma(x) = x + \underbrace{(1 - t_b)Max(b - x, 0)}_{\text{term}(1)} + \underbrace{(t_d - 1)Max(x - d, 0)}_{\text{term}(2)},$$

$$(4)$$

We call the above function the first-order MultiMax function and we will generalize it to a higher-order version towards the end of this section. For now, the first-order MultiMax has an intuitive interpretation:

$$\sigma(x) = \begin{cases} t_b x + (1 - t_b)b & x < b \\ x & b \le x \le d \ , \\ t_d x + (1 - t_d)d & x > d \end{cases} \quad (5)$$

where the bias terms $(1 - t_b)b$ and $(1 - t_d)d$ guarantees continuity of the modulator, e.g., $\lim\limits_{x \to b^-} \sigma(x) = \lim\limits_{x \to b^+} \sigma(x) = b$. To guarantee differentiability, subgradients can be defined for the turning points, e.g., $d\sigma(x)/dx = 1$ at $x = b$, please refer to (Boyd et al., 2003) for more details. For $t_b > 1$ and $0 < t_d < 1$, we could prove that MultiMax achieve a better balance between multi-modality and sparsity than SoftMax. Intuitively, a large t_b pushes small entries closer to zero, while a small t_d reduces the gap between large entries. Therefore, the output distribution is modulated to exhibit higher sparsity as well as multi-modality.

To disclose the mechanism behind, we first study the impact of modulating only the small entries on the output distribution. Then we show that additionally modulating the large entries increases multi-modality further.
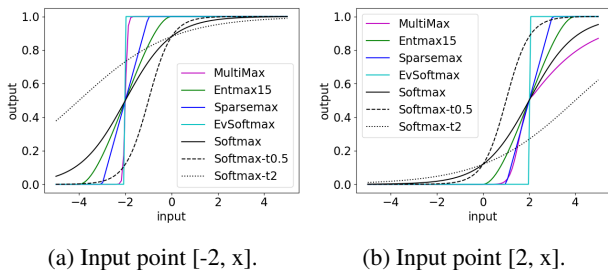

(a) Input point [-2, x].   (b) Input point [2, x].

Figure 2. Illustration of different reweighting functions in the two-dimensional case. It can be seen clearly that MultiMax weigh the entries at small and large value ranges in a different manner, thus it does not suffer from the trade-off between sparse and multi-modal.
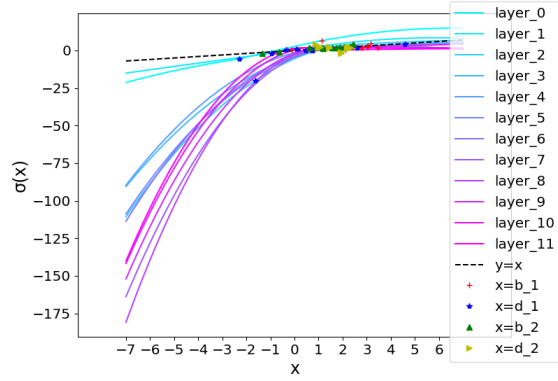

Figure 3. The learned modulator functions σ (Equation (6)) at each layer, comparing to identity mapping of the SoftMax input x (dashed black line). All layers except for the first two converge to a form that is consistent to our analysis, i.e., low temperature (steep slope) for small entries and high temperature (flat slope) for large entries.

4.2. Improved Pareto Efficiency

Improving sparsity With the above defined metrics, we show that adding term (1) alone (denoted by MultiMax-l), i.e., modulating smaller entries, already leads to a better *Pareto Optimality* (Buchanan, 1962) regarding sparsity and multi-modality than SoftMax.

Proposition 4.2. *The following properties hold for $t_b > 1$.*

1. *MultiMax-l generates sparser distribution than SoftMax with temperature 1.*

2. *MultiMax-l achieves better multi-modality than SoftMax with temperature 1.*

(See Appendix B.5 for the proof.)

From the above analysis, we could see that MultiMax-l has higher *Pareto Efficiency* than SoftMax: MultiMax-l with $t_b > 1$ has both better sparsity and multi-modality than Softmax with temperature 1 (Proposition 3.5), and Softmax can not improve both properties at the same time by changing temperature (Proposition 4.2).

Enhancing multi-modality further As shown in Proposition 4.3, including the modulation of larger entries further enhances multi-modality while retaining better sparsity than SoftMax.

Proposition 4.3. *The following properties hold for $t_d < 1$ and $t_b > 1$:*

1. *MultiMax can achieve better sparsity than SoftMax with temperature 1.*

2. MultiMax can achieve better multi-modality than MultiMax-l.

(See Appendix B.6 for the proof.)

4.3. Generalization

4.3.1. GENERALIZATION TO OTHER ACTIVATIONS

Piece-wise linear activation functions are widely adopted in modern machine learning algorithms, e.g., ReLU (Agarap, 2018), Leaky ReLU (Maas et al., 2013) and PReLU (He et al., 2015). Although MultiMax focuses on a different purpose, it can seen from Equation (4) that the modulator/rectifier function σ of MultiMax is a generalization of these activation functions. For example, if $b = d = 0$, $t_d = 1$ and $t_b = 0$, then σ is reduced to ReLU. For the rest, it can be shown easily in a similar way.

4.3.2. GENERALIZATION TO HIGHER-ORDER POLYNOMIALS

So far, it has been shown that higher *Pareto Efficiency* can be realized with a piece-wise linear modulation function, which belongs to the family of first-order polynomials. To obtain smoother transitions at turning points and larger capacity, second-order terms are included in our final formulation of MultiMax:

$$\sigma(x) = x + \sum_{n=1}^{N} \underbrace{(1 - t_{b_n})Max(b_n - x, 0)^n}_{term(1)} + \underbrace{(t_{d_n} - 1)Max(x - d_n, 0)^n}_{term(2)},$$
(6)

where n ranges from 1 to 2. We don't include higher orders beyond the second, because it proves to be sufficient in practice. We show in the ablation Section 5.3 that the extra nonlinearities brought by the second-order terms benefit the learning of the modulation scheme, in analogy to the previous study on activation functions (Hendrycks & Gimpel, 2016; Clevert et al., 2015; Elfwing et al., 2018).

