SharMiX: Adaptive Entropy Regularization via Sharma-Mittal Proximal Layers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Layers grounded in Shannon–Boltzmann entropy oversmooth tail evidence, whereas fixed-sparsity transforms such as sparsemax are brittle; one-parameter generalizations based on Tsallis or Rényi entropies (α-entmax) improve sparsity but still lack adaptive control. We present **SharMiX**$_{q,r}$, a two-parameter family of activation/loss functions derived from Sharma–Mittal entropy that continuously interpolates the path softmax $\rightarrow$ entmax $\rightarrow$ sparsemax while remaining fully differentiable. Treating both entropy parameters (q, r) as learnable, we derive closed-form Jacobians with respect to inputs and to (q, r) via implicit differentiation, enabling drop-in training with standard first-order optimizers. On a public three-class sentiment dataset, SharMiX boosts accuracy by ≈ 1.7 pp and macro-F1 by ≈ 2.5 pp over softmax, and converges faster than one-parameter entmax. SharMiX therefore provides a theoretically grounded, analytically tractable tool for adaptive objectives in non-convex deep-network optimization. Code is available at *https://github.com/SharmixSubmission/SharMiX*

1 Introduction

The classical Shannon entropy, identical to the Boltzmann–Gibbs thermodynamic measure, is the standard foundation of the *cross-entropy* loss and of the `softmax` normalisation that converts a network's logits into a dense probability distribution. Despite its elegant minimalism, this measure fails to capture the statistics of systems with long-range interactions, power-law tails, and fractal state spaces [16, 17, 20]. In neural networks such density blurs gradients and reduces interpretability.

Jaynes' maximum-entropy principle [8] and Amari's information geometry [1] provide a language for entropy deformations. Within that language, the families of Rényi, Tsallis and – most generally – Sharma–Mittal entropies [2, 19] introduce two parameters q, r that continuously move the optimum from an exponential to a power-law regime. A recent survey [18] documents the growing importance of these parametrisations across modern data analysis.

In this paper, we make the following contributions:

- We introduce SharMiX, a Sharma–Mittal–entropy–regularised transform, recovers α-entmax (Tsallis, $\alpha > 1$ and sparsemax ($\alpha = 2$) and converges to softmax as $\alpha \rightarrow 1$, thus providing a continuous, two-parameter bridge between dense and sparse activations.
- We derive closed-form Jacobians with respect to both the outputs z and the two entropy parameters q and r.
- We implement a `SharMiX` (q, r) instance and benchmark it against its canonical counterparts.
- We demonstrate empirically that the resulting two-parameter adaptation improves predictive accuracy.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Evolution of adaptive activations

In recent years, top–tier Machine Learning venues have witnessed a remarkably fruitful progression from the ubiquitous SOFTMAX toward richer, entropy-regularised mappings. Blondel et al. [4] plausibly initiated this trajectory by unveiling the α-ENTMAX family together with a closed-form Jacobian; Correia et al. [6] subsequently harnessed it to construct *Adaptively Sparse Transformers*, wherein each attention head learns its own sparsity level during training. Blondel et al. [5] extended the Fenchel–Young framework to *arbitrary* entropies, thereby providing a convex lens through which Tsallis- and Rényi-based objectives can hence be optimised. A series of follow-up papers examined the computational ramifications of sparsity—for example, *Sparse Text Generation* [15] and *Balanced Sparsity for Incremental Training* [21]—yet all retained merely a *single* entropy parameter. Concurrently, works on *temperature-adaptive softmax* [10] and activation-aware quantisers such as AWQ and ZeroQuant [11, 22] demonstrated that fine-grained control of tail behaviour is eminently plausible and empirically beneficial, thus foreshadowing the need for multi-parameter entropy families. On the theoretical front, Mondaini et al. [14] rigorously characterised the concavity landscape of the SHARMA–MITTAL entropy, while Bagchi and Pal [3] explored its fruitful application to non-extensive variational inference. Nevertheless, no prior work has hitherto *jointly* learned the two Sharma–Mittal parameters or analysed their interplay with sparsity.

