Input differentiation via negative computation

Linghao Kong* LINGHAO@MIT.EDU

Angelina Ning* ANGN_731@MIT.EDU

Nir N. Shavit SHANIR@MIT.EDU

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Understanding neuronal mechanisms in large language models remains challenging, particularly due to polysemanticity and superposition. In this work, we further investigate the previously identified "Wasserstein neurons," characterized by non-Gaussian pre-activation distributions. Our analysis reveals that these neurons are more prevalent and exhibit faster learning dynamics in larger models. Critically, we demonstrate for the first time the mechanistic significance of the negative activation space, showing that Wasserstein neurons leverage negative pre-activations for nuanced input differentiation, especially regarding syntactic and structural tokens. Ablation experiments confirm that constraining negative activations significantly degrades model performance, highlighting previously underappreciated computational roles. These findings offer new directions for interpretability research by emphasizing the importance of negative computation.

Keywords: Interpretability, Activation function, Input differentiation, Entanglement

1. Introduction

Interpretability in machine learning has been a longstanding goal, though it is significantly complicated by the presence of polysemantic neurons [5, 6, 9, 16, 20]. Polysemantic neurons respond to multiple, seemingly unrelated concepts, obscuring straightforward interpretations of neuron functionality. Such neurons have been observed in a wide range of models, notably in large language models (LLMs) [2, 7, 15, 16]. One driver of polysemanticity is the superposition phenomenon, which posits that neural networks are able to encode more features than they have neurons due to the sparse nature of input features [4], though recent work has shown evidence that polysemanticity can arise in settings without more features than neurons [1, 10].

Despite these interpretability challenges, prior works have successfully investigated the roles of specific neurons within LLMs [6, 7, 19]. In these studies, the responsiveness of polysemantic neurons is generally characterized in terms of binary pre-activation (positive or negative). More recently, a group of neurons with notably non-Gaussian output distributions—termed Wasserstein neurons due to the distinctive statistical properties of their pre-activations—was identified within the feedforward up-projection matrices of transformer blocks [17]. These neurons exhibit high sensitivity to weight sparsification and uniquely map similar input vectors to dissimilar scalar outputs.

In this work, we build upon such findings to further investigate the role of Wasserstein neurons in the Pythia suite of LLMs [3]. We observe that Wasserstein neurons are more prevalent in larger models and, intriguingly, begin learning and converge earlier compared to other neuron populations. We also find that Wasserstein neurons in early layers seem to attend to token pairs that

*Equal contribution.

those in other layers do as well. We empirically investigate the pairs of tokens that Wasserstein neurons map the furthest relative to their similarity, finding that those are grammatical and structural in nature. Finally, we discover that Wasserstein neurons in particular utilize the negative portion of the activation space to perform such differentiation, which we term *negative computation*. To our knowledge, this work is the first to show the functional and mechanistic significance of negative activation spaces, moving beyond previous examinations limited primarily to their training advantages [8, 11, 12, 14, 18]. By highlighting the nuanced role of negative activations, our findings encourage future interpretability frameworks to investigate beyond just the polarity of neuron activations.

2. Results

We investigate Wasserstein neurons in the up projection matrices of each feedforward block in the Pythia suite of LLMs, leveraging the availability of model checkpoints during training [3]. Here, we define a neuron as a single row vector within a linear layer's weight matrix. Each neuron computes a scalar output as the dot product of its weight vector with an incoming input vector. To analyze neuron behavior, we record input vectors and their corresponding scalar outputs for every neuron by running the models on the Wikitext-2 test dataset [13].

We adopt previously established metrics: the Wasserstein distance (WD) of a neuron's output distribution, and the mapping difficulty (MD) of a neuron [17]. Briefly, to compute WD, we normalize a neuron's output distribution to zero mean and unit variance, then calculate the 1-WD between this distribution and a standard normal distribution. For MD, we sample pairs of input vectors and compute the L_2 distances between them, as well as the L_2 distances between their respective scalar outputs. Input distances are normalized between zero and one, while output distances are normalized by their median value. MD is then calculated as the average ratio of normalized output distances to normalized input distances (OI ratio) across all sampled pairs. Given the strong correlation between WD and MD [17], we use these metrics interchangeably throughout our analysis.