As shown in Figure 1b, the output of SoftMax with varied temperatures forms a trajectory and converges to sparsemax as temperature approaches 0. EntMax-α stays close to the trajectory with $\alpha = 1.5$, and is indeed equivalent to softmax or SparseMax when $\alpha = 1$ or 2. MultiMax achieves, in the example, an otherwise non-reachable trade-off, with values close to the simplex that vary in two out of three possible modes. For a less complex illustration, we also provide the comparison with other reweighting functions with 2D inputs in Figure 2, in which case SoftMax is equivalent to Sigmoid. While other approaches handle small and large entries equally, MultiMax provides an input-adaptive reweigthing scheme.

We show in Figure 3 the learned modulator function of deit-small on ImageNet and compare it to the original input

x (dashed black line) when used in attention layers. The learned functions at most layers (except the first two) conforms to our analysis: steeper slope for small entries (below the dashed black line on the left side means temperature smaller than 1) and flatter slope for large entries (below the dashed black line on the right side means temperature larger than 1). This conforms to our theoretical analysis that small entries should be suppressed with smaller temperature and large entries should be pushed closer with large temperature. Moreover, it is noteworthy that the need for sparsity increases as the layer goes deeper, according to the learned curves.

4.3.3. GENERALIZATION BEYOND ATTENTION

As shown in the above analysis, the proposed MultiMax not only generalizes SoftMax, but also achieves a better Pareto optimality w.r.t. sparsity and multi-modality with appropriate parameterization. Due to its fully parameterized formulation, it is learnable and adaptable to any scenario where a reweighting function is required. Since the need for the degree of multi-modality and sparsity may vary among different applications, we do not explicitly constrain any of the parameters and optimize them jointly with the model.

4.4. Computational Efficiency

The extra computation of MultiMax is negligible for modern machine learning algorithms: As shown in Equation (4), the total amount of additional parameters for a 12 layer Transformer with 2nd-order MultiMax is just $8 \times 12 = 96$, because each order only contains 4 parameters, including t_b, t_d, b and d. Moreover, the modulation function $\sigma(x)$ merely consists of cheap element-wise operations, i.e., multiplication with t_b and t_d, subtraction with b and d, two Max operations, addition of the two terms at each order as well as a residual addition. Thus a second-order MultiMax requires $7 \times 2 + 1 = 15$ extra Floating Point Operations (FLOPs) for a univariant input. For Deit-small model with input length of 256, hidden dimension of 384 and 12 layers, replacing MultiMax with SoftMax in all attention layers leads to 0.0168G extra FLOPs, i.e. only 0.37% of the original model's 4.6G FLOPs.

In practice, customized layers often run much slower than the highly optimized built-in Pytorch layers. The performance gap between theory and practice is mainly because the PyTorch framework is eagerly evaluated and thus brings additional memory access time and kernel launch time, please refer to this page [1] for more details. Thus a native Pytorch implementation of MultiMax increases the training time of Deit-small on ImageNet by about 40% (0.19 s/iteration vs 0.26 s/iteration), while the increase in infer-

---

[1]https://residentmario.github.io/
pytorch-training-performance-guide/jit.html

Table 2. Comparing to Deit (Touvron et al., 2021a) baseline and anti-over-smoothing methods on ImageNet-1k by replacing SoftMax with MultiMax in the attention and/or output layers. * denotes that results are not strictly comparable: these methods rely on a different training setup. For example, additional training epochs are adopted by both works, talking-head (Shazeer et al., 2020) and a higher drop-path (Huang et al., 2016) rate are applied together with Patch Diversification.

| Model | Method | Parameters | Epochs | Modulation | | Acc. (%) |
				Output	Attention	
Deit-tiny	SoftMax	5M	300	N/A	N/A	72.8
	MultiMax		300	✓	✓	**73.4**
Deit-small	Softmax		300	N/A	N/A	80.4
	Top-k (Wang et al., 2022b)		300	✓	N/A	80.6
	Ev-SoftMax (Chen et al., 2021)	22M	300	-	✓	80.0
			300	✓	-	80.7
	MultiMax		300	-	✓	80.7
			300	✓	✓	**81.0**
Deit-base	SoftMax	86M	300	N/A	N/A	82.1
	MultiMax		300	✓	✓	**82.6**
Deit-small	Patch Diversification (Gong et al., 2021b)		400	N/A	N/A	81.2*
	AttnScale (Wang et al., 2022c)		500	✓	N/A	80.9*
	MultiMax		400	✓	✓	81.2
			500	✓	✓	**81.3**

ence time is negligible (less than 2%). However, we are able to achieve a reduction from 40% (native Pytorch implementation) to only about 10% increase of training time (0.21 s/iteration) by implementing the *Max* operator with 0 as built-in ReLU function and applying torch.jit.script decorator to fuse the remaining elementwise operations of our MultiMax, following the documentation [2]. Notably, a fully optimized implementation of MultiMax in C++ or CUDA as done with Pytorch built-in layers might further reduce the gap.

5. Experiments

In this section, we replace SoftMax with MultiMax in different baselines and apply them to the corresponding tasks, including image classification on ImageNet1K, langauge modeling on Wiki-Text-103 corpus and machine translation on IWSLT-2014 corpus. Experimental results demonstrate consistent improvement with MultiMax, without any extra changes, e.g. hyperparameters or architecture. Moreover, we provide additional insights and demonstrate that advantageous properties, including reduced over-smoothing (Section 5.2.1) and improved sparsity & multi-modality (Section 5.2.2), are achieved.

5.1. Benchmarking

5.1.1. IMAGENET1K CLASSIFICATION

For classification, we train the widely adopted Deit (Touvron et al., 2021a) from scratch on ImageNet1K as baseline. Following the same training setup, we train Deit by only replacing the SoftMax function with our MultiMax, in the attention layers and/or output layer for a fair comparison.