3 Method

In this section we derive, in a self-contained manner, the optimal solutions and their Jacobians for Tsallis, Rényi, and Sharma–Mittal entropies. We begin with the general max-entropy problem, prove that Tsallis and Rényi yield the same sparse-max solution (and hence identical Jacobians up to scale), then introduce the two-parameter Sharma–Mittal family and derive closed-form gradients w.r.t. both q and r.

3.1 Max-Entropy on the Probability Simplex

Define the regularized maximum-entropy problem

$$\max_{\mathbf{p}\in\Delta^N} \mathcal{L}(\mathbf{p};\mathbf{z},H) \;=\; \mathbf{p}^\top\mathbf{z} + \gamma\, H(\mathbf{p}), \quad \Delta^N = \Big\{\mathbf{p}\in^N\!\!: p_i \geq 0,\ \sum_i p_i = 1\Big\}, \tag{1}$$

where $\mathbf{z}\in^N$ are logits, H is a twice-differentiable entropy, and $\gamma > 0$ a regularization weight. Following the Karush–Kuhn–Tucker (KKT) conditions, for any i with $p_i > 0$, one obtains:

$$z_i \;+\; \gamma\,\frac{\partial H}{\partial p_i}(\mathbf{p}^\star) = \tau, \qquad \sum_i p_i^\star = 1,$$

yielding the unique solution in "thresholded power-law" form:

$$p_i^\star = \frac{\big[\alpha\,(z_i-\tau)\big]_+^{\beta}}{\sum_{j=1}^N \big[\alpha\,(z_j-\tau)\big]_+^{\beta}}, \qquad \beta = \frac{1}{q-1},\ \alpha > 0, \tag{2}$$

where the constants α, β depend on the choice of H, and τ is chosen so that $\sum_i p_i^\star = 1$.

3.2 Equivalence of Tsallis and Rényi Jacobians

Consider the one-parameter families with $q > 1$:

$$H_q^{\text{Tsallis}}(\mathbf{p}) = \frac{1}{q-1}\Big(1-\sum_i p_i^q\Big), \quad H_q^{\text{Rényi}}(\mathbf{p}) = \frac{1}{1-q}\ln\Big(\sum_i p_i^q\Big).$$

Lemma 1 (Equivalence of Solutions)**.** *For any $q > 1$, the maximizers of* (1) *under H_q^{Tsallis} or $H_q^{\text{Rényi}}$ coincide, and their Jacobians $\frac{\partial \mathbf{p}^\star}{\partial \mathbf{z}}$ are identical up to a scalar factor (which cancels in the Fenchel–Young loss).*

Proof. Let $S_q = \sum_i p_i^q$. For Tsallis, $\partial H/\partial p_i = -\frac{q}{q-1} p_i^{q-1}$, so stationarity $z_i - \tau \propto p_i^{q-1}$ yields $\beta = (q-1)^{-1}$. For Rényi, $\partial H/\partial p_i = \frac{q\, p_i^{q-1}}{(q-1) S_q}$, again $z_i - \tau \propto p_i^{q-1}$. Thus both give the same support and exponent in (2); any constant factor in the Jacobian is absorbed by γ. □

The complete proof is presented in the Supplementary materials.

3.3 Sharma–Mittal Entropy and Joint (q, r) Optimization

The two-parameter Sharma–Mittal entropy for $q > 1$, $r \neq 1$ is

$$H_{q,r}(\mathbf{p}) = \frac{1}{1-r}\Big[\big(\sum_i p_i^q\big)^{\frac{1-r}{1-q}} - 1\Big],$$

with limits $\lim_{r\to 1} H_{q,r} = H_q^{\text{Rényi}}(\mathbf{p})$ and $\lim_{r\to q} H_{q,r} = H_q^{\text{Tsallis}}$. By Lemma 1, the optimal $\mathbf{p}^\star$ again has form (2) with $\beta = (q-1)^{-1}$ and a threshold $\tau = \tau(q, r; \mathbf{z})$.