2.1. Wasserstein neurons are more prevalent in and learn faster in larger models

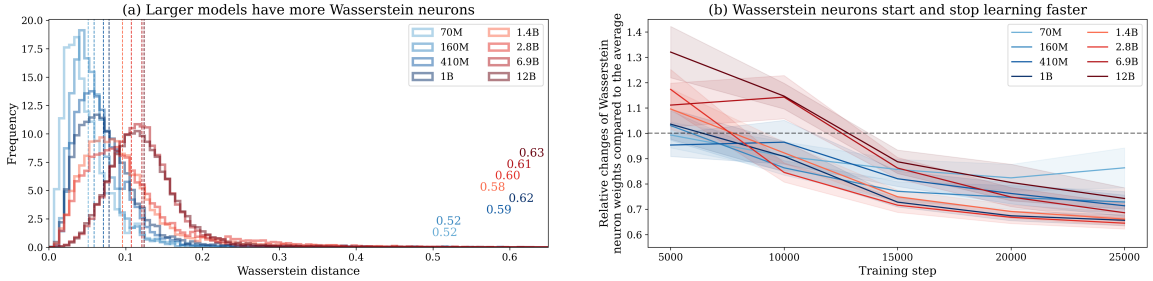

Figure 1: Wasserstein neurons across model size and training. (a) Larger models have more Wasserstein neurons, both on average, as indicated by the dotted vertical line, as well as by the maximum, in general. (b) Especially for larger models, Wasserstein neurons tend change their weights much more rapidly initially during training, but then change less than other neurons. The top 10 Wasserstein neurons were collected for each model. Error bars represent ±1 standard error of the mean.

We measured the WD for all neurons in the up projection matrix of the second transformer block across Pythia models ranging from 70 million to 12 billion parameters, evaluated at multiple

training checkpoints. As network size increases, both the average WD per neuron and the maximum WD observed across neurons tend to increase (Figure 1a). Notably, this increase in WD is broadly distributed across neurons rather than confined to a small subset, indicating that a larger fraction of neurons exhibit greater deviation from Gaussian-like activation distributions as model size scales up. These observations suggest a potential relationship between the prevalence of Wasserstein neurons and the overall expressive capability of larger neural networks.

We also observed interesting dynamics in how Wasserstein neurons learn compared to typical neurons. Early in training, Wasserstein neurons exhibit rapid changes in their weights, as measured by the L_2 norm difference computed at successive 5000-step intervals. However, after this initial rapid adjustment, their weights stabilize more quickly than those of other neurons, with their L_2 norm difference between subsequent intervals correspondingly smaller (Figure 1b). This pattern potentially indicates rapid convergence toward stable weight configurations, suggesting that Wasserstein neurons might initialize closer to effective solutions.

2.2. Early layer Wasserstein neurons attend to token pairs of other Wasserstein neurons

We now focus our analysis on the Pythia-1.4B model. For each feedforward block, we compute the mapping difficulty (MD) of every neuron in the up projection matrix. Following prior work, we refer to neurons with high MD as entangled neurons [17]. For each of the 10 most entangled neurons in a given layer, we identify the 10 input pairs with the highest OI ratios. This gives us a pool of 100 "high-effort" token pairs per layer. (See Section A for more implementation details.)