[2] https://pytorch.org/tutorials/recipes/recipes/tuning_guide.html

For training, we closely follow the training settings provided in (Touvron et al., 2021a) and train all the models for 300 epochs. Following the more recent works (Chu et al., 2021; Liu et al., 2021), we also adopt Global Average Pooling (GAP) instead of using Class Token (CLT) as classification head. While class token causes discrepancy in attention (Touvron et al., 2021b) and breaks translation invariance (Chu et al., 2021), GAP avoids this problem and improves the accuracy.

The results in Table 2 show a consistent improvement by using MultiMax for both attention and output activation layers. Although those sparse SoftMax variants work well for Machine Translation tasks, most of them have issues with Deit models. Ev-SoftMax decreases the performance when used in attention layers and the training does not converge (accuracy below 10%) when used in the output layer. For the inferior performance of Ev-SoftMax, we hypothesize that less sparsity is required for the attention among image patches than for language tokens, and zeroing out the entries smaller than average might be too aggressive. For the unstable training, their simple training-time modification might not be sufficient. The alternative losses provided by Sparse SoftMax and EntMax-1.5 require integer labels, thus are not compatible with the widely adopted label smoothing technique in vision transformers. Training instability issues are also encountered when using SparseMax in attention layers only. Therefore, we excluded them for the image classification task.

5.1.2. LANGUAGE MODELING

We test the effectiveness of our MultiMax further on the Language Modeling task on WikiText-103 (Merity et al., 2016) using a 6-layer Transformer Decoder with 156M parameters. The implementation is based on the official fairseq repos-

itory[3] and the training setup is kept as default, i.e., $5e{-}4$ learning rate with a maximum of 2048 tokens per GPU for 50k iterations on 4 GPUs. The results of the baseline transformer using SoftMax attention and our MultiMax are shown in Table 3. We again observe a consistent improvement by applying MultiMax in the output activation for this task.

Table 3. Evaluation of the performance on WikiText-103 language modeling task by test perplexity.

Method	Attention	Output	Perplexity ↓
SoftMax	-	-	29.4
Top-k (Gupta et al., 2021)	✓	N/A	29.1
MultiMax	✓	-	29.0
	✓	✓	**28.7**

Table 4. Comparing to other SoftMax variants using two different baseline settings (see Section 5.1.3 for more details) on IWSLT 2014 English to German Translation task.

SoftMax	SparseMax	EntMax-1.5	EvSoftMax	MultiMax
34.4 ± 0.07	28.7 ± 0.16	34.6±0.09	34.7 ± 0.06	34.7 ± 0.07

5.1.3. MACHINE TRANSLATION

Following previous approaches, we also evaluate our method on the task of machine translation. We train a 38M 12-layer Transformer baseline with encoder-decoder (6 layers each) architecture (Vaswani et al., 2017) from scratch on the IWSLT2014 German to English dataset (Cettolo et al., 2017), following the training setup provided in the fairseq repository (Footnote 3). Under the same setting, we also train the transformer with our MultiMax in replacement of SoftMax in the attention layers, following the common setup in previous work. The single best checkpoint and a beam size of 5 is adopted. The detokenized SacreBLEU (Post, 2018) scores (mean and standard deviation) of 3 runs are compared in Table 4. MultiMax performs on par with EvSoftMax and is slightly better than EntMax-1.5 for this task.

5.2. Empirical Studies and Insights

In this section, we empirically verify the positive impact of MultiMax on the over-smoothing issue, as well as the improvement on multi-modality and sparsity in the attention scores of Deit-small trained on ImageNet1K.

5.2.1. ANALYSIS ON OVER-SMOOTHING

To validate the efficacy of our MultiMax on preventing over-smoothing, we adopt the *Patch Similarity* (Gong et al., 2021b) or *Mean Average Distance* (MAD) (Chen et al., 2020) metric to compare transformers using SoftMax and
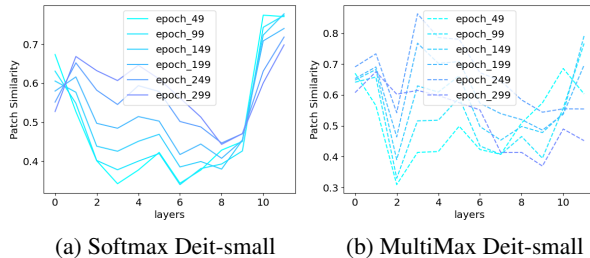
(a) Softmax Deit-small   (b) MultiMax Deit-small

Figure 4. Patch similarities for each layer and at different epochs. Darker color denotes the patch similarities at a larger training epoch.

MultiMax on ImageNet1K. The numbers are shown in Figure 4. It can be observed that patch similarity increases as the depth grows for SoftMax attention during the entire training, whereas the patch similarity converges to a much lower level for MultiMax attention in deeper layers. We attribute this to the undesirable amount of attention assigned to irrelevant tokens which contributes the over-smoothing issue in Transformers. Moreover, it also showcases the flexibility of MultiMax's parameterized formulation, which can encourage exploration in the early stage and shift the distribution gradually towards higher sparsity as the training progresses. We have also examined the increased discrepancy between single layer attention and accumulated roll-out attention (Abnar & Zuidema, 2020), which further indicates the strong connection between non-sparse SoftMax attention and the over-smoothing issue. Please refer to Appendix C.3 for more details.

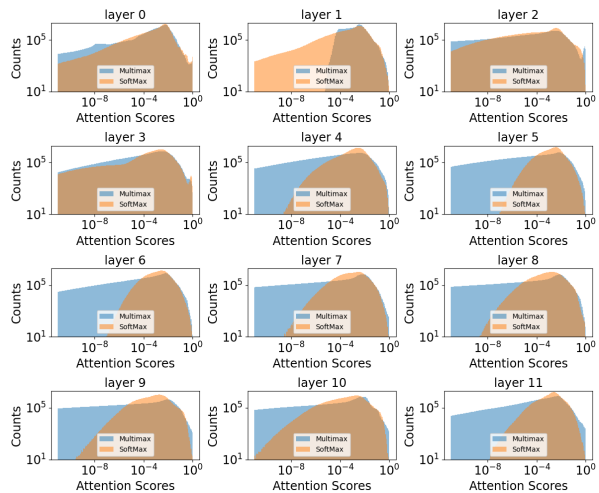### 5.2.2. ANALYSIS ON SPARSITY AND MULTI-MODALITY

Figure 5. Histograms of the attention scores at each layer. Multi-Max attention is distributed towards both ends: small scores are pushed closer to zero and more scores lie above 0.1.