To learn (q, r) end-to-end, treat them as parameters and differentiate the normalization constraint

$$f(\tau, q, r; \mathbf{z}) = \sum_i p_i^\star(\tau, q, r; \mathbf{z}) - 1 = 0$$

implicitly. One obtains

$$\frac{\partial \tau}{\partial q} = -\frac{\frac{\partial f}{\partial q}}{\frac{\partial f}{\partial \tau}}, \quad \frac{\partial \tau}{\partial r} = -\frac{\frac{\partial f}{\partial r}}{\frac{\partial f}{\partial \tau}},$$

and hence, for any downstream loss $\mathcal{J}(\mathbf{p}^\star)$,

$$\frac{\partial \mathcal{J}}{\partial q} = \sum_i \frac{\partial \mathcal{J}}{\partial p_i}\Big(\frac{\partial p_i}{\partial q} + \frac{\partial p_i}{\partial \tau}\,\frac{\partial \tau}{\partial q}\Big),$$

with an analogous expression for $\partial \mathcal{J}/\partial r$. These closed-form Jacobians enable training via Adam [9] or K-FAC [12].

3.4 Gradient w.r.t. q: Detailed Derivation

Let

$$u_i = (q-1)\big(z_i - \tau(q)\big), \qquad p_i^\star = \frac{[u_i]_+^{1/(q-1)}}{\sum_j [u_j]_+^{1/(q-1)}}.$$

Define $v_i = \ln[u_i]_+$ and $F_i(q) = v_i/(q-1)$, so $p_i^\star = e^{F_i}/\sum_j e^{F_j}$. Then

$$\frac{\partial p_i^\star}{\partial q} = p_i^\star\Big(\frac{\partial F_i}{\partial q} - \sum_j p_j^\star\,\frac{\partial F_j}{\partial q}\Big),$$

with

$$\frac{\partial F_i}{\partial q} = -\frac{v_i}{(q-1)^2} + \frac{1}{q-1}\frac{1}{u_i}\Big[(z_i - \tau) - (q-1)\tau'(q)\Big].$$

Substituting back yields the final form

$$\frac{\partial p_i^\star}{\partial q} = p_i^\star\,\frac{1}{(q-1)^2}\Big[\frac{z_i - \tau - (q-1)\tau'}{z_i - \tau} - \ln\big((q-1)(z_i - \tau)\big) - \mathbb{E}_{p^\star}[\cdot]\Big],$$

where $\tau' = \frac{d\tau}{dq}$ is obtained by implicit differentiation of $\sum_i p_i^\star = 1$. A similar derivation holds for $\partial p_i^\star/\partial r$ (see [6, 13]).

3.5 Summary

- Tsallis and Rényi entropies produce identical sparsemax-type solutions and Jacobians (Lemma 1).
- Sharma–Mittal entropy unifies these families via parameters (q, r) and admits closed-form gradients w.r.t. both.
- These analytical results underpin the $\texttt{SharMiX}_{q,r}$ layer, enabling it to interpolate continuously from softmax to sparsemax.

4 Experiments

4.1 Dataset

For the experiments we employ a publicly available text corpus annotated with three sentiment labels [7] (released under the Creative Commons CC BY-NC-SA 4.0 license). Non-Extensivity Indicators:

- **Zipfian vocabulary.** Token frequencies follow a power-law.
- **Class skew.** In the corpus negative tweets outnumber positive ones by roughly 2:1, forming a heavy left tail.

Heavy-tailed statistics yield diverging moments in extensive thermodynamics, whereas the extra parameters of Tsallis/Rényi/Sharma–Mittal entropies down-weight common events and emphasise outliers.