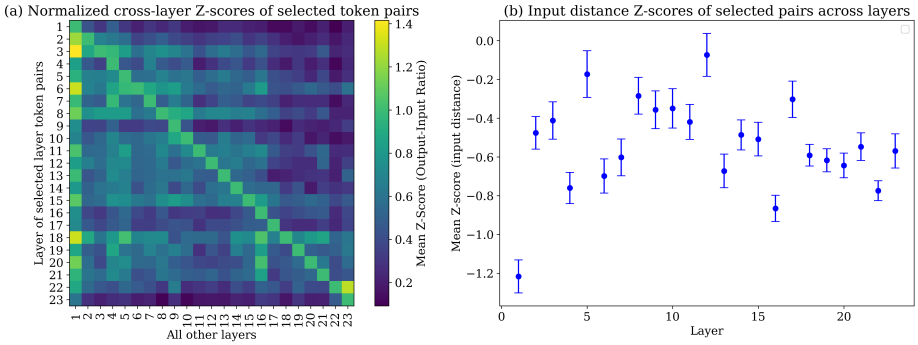

Figure 2: Specialized processing of high-OI token pairs across layers. (a) The normalized cross-layer z-scores of the top OI ratio token pairs for each layer. (b) The z-score of the input distances of the top OI ratio pairs for each layer. Layer 1 (index 0) is excluded due to instability in the OI ratio metrics, as many inputs remain indistinguishable at this early stage, yielding unreliable measurements. Layer indices are zero-indexed. Error bars represent ±1 standard error of the mean.

To investigate how these high-effort token pairs are handled across the network, we compute the z-score of the OI ratio for each selected pair across all layers, normalized to the layer from which the entangled neurons were selected. This measures how anomalously difficult a pair is to map within the distribution of pairs seen by entangled neurons in different layers. We find that these token pairs typically receive the highest z-scores in the same layer from which they were selected, implying that each layer's entangled neurons specialize to process certain types of inputs. Once a token pair is processed heavily in one layer, it tends to receive less computation in subsequent layers, suggesting a layer-wise division of labor in handling tokens that must be distinguished.

Interestingly, layer two exhibits a distinct pattern. Despite not being the source of the selected token pairs in other layers, it often has elevated z-scores to these pairs, sometimes even more than the source layer. Such a trend indicates that many difficult distinctions are already being emphasized in one of the very first feedforward blocks, suggesting this layer's foundational role in highlighting subtle input differences, possibly alleviating the computational difficulty of subsequent layers.

We also analyze the input distances of the selected top OI ratio token pairs. Compared to all input pairs, the 100 high-OI pairs consistently have smaller input distances across all layers. This trend implies that entangled neurons are especially focused on mapping nearby inputs to divergent outputs. Again, this effect is strongest in layer two, reinforcing its unique role in early representations. Taken together, these findings suggest that entangled neurons are not uniformly distributed in their function, but instead contribute to a hierarchical and specialized processing pipeline. Early layers, particularly layer two, appear to play an outsized role in transforming subtle differences into separable representations, which are then further refined or preserved in later stages.

2.3. Empirical investigation into specific Wasserstein neurons

To better understand the nature of semantic processing in layer two, we analyze its 100 most entangled token pairs. Consistent with earlier observations that layer two plays a foundational role in subsequent computation (Section 2.2), we interestingly find that it is primarily involved in syntactic construction, establishing grammatical relationships and sentence structure. A substantial fraction of these pairs involve functional and transitional tokens such as "the," "and," "to," "of," and "in," highlighting the model's need to differentiate between high-frequency tokens that carry limited semantic content individually but are critical for structuring sentences. These findings suggest that Wasserstein neurons in layer two serve as a *syntactic scaffold* that future layers may build upon.

To probe deeper into contextual processing, we also examined the tokens that follow each entangled pair. No strong patterns emerged, suggesting that the key computational burden at this stage lies in disambiguating contextually similar tokens that nonetheless lead to divergent subsequent ones, a transformation likely handled in subsequent layers (see Section B for further analysis).

(a) Neuron 5176 Token Associations

T1	T2	NT1	NT2
Ġfor	Ġthe	Ġa	Ġunrest
ĠThe	Ġfor	Ġgrowing	Ġhis
Ġto	Ġof	Ġthe	Ġthe
Ġthe	Ġin	Ġearly	ĠQ
Ġof	Ġto	Ġnational	Ġthat