In this section, we empirically evaluate the impact of using our MultiMax on the sparsity of attention scores. To achieve this, we evaluate the trained model on 1000 images and collect the attention scores at each layer.

As shown in Figure 5 in a log-log histogram, the attention scores of MultiMax are distributed more towards both ends of the score range, i.e., extremely small values near zero and large values between 0.1 and 1. In comparison, the attention scores of SoftMax are concentrated in the region in between, which corresponds to the bumps in the figure. Note that the number of counts are drawn at logarithmic scale, thus a small bump indeed indicates a large amount of counts. Notably, MultiMax attention behaves differently in the first two layers, which actually shows the flexibility of learning: the need for multi-modality or sparsity varies with varying context. Thus it can be a disadvantage to manually define the trade-off in advance. We also visualize the cumulative distribution of these attention scores in Appendix C.2, which also indicates a stronger sparsity achieved by MultiMax.

5.3. Ablation

To study the effect of each design component of our MultiMax independently, we conduct experiments using Deit-small as the baseline on ImageNet1K for ablation, as shown in Table 5. Since the language modeling and image classification tasks are computationally heavy, we report the result of a single run with the seed unchanged for all these experiments, as commonly done for ImageNet models.

Table 5. Impact of each MultiMax component.

Config	term (1)	term (2)	second order	Acc
1	-	-	-	80.4
2	✓	-	-	80.6
3	✓	✓	-	80.7
4	✓	✓	✓	81.0

To further validate the statistical significance of these results, we additionally conduct experiments using Deit-small with GAP on ImageNet1K and the results are recorded in Table 6. Comparing to the relatively small standard deviation, the improvement of using MultiMax is reliable.

Table 6. Multiple runs with random seeds using Deit-small on ImageNet1k. MultiMax shows consistent improvement over SoftMax.

Method	Runs			Mean	Std
	1	2	3		
SoftMax	80.4	80.3	80.3	80.3	0.05
MultiMax	**81.0**	**80.8**	**80.7**	**80.8**	0.12

5.4. Attention Visualization

As Transformer models (Vaswani et al., 2017; Liu et al., 2021; Zhou et al., 2022a;b; Wang et al., 2022a) stack a number of attention layers and aggregates the information repetitively, the attention scores at a single layer do not reflect the true information flow. To evaluate the impact on the classification more directly, we employ the well-established Grad-CAM (Selvaraju et al., 2017) to qualitatively evaluate the impact on the model's decision making. We additionally provide single layer attention scores in Appendix C.1 for reference.
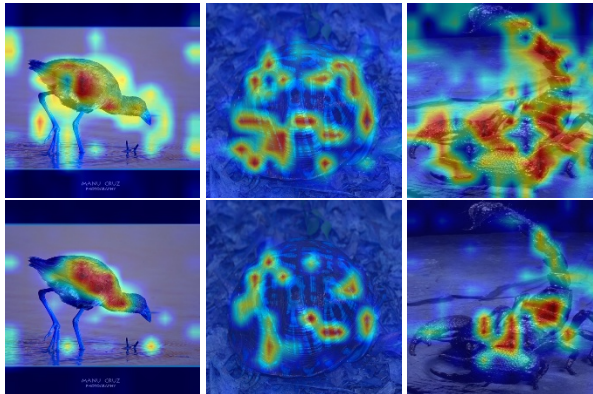

Figure 6. Grad-CAM of Deit-small using SoftMax (top row) and MultiMax (bottom row). The MultiMax attention maps are better localized on the objects and are close to zero in most background regions, indicating sparsity at the attention level.

6. Conclusion

In this paper, we formalize, analyze, and evaluate the sparsity and multi-modality trade-off of SoftMax and proposed MultiMax as a remedy for tension between these two desirable objectives. Through both experimental evaluation and analysis, we validated that MultiMax successfully learns to achieve higher multi-modality and sparsity at the same time. Although we have already demonstrated the benefits of MultiMax in attention layers and output activation of a classifier and a generative model across a wide range of tasks, we believe it has an even broader range of applications, such as in value networks and policy gradient for reinforcement learning as well as the learning of categorical distributions with Gumbel Softmax (Jang et al., 2016).

Impact Statement

This paper contributes to the understanding of core ML/AI methodology and improves the performances on a range of tasks that are broadly used as benchmark datasets in the field. Therefore, no negative impact that would be specific to our method is foreseeable at this point and we rather expect an overall positive impact by contributing the knowledge and understanding of these method that makes them more reliable.

Acknowledgements

References

Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

Agarap, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*. Springer, 2006.

Boyd, S., Xiao, L., and Mutapcic, A. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.

Buchanan, J. M. The relevance of pareto optimality. *Journal of conflict resolution*, 1962.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. Overview of the iwslt 2017 evaluation campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation*, 2017.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

Chen, P., Itkina, M., Senanayake, R., and Kochenderfer, M. J. Evidential softmax for sparse multimodal distributions in deep generative models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., and Shen, C. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. Latent alignment and variational attention. *Advances in eural information processing systems (NeurIPS)*, 2018.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 2018.

Ganea, O., Gelly, S., Bécigneul, G., and Severyn, A. Breaking the softmax bottleneck via learnable monotonic pointwise non-linearities. *International Conference on Machine Learning (ICML)*, 2019.

Gao, B. and Pavel, L. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Gehring, J., Auli, M., Grangier, D., and Dauphin, Y. N. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*, 2016.

Gong, C., Wang, D., Li, M., Chandra, V., and Liu, Q. Improve vision transformers training by suppressing oversmoothing. *arXiv preprint arXiv:2104.12753*, 2021a.

Gong, C., Wang, D., Li, M., Chandra, V., and Liu, Q. Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753*, 2021b.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*. MIT Press, 2016.

Gupta, A., Dar, G., Goodman, S., Ciprut, D., and Berant, J. Memory-efficient transformers via top-k attention. *arXiv preprint arXiv:2106.06899*, 2021.