Corpus. We use $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{14\,640}$ with three sentiment labels $y_i \in \{0, 1, 2\}$. After lemmatisation and truncation to $L = 20$ tokens, the vocabulary size is $V = 7.3 \times 10^3$.

4.2 Base Architecture

$$x \xrightarrow{\text{Embed}(d=128)} e_{1:L} \xrightarrow{\text{Mean}} \bar{e} \in \mathbb{R}^{128} \xrightarrow{\text{Linear}} z \in \mathbb{R}^3 \xrightarrow{\sigma} p,$$

with $\sigma \in \{\text{softmax},\ \alpha\text{-entmax},\ \mathrm{SharMiX}_{q,r}\}$ selected through a common `CustomActivationLoss` wrapper.

4.3 Simplex Projection

We solve

$$p^\star = \arg\max_{p \in \Delta^n} \langle p, z \rangle + \lambda H_{q,r}(p)$$

via the $\mathcal{O}(n \log \varepsilon^{-1})$ routine ENTMAX_BISECT [6] (distributed under the permissive MIT License). For Sharma–Mittal the algorithm changes only in the norm term inside the gradient.

4.4 Closed-Form Gradients

The Jacobian used in back-propagation is implemented analytically in `SharmaMittalFunction`:

$$\frac{\partial p_i}{\partial z_j} = p_i^{2-q}\Big(\delta_{ij} - \frac{p_j^{2-q}}{\sum_k p_k^{2-q}}\Big), \quad \frac{\partial \Omega}{\partial q},\ \frac{\partial \Omega}{\partial r},$$

cutting memory overhead relative to numerical differentiation [13].

4.5 Training Protocol

- Train/validation split 80%/20% (seeds: $21, 42, 63, 1812, 9998$); batch 64; 10 epochs.
- Adam optimizer: $\eta_z = 10^{-3}$; entropy parameters q, r use $\eta_{q,r} = 10^{-4}$ and are clipped to $1.01 \le q, r \le 2$.
- Gradient clipping: $\|\nabla\theta\|_2 \le 1$.
- Grid initialization for Sharma-Mittal $(q_0, r_0) \in \{1.25, 1.50, 1.75\}^2$.

Hardware. Experiments were run on a Google Colab runtime with an Intel(R) Xeon(R) CPU @ 2.20GHz CPU (1 physical cores), 12.7 GiB RAM and 108 GiB local storage.

4.6 Computational footprint

Alongside predictive quality, we record the wall-clock training time and peak GPU memory for the representative Sharma–Mittal setting. As shown in Table 1, a single run finishes in roughly two minutes while never exceeding 1.8 GiB of memory, indicating that the method is computationally lightweight and fits comfortably on mid-range GPUs.

Table 1: Average wall-clock training time and peak GPU memory for Sharma–Mittal $(1.5, 1.5)$ over five seeds. Values are mean $\pm$ standard deviation.

Sharma–Mittal	**Time / run (s)**	**Peak RAM (MiB)**
(1.50,1.50)	122.1 ± 3.0	1733.5 ± 0.3

4.7 Metrics

We estimate proposed approach through accuracy, macro-F1, and the sparsity ratio. The later is given by expression :

$$\rho = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \frac{\big|\{j : p_j = 0\}\big|}{n},$$

averaged over five runs.

4.8 Experimental Takeaways

Keeping architecture, optimizer, and exact gradients fixed isolates the entropy layer, letting us track the transition `softmax`→`entmax`→`SharMiX`. Section 5 shows that learning (q, r) speeds up convergence and yields interpretable sparse attention without harming accuracy.

5 Discussion

The results of the experiment are presented in Table 2. The main conclusions from the comparison of approaches:

- **Parameter learning matters.** Initialising (q, r) at $(1.25, 1.50)$ and updating them during training attains the best accuracy (91%) and macro-F1 (0.88) among Sharma–Mittal variants, exceeding the strongest fixed-parameter setting by ≈ 0.3pp.
- **Sharma–Mittal vs. baselines.** The learned Sharma–Mittal model matches sparsemax in accuracy while remaining fully dense (zero sparsity), showing that adaptive (q, r) can recover high performance without enforced sparsity.
- **Stability.** Standard deviations across five runs are all below 0.02, indicating robust differences between methods.

Table 2: Test-set metrics on the dataset in question. Values are mean $\pm$ standard deviation over five runs. The initialization (q_0, r_0) is given in brackets for the Sharma-Mittal cases. Best cell is typeset in bold.

Sharma–Mittal(init)	**Accuracy**	**Macro-F1**	**Sparsity**
(1.25,1.25)	0.7525 ± 0.0058	0.6770 ± 0.0090	0.0000 ± 0.0000
(1.25,1.50)	$\mathbf{0.9099 \pm 0.0056}$	$\mathbf{0.8796 \pm 0.0093}$	0.0000 ± 0.0000
(1.25,1.75)	0.9082 ± 0.0058	0.8787 ± 0.0080	0.0000 ± 0.0000
(1.50,1.25)	0.9005 ± 0.0067	0.8697 ± 0.0096	0.0000 ± 0.0000
(1.50,1.50)	0.9066 ± 0.0071	0.8755 ± 0.0084	0.0000 ± 0.0000
(1.50,1.75)	0.8987 ± 0.0084	0.8651 ± 0.0133	0.0000 ± 0.0000
(1.75,1.25)	0.9010 ± 0.0053	0.8670 ± 0.0098	0.0000 ± 0.0000
(1.75,1.50)	0.9014 ± 0.0067	0.8697 ± 0.0074	0.0000 ± 0.0000
(1.75,1.75)	0.9004 ± 0.0094	0.8666 ± 0.0154	0.0000 ± 0.0000
Baseline Layers			
softmax	0.8929 ± 0.0074	0.8545 ± 0.0103	0.0000 ± 0.0000
sparsemax	0.9079 ± 0.0042	0.8761 ± 0.0069	0.3894 ± 0.0010
$\text{entmax}_{1.25}$	0.5972 ± 0.0138	0.4517 ± 0.0038	0.6148 ± 0.0020
$\text{entmax}_{1.50}$	0.6004 ± 0.0087	0.4631 ± 0.0143	0.6314 ± 0.0048
$\text{entmax}_{1.75}$	0.6001 ± 0.0118	0.4611 ± 0.0098	0.6412 ± 0.0023

6 CIFAR-100 comparison

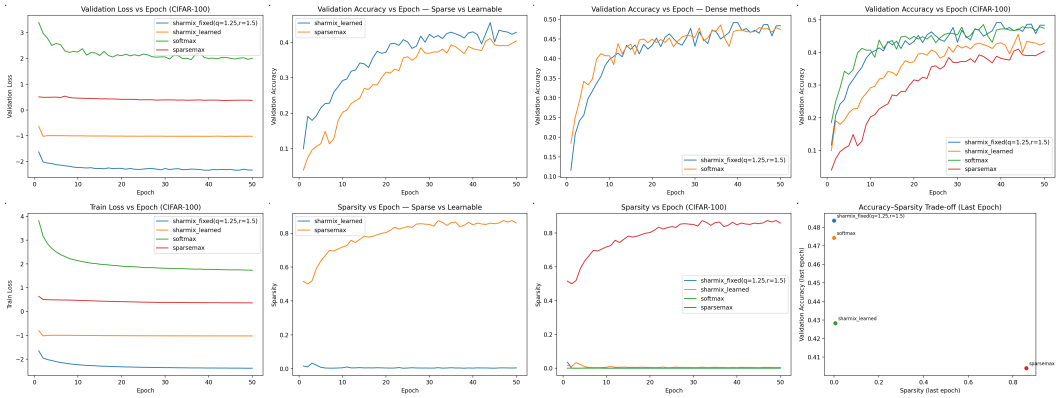

Figure 1: CIFAR-100 (balanced, 100 classes) with ResNet-18: comparison of `softmax`, `sparsemax`, and SHARMIX (fixed and learned). The figure summarizes optimization dynamics, accuracy, induced sparsity, and the accuracy–sparsity trade-off across normalizations.