Hasanzadeh, A., Hajiramezanali, E., Boluki, S., Zhou, M., Duffield, N., Narayanan, K., and Qian, X. Bayesian graph neural networks with adaptive connection sampling. *International Conference on Machine Learning (ICML)*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

He, R., Ravula, A., Kanagal, B., and Ainslie, J. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *European Conference on Computer Vision (ECCV)*, 2016.

Hurley, N. and Rickard, S. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 2009.

Itkina, M., Ivanovic, B., Senanayake, R., Kochenderfer, M. J., and Pavone, M. Evidential sparsification of multimodal latent spaces in conditional variational autoencoders. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

Kim, Y., Denton, C., Hoang, L., and Rush, A. M. Structured attention networks. *International Conference on Learning Representations (ICLR)*, 2017.

Laha, A., Chemmengath, S. A., Agrawal, P., Khapra, M., Sankaranarayanan, K., and Ramaswamy, H. G. On controllable sparse alternatives to softmax. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 2015.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

Martins, A. and Astudillo, R. From softmax to sparsemax a sparse model of attention and multilabel classification. *International Conference on Machine Learning (ICML)*, 2016.

Maruf, S., Martins, A. F., and Haffari, G. Selective attention for context-aware neural machine translation. *arXiv preprint arXiv:1903.08788*, 2019.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Niculae, V. and Blondel, M. A regularized framework for sparse and structured neural attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Peters, B., Niculae, V., and Martins, A. F. Sparse sequence-to-sequence models. *arXiv preprint arXiv:1905.05702*, 2019.

Post, M. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

Rong, Y., Huang, W., Xu, T., and Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Rummery, G. A. and Niranjan, M. *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

Schirrmeister, R., Zhou, Y., Ball, T., and Zhang, D. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Conference on Machine Learning (ICML)*, 2017.

Shazeer, N., Lan, Z., Cheng, Y., Ding, N., and Hou, L. Talking-heads attention. *arXiv preprint arXiv:2003.02436*, 2020.

Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E. H., Schärli, N., and Zhou, D. Large language models can be easily distracted by irrelevant context. 2023.

Shi, H., Gao, J., Xu, H., Liang, X., Li, Z., Kong, L., Lee, S., and Kwok, J. T. Revisiting over-smoothing in bert from the perspective of graph. *arXiv preprint arXiv:2202.08625*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. *International Conference on Machine Learning (ICML)*, 2021a.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, H., Du, Y., Zhang, Y., Li, S., and Zhang, L. One-stage visual relationship referring with transformers and adaptive message passing. *IEEE Transactions on Image Processing*, 2022a.

Wang, P., Wang, X., Wang, F., Lin, M., Chang, S., Li, H., and Jin, R. Kvt: k-nn attention for boosting vision transformers. In *European Conference on Computer Vision (ECCV)*, 2022b.

Wang, P., Zheng, W., Chen, T., and Wang, Z. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022c.

Weston, J. and Sukhbaatar, S. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*, 2023.

White, D. A. and Sofge, D. A. The role of exploration in learning control. *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, 1992.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

Zhang, B., Titov, I., and Sennrich, R. On sparsifying encoder outputs in sequence-to-sequence models. *arXiv preprint arXiv:2004.11854*, 2020.

Zhao, G., Lin, J., Zhang, Z., Ren, X., Su, Q., and Sun, X. Explicit sparse transformer: Concentrated attention through explicit selection. *arXiv preprint arXiv:1912.11637*, 2019.

Zheng, C., Zong, B., Cheng, W., Song, D., Ni, J., Yu, W., Chen, H., and Wang, W. Robust graph representation learning via neural sparsification. *International Conference on Machine Learning (ICML)*, 2020.

Zhou, Y., Cheng, Z.-Q., Li, C., Fang, Y., Geng, Y., Xie, X., and Keuper, M. Hypergraph transformer for skeleton-based action recognition. *arXiv preprint arXiv:2211.09590*, 2022a.

Zhou, Y., Xiang, W., Li, C., Wang, B., Wei, X., Zhang, L., Keuper, M., and Hua, X. Sp-vit: Learning 2d spatial priors for vision transformers. In *33rd British Machine Vision Conference*, 2022b.

A. Lemmas

Lemma A.1. *The following inequalities hold:*

$$x_i > tx_i + (1-t)b, \quad \forall x_i < b \quad and \quad \forall t > 1$$
$$x_i < tx_i + (1-t)b, \quad \forall x_i > b \quad and \quad \forall t > 1$$
$$x_i < tx_i + (1-t)b, \quad \forall x_i < b \quad and \quad \forall t < 1$$
$$x_i > tx_i + (1-t)b, \quad \forall x_i > b \quad and \quad \forall t < 1$$

(See Appendix B.3 for the proof.)

Lemma A.2. *The following inequality holds* $\forall \epsilon \leq \frac{1}{L}(\sum\limits_{x_l<b}^{L} X_l - lnL)$ *and* $\forall t > 1$:

$$\sum_{x_l<b}^{L} e^{t(x_l-x_i)} \geq \sum_{x_l<b}^{L} e^{x_l-x_i}$$

(See Appendix B.4 for the proof.)

B. Proofs

B.1. Proof of Lemma 3.4

$\frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi(\boldsymbol{x})_l} = \frac{1}{1-s}\exp\left(\frac{s-\phi(\boldsymbol{x})_l}{s}-1\right). \quad \forall s < 1 \Rightarrow \frac{1}{1-s} > 0.$ Since the exponential term is always positive, we have $\frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi(\boldsymbol{x})_l} > 0, \quad \forall \phi(\boldsymbol{x})_l.$

B.2. Proof of Proposition 3.5

Proof. Statement 1

from Equation (1) and Definition 3.2

$$\frac{\partial \mathcal{M}(\boldsymbol{x})}{\partial t} = \frac{(x_{max}-x_n)e^{\frac{x_n-x_{max}}{t}}}{t^2 \sum_{k=1}^{K} e^{\frac{x_k-x_{max}}{t}}}$$
$$+ \frac{(1-e^{\frac{x_n-x_{max}}{t}})\sum_{k=1}^{K}\frac{x_{max}-x_k}{t^2}e^{\frac{x_k-x_{max}}{t}}}{(\sum_{k=1}^{K} e^{\frac{x_k-x_{max}}{t}})^2}$$

since $x_n - x_{max} < 0$, we have $0 < e^{\frac{x_n-x_{max}}{t}} < 1$

$$\Rightarrow \quad \frac{\partial \mathcal{M}(\boldsymbol{x})}{\partial t} > 0 \quad \text{holds} \quad \forall t \qquad \square$$

Proof. Statement 2

from Equation (1)

$$\frac{\partial \phi(\boldsymbol{x})_l}{\partial t} = \frac{\sum_{k=1}^{K}(x_k-x_l)e^{\frac{x_k-x_l}{t}}}{t^2(\sum_{k=1}^{K} e^{\frac{x_k-x_l}{t}})^2}$$

from Chebyshev's sum inequality

$$\sum_{k=1}^{K}(x_k-x_l)e^{\frac{x_k-x_l}{t}} > \frac{1}{K}\sum_{k=1}^{K}(x_k-x_l)\sum_{k=1}^{K} e^{\frac{x_k-x_l}{t}}$$

since $x_l < \epsilon \leq \frac{\|\boldsymbol{x}\|_1}{K}$, we have $\sum_{k=1}^{K}(x_k-x_l) \geq 0$

$$\Rightarrow \quad \frac{\partial \phi(\boldsymbol{x})_l}{\partial t} > 0$$

from Lemma 3.4

$$\Rightarrow \quad \frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial t} = \frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi(\boldsymbol{x})_l}\frac{\partial \phi(\boldsymbol{x})_l}{\partial t} > 0 \qquad \square$$

B.3. Proof of Lemma A.1

From basic laws of algebra, $x-tx-(1-t)b = (1-t)(x-b)$. For $t > 1$ and $x < b$, we have $(1-t)(x-b) > 0 \Rightarrow x > tx + (1-t)b$, and vice versa.

B.4. Proof of Lemma A.2

since $e^{x_l} > 0$, from Hoelder's inequality, we have

$$\sum_{x_l<b}^{L} e^{x_l-x_i} = \sum_{x_l<b}^{L} \left|e^{x_l-x_i}\right|^1 \cdot 1$$
$$\leq \sum_{x_l<b}^{L}\left((|e^{x_l-x_i}|)^t\right)^{\frac{1}{t}} \cdot \left(\sum_{l=1}^{L} 1^{\frac{t}{t-1}}\right)^{1-\frac{1}{t}}$$

raise both sides to the power of t and multiply by L^{1-t}

$$\Rightarrow L^{1-t}(\sum_{x_l<b}^{L} e^{(x_l-x_i)})^t \leq \sum_{x_l<b}^{L} e^{t(x_l-x_i)}$$

the above inequality holds if

$$\sum_{x_l<b}^{L} e^{x_l-x_i} \leq L^{1-t}(\sum_{x_l<b}^{L} e^{(x_l-x_i)})^t$$

take the natural log on both sides

$$ln\sum_{x_l<b}^{L} e^{x_l-x_i} \leq (1-t)lnL + tln\sum_{x_l<b}^{L} e^{x_l-x_i}$$

$$\Rightarrow lnL \leq ln\sum_{x_l<b}^{L} e^{(x_l-x_i)}$$

since e^x is convex and $x_i < \epsilon$

$$\sum_{x_l<b}^{L} e^{(x_l-x_i)} \geq e^{\sum\limits_{x_l<b}^{L}(x_l-x_i)} \geq e^{\sum\limits_{x_l<b}^{L}(x_l-\epsilon)}$$

the condition is satisfied for $\epsilon \leq \frac{1}{L}(\sum\limits_{x_l<b}^{L} x_l - lnL)$

B.5. Proof of Proposition 4.2

When only term (1) is considered, Equation (5) is reduced to:

$$\sigma(x) = \begin{cases} t_b x + (1-t_b)b & x < b \\ x & x \geq b \end{cases}, \tag{7}$$

and we obtain:

$$\phi_{\textit{MultiMax-l}}(\boldsymbol{x})_i = $$

$$\begin{cases} \dfrac{e^{t_b x_l + (1-t_b)b}}{\sum_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b} + \sum_{x_n \geq b}^{N} e^{x_n}} & x_l < b \\[4mm] \dfrac{e^{x_n}}{\sum_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b} + \sum_{x_n \geq b}^{N} e^{x_n}} & x_l \geq b \end{cases}, \tag{8}$$

where L and N denote the number of entries smaller than or greater than b and $L + N = K$.

Proof. Statement 1

from Equation (8), $\forall x_i < \epsilon \leq b$, eliminate the numerator

$$\phi_{\textit{MultiMax-l}}(\boldsymbol{x})_i = \frac{1}{\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum\limits_{x_n \geq b}^{N} e^{x_n - (t_b x_i + (1-t_b)b)}}$$

substitute $t_b x_i + (1 - t_b)b$ with x_i at lower right and $\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)}$ at lower left, from Lemma A.1 and Lemma A.2

$$\leq \frac{1}{\sum\limits_{x_l<b}^{L} e^{x_l - x_i} + \sum\limits_{x_n \geq b}^{N} e^{x_n - x_i}}$$

$$\Rightarrow \phi_{\textit{MultiMax-l}}(\boldsymbol{x})_i < \phi_{\textit{SoftMax}}(\boldsymbol{x})_i \qquad \square$$

Proof. Statement 2

Eliminate e^{x_i}, from Equation (8), $\forall x_i > x_j > b$

$$m_{\textit{MultiMax-l}} = 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b - x_i} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}}$$

substitute $(1 - t_b)b - x_i$ with $-t_b x_i$, from Lemma A.1

$$> 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}}$$

substitute $\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)}$ with $\sum\limits_{x_l<b}^{L} e^{x_l - x_i}$, from

Lemma A.2 $\quad \forall \epsilon \leq \frac{1}{L}(\sum\limits_{x_l<b}^{L} x_l - lnL)$

$$> 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{x_l - x_i} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}} = \mathcal{M}_{\textit{SoftMax}} \square$$

B.6. Proof of Proposition 4.3

Combined Equation (5) with SoftMax, we obtain:

$$\phi_{\textit{MultiMax}}(\boldsymbol{x})_i = $$

$$\begin{cases} \dfrac{e^{t_b x_i + (1-t_b)b}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & x_i < b \\[6mm] \dfrac{e^{x_i}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & b \leq x_i \leq d \\[6mm] \dfrac{e^{t_d x_i + (1-t_d)d}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & x_i > d \end{cases},$$

$$\tag{9}$$

where L, M and N denote the number of entries belonging to different ranges and $L + M + N = K$.