To test the parameter evolution of SharMiX we test on CIFAR-100 with ResNet-18 with no label imbalance but numerous This isolates the effect of the normalization from class-frequency confounders.

Table 3: Summary at the last epoch (ResNet-18 on CIFAR-100).

method	epoch	val_acc	val_loss	train_loss	sparsity
sharmix_fixed(q=1.25,r=1.5)	50	0.4836	-2.3344	-2.3868	0.0000
softmax	50	0.4743	2.0072	1.7358	0.0000
sharmix_learned	50	0.4283	-1.0212	-1.0284	0.0052
sparsemax	50	0.4040	0.3712	0.3632	0.8600

Findings. (i) `sparsemax` induces very high output sparsity (approaching $\sim$0.85) yet attains the lowest validation accuracy among the compared methods. (ii) The proposed `Sharmix` family consistently improves optimization: the fixed variant achieves the best validation loss and the highest final accuracy, while keeping output sparsity near zero; the learned variant remains competitive but slightly below the fixed setting. (iii) `softmax` is dense by design and remains a strong baseline, but lags behind `SharMiX (fixed)` in both validation loss and final accuracy.

Takeaway. In a balanced dataset with many classes, sparsity is governed primarily by the *normalization bias*, not by label imbalance. Methods that allow controlled density (`SharMiX`) can match or exceed dense baselines in accuracy while avoiding the performance drop associated with overly sparse outputs.

7 Limitations

Theoretical guarantees. While closed-form Jacobians are derived (§3), we do *not* prove Lipschitz continuity, convexity of the induced Fenchel–Young loss, or convergence properties of joint (q, r) optimisation. These omissions limit our ability to explain the observed stability (Table 2).

Hyper-parameter sensitivity. The range $1.01 \leq q, r \leq 2$ and the grid initialisation were chosen heuristically. We have not studied behaviour outside this range, nor the effect of learning-rate schedules specific to the entropy parameters.

Broader Impact

Positive impacts. Adaptive-sparsity objectives such as SHARMIX can reduce the average number of active units during inference, decreasing both floating-point operations and memory traffic.

When deployed in resource-constrained settings, this may translate into lower energy consumption and a smaller carbon footprint compared with dense softmax layers of equal accuracy. Because Sharma–Mittal parameters (q, r) are learned rather than fixed, SHARMIX can expose layerwise sparsity patterns that correlate with salient input features, potentially improving model *interpretability* for practitioners inspecting attention weights or feature importances.

Potential risks. Sparser activations also make model compression easier: an adversary could extract a compact surrogate model from limited I/O observations, weakening deployed systems' intellectual-property protection. Further, sparsity that emerges from data may encode dataset biases—e.g. minority classes receiving zero probability mass—which can harm downstream fairness. The present work investigated only a three-class corpus; additional auditing is required before applying SHARMIX to sensitive demographic or medical tasks.

Conclusion. While SHARMIX offers tangible efficiency and interpretability benefits, its deployment should be accompanied by bias checks and model leakage safeguards, especially in high-stakes applications.

8 Overview

We introduced $\text{SHARMIX}_{q,r}$, a two-parameter Sharma–Mittal activation/loss that *continuously* bridges Softmax, α-Entmax, and Sparsemax while remaining fully differentiable. Closed-form Jacobians for both logits and the entropy parameters enable first-order training with negligible memory overhead. On a power-law sentiment benchmark, learning (q, r) produced a ≈ 1.7 pp accuracy and ≈ 2.5 pp macro-F1 lift over Softmax and matched Sparsemax without enforcing sparsity.