Proof. Statement 1

from Equation (9), $\forall x_i < \epsilon$, eliminate the numerator, then substitute $x_i + (1 - t_b)b$ with $t_b x_i$, from Lemma A.1

$$< 1/(\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m - x_i}$$

$$+ \sum\limits_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b})$$

from Lemma A.2, if $\epsilon \leq \frac{1}{M}(\sum\limits_{x_m<b}^{M} X_m - lnM)$

$$< 1/(\sum\limits_{x_l<b}^{L} e^{x_l - x_i} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m - x_i}$$

$$+ \sum\limits_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b})$$

if $\sum\limits_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b} > \sum\limits_{x_n>d}^{N} e^{x_n - x_i}$

$$\Rightarrow \phi_{\textit{MultiMax}}(\boldsymbol{x})_i < \phi_{\textit{SoftMax}}(\boldsymbol{x})_i$$

This is satisfied when $t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b > x_n - x_i$ holds $\forall x_n$, which can be reduced to

$$x_i < b - \frac{1 - t_d}{t_b - 1}(x_n - d)$$

where $x_n \geq d$, $t_d < 1$ and $t_b > 1$, and this is satisfied for

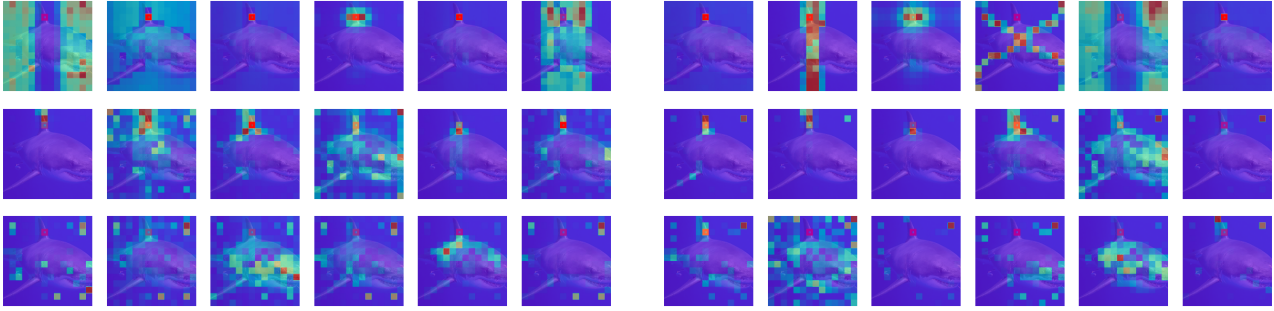$$\Rightarrow \epsilon \leq b - \frac{1 - t_d}{t_b - 1}(x_n - d) \qquad \square$$

Figure 7. Attention scores of SoftMax (left) and MultiMax(right) at the input and hidden layers (1st, 5th and 10th) w.r.t query 34. The query lies on the shark fin and is marked with red square. We see, from left to right, are attention scores of 6 heads for each method, where blue refers to low attention score and red indicates a high attention score. MultiMax attention is better localized while allowing for multiple modes.

Proof. Statement 2

from Equation (9), $\forall x_i < \epsilon$, eliminate the numerator

$$m_{MultiMax} = 1 - (1 - e^{t_d(x_j-x_i)})/\Big(\sum_{x_l<b}^{L} e^{\sigma(x_l)-t_dx_i-(1-t_d)d}$$

$$+ \sum_{b\leq x_m \leq d}^{M} e^{x_m-t_dx_i-(1-t_d)d} + \sum_{x_n>d}^{N} e^{t_d(x_n-x_i)}\Big)$$

since $x_j - x_i < 1$ and $t_d < 1$, we have $e^{t_d(x_j-x_i)} > e^{x_j-x_i}$, also substitute $t_dx_i + (1-t_d)d$ with t_x, from Lemma A.1

$$> \frac{1 - e^{x_j-x_i}}{\sum_{x_l<b}^{L} e^{t_bx_l+(1-t_b)b-x_i} + \sum_{b\leq x_m\leq d}^{M} e^{x_m-x_i} + \sum_{x_n>d}^{N} e^{t_d(x_n-x_i)}}$$

$$\Rightarrow \mathcal{M}_{MultiMax}(\boldsymbol{x}) > \mathcal{M}_{MultiMax-l}(\boldsymbol{x}) \qquad \square$$

C. More visualizations

C.1. Single layer attention scores

As mentioned in Section 5.2, single layer attention scores are not informative for human beings, due to the complex interaction of information in deep transformer models.

C.2. Cumulative distribution of attention scores

We could calculate the cumulative distribution for each layer, i.e., the portion of attention scores smaller than a threshold as the thresholds increases. The result is shown in Figure 8. It can be seen that for most of the layers, MultiMax results in a sparser attention distribution, i.e., a large portion of attention scores are closer to zero comparing to SoftMax attention. Notably, the first two layers' attention distributions have a smaller degree of sparsity comparing to SoftMax. This shows that a smoother distribution is desired in these

two layers, as an optimized result of the training. This conforms to the observation in the previous studies that common low-level features in the shallow layers are shared across image patches (Schirrmeister et al., 2020). A sparse attention has a high risk of information lost.
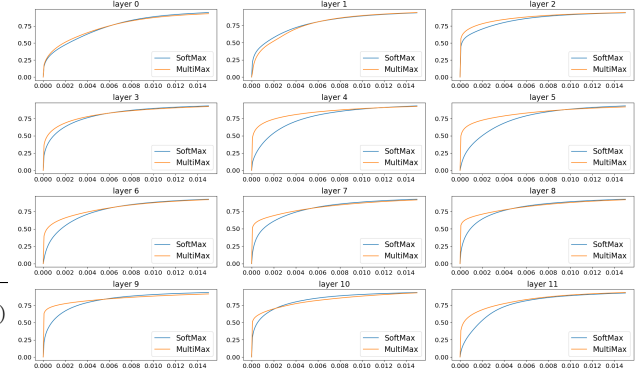

Figure 8. Cumulative distribution of the attention scores at each layer.