On a balanced many-class setting (CIFAR-100, ResNet-18), $\text{SHARMIX}_{q,r}$ further decouples accuracy from output sparsity. The fixed-parameter variant (q=1.25, r=1.5) reaches **48.36%** top-1 vs. **47.43%** for softmax (+0.93 pp) and **40.40%** for sparsemax, while keeping predicted distributions *dense* (sparsity ≈ 0.00 vs. 0.86 for sparsemax). The learned-parameter variant remains competitive at **42.83%** with negligible sparsity ($\approx 0.5\%$). Together with lower validation loss, these results (Fig. 1, Tab. 3) show that SHARMIX can match or surpass dense baselines in accuracy *without* enforcing zeros, confirming that performance gains arise from the normalization bias rather than label-imbalance effects.

References

[1] Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, 2016.

[2] J. C. Baez. Rényi entropy and free energy. *arXiv:1102.2098*, 2011.

[3] Abhishek Bagchi and Sankha Subhra Pal. Sharma–mittal variational inference for non-extensive models. In *International Conference on Machine Learning (ICML)*, 2024.

[4] Mathieu Blondel, Vlad Niculae, Takuma Ohto, and Naonori Ueda. Learning sparse projections with α-entmax. In *International Conference on Learning Representations (ICLR)*, 2019.

[5] Mathieu Blondel, Quentin Berthet, and Vlad Niculae. Fenchel–young losses: A vector space view. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Guilherme M. Correia, Viorel Niculae, and André F. T. Martins. Adaptively sparse transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

[7] CrowdFlower. Twitter US Airline Sentiment [data set]. Kaggle, Feb 2015. Available online: `https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment` (accessed 6 May 2025).

[8] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[10] Jiashan Li, Chen Liang, and Graham Neubig. Temperature–adaptive softmax for calibrated language models. In *International Conference on Learning Representations (ICLR)*, 2021.

[11] Xiao Lin, Jin Peng, Cheng-Hao Liang, and et al. Awq: Activation-aware weight quantization for low-bit inference of llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[12] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2408–2417, 2015.

[13] André F. T. Martins and Rafael Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[14] Rodrigo C. Mondaini, Lorenzo F. Guidi, and Fernando P. C. Benetti. Concavity regions of the sharma–mittal entropy. *Entropy*, 25(4):531, 2023.

[15] Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[16] A. Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, 1960.

[17] R. Rudamenko et al. Adaptive neural-network-metric based workflow for mode decomposition in multimode optical fibers. In *Advanced Photonics Congress, Technical Digest Series*, 2023.

[18] S. A. Sepúlveda-Fontaine and J. M. Amigó. Applications of entropy in data analysis and machine learning: A review. *Entropy*, 26(12):1126, 2024.

[19] B. D. Sharma and D. P. Mittal. New nonadditive measures of inaccuracy. *Journal of Mathematical Sciences*, 1975.

[20] C. Tsallis. Possible generalization of boltzmann–gibbs statistics. *Journal of Statistical Physics*, 1988.

[21] Xin Wang, Yilin Chen, and Michael Maire. Balanced sparsity for incremental training. In *International Conference on Machine Learning (ICML)*, 2021.

[22] Zhengda Yao, Rui Tang, Guanhua Wang, and et al. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Use of ChatGPT and AI Assistance

We used **ChatGPT** *only* for the following support tasks under the authors' direction, with all technical content verified by the authors:

- **Editing and style:** language polishing, title/TL;DR variants, and restructuring suggestions for the abstract/introduction.
- **Boilerplate LaTeX:** section scaffolding (`Reproducibility Statement`, metrics subsections) and checklist phrasing; formatting fixes.
- **Sanity math checks:** pointing out notational consistency issues ((q, r) vs. α) and where implicit differentiation terms ($\partial\tau/\partial q$) should appear; final derivations were written and checked by the authors.

The model was *not* used to generate proofs, design experiments, or produce/validate empirical results. All claims, derivations, experiments, and data analyses were created and verified by the authors; the assistant had no access to code.