C.3. Connection between sparsification and over-smoothing

As shown by (Abnar & Zuidema, 2020), information originating from different input tokens gets increasingly mixed in deeper layers, and the information flow can be estimated by taking the attention weights out and multiplying them sequentially. Such a matrix multiplication makes the identity of each token fades exponentially, which relates to the over-smoothing problem in GCNs (Oono & Suzuki, 2019). Considering the information exchange across different attention heads, we take the the mean attention score over all heads out for multiplication, following the rollout technique (Abnar & Zuidema, 2020). In Figure 9, the discrepancy between the single layer and average accumulated SoftMax attention scores keeps increasing in the deeper layers. And

Layers	t_{b_1}	t_{d_1}	t_{b_2}	t_{d_2}	b_1	d_1	b_2	d_2
1	1.8347933	2.815388	0.9864913	0.68440557	1.185235	-1.208543	-2.1076407	1.9158255
2	1.9773115	1.9971638	0.985555	0.74650276	-0.8580209	0.02481092	-0.49835142	1.9772723
3	-1.1411996	1.4711196	1.9901285	0.8758977	0.18852632	2.8039892	2.9608543	1.0462786
4	0.6694808	1.206692	1.8682657	0.93786246	3.4023566	-1.5490056	2.500237	0.986331
5	0.8902384	1.5881691	1.8920481	0.72857785	2.5070796	-1.1942928	1.8854694	1.2248528
6	0.6015882	0.87738	2.818536	0.96271396	2.6490533	0.8454426	1.6205754	0.89434063
7	0.8023207	1.2427123	3.040797	0.84531546	2.6984618	1.2127148	1.2652112	1.2134424
8	0.64486825	0.79173684	2.5263662	0.968745	3.0230901	0.62191963	1.6307493	1.6259384
9	0.5796288	0.6852025	3.500835	0.99119073	2.675157	0.68776745	1.3239485	1.5808712
10	0.54873073	0.8240905	3.5563424	0.9692498	2.176066	0.39797062	0.9276044	1.5223614
11	0.38645744	0.6951747	4.0935583	0.9958999	1.6583583	0.29572898	0.77263904	2.9975116
12	0.16383016	0.25565386	3.2074118	0.99102634	1.6852132	-0.04795134	0.9796309	2.1836245

Table 7. MultiMax parameters of Deit-small trained on ImageNet.

Layers	t_{b_1}	t_{d_1}	t_{b_2}	t_{d_2}	b_1	d_1	b_2	d_2
1	0.6467285	0.7980957	0.98324585	0.9649048	0.7475586	-0.87939453	0.3395996	-0.14501953
2	0.69018555	0.8063965	0.98350525	0.9720764	0.25073242	0.15991211	0.2956543	-0.17687988
3	0.8557129	0.79797363	0.98939514	0.9855194	-0.12609863	0.06817627	0.14794922	-0.14428711
4	0.9662781	0.83569336	1.0231781	1.0240021	-0.07574463	0.8510742	-0.13220215	0.27368164
5	0.9260864	0.9187622	0.98670197	1.039093	-0.5239258	0.51416016	0.23999023	0.09521484
6	1.1514893	1.152832	0.98441315	1.0156403	0.1751709	0.05374146	-0.13269043	-0.08825684

Table 8. MultiMax parameters of the 6-layer Language Transformer trained on WikiText-103.
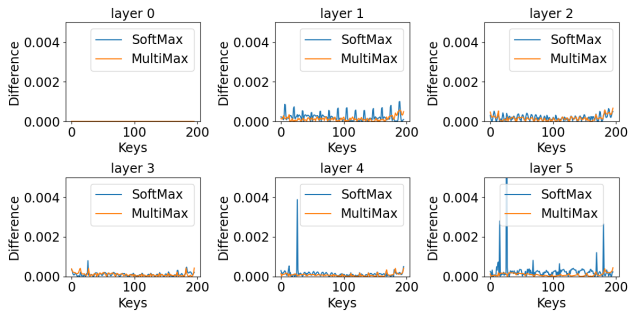

Figure 9. Comparing the discrepancy between rollout attention score and single layer attention score for SoftMax and MultiMax.

the comparison shows a much less accumulated error for our MultiMax attention.

D. The learned parameters of MultiMax

In this section, we provide the learned parameters of MultiMax for reference. There are differences and similarities between the learned modulation functions of vision and language transformers, which could be observed after plotting the curves as shown in Figure 10.:

- Similarly, the need for sparsity increases as the layer goes deeper, but much less sparsity are needed in general for the language transformer compring to vision transformer, according to the learned parameters.

- As opposed to vision transformer, stronger multi-modality is needed at shallower layers of the language transformer.
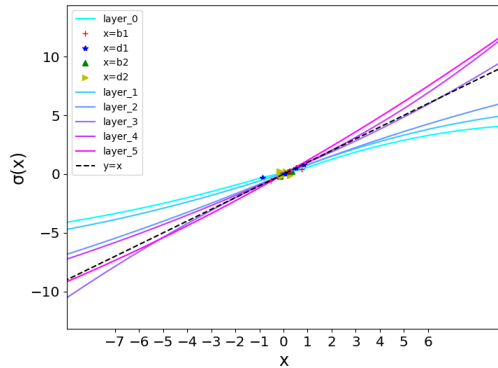

Figure 10. The learned modulator functions σ (Equation (6)) at each layer of the 6-layer language transformer trained on WikiText-103, comparing to identity mapping of the SoftMax input x (dashed black line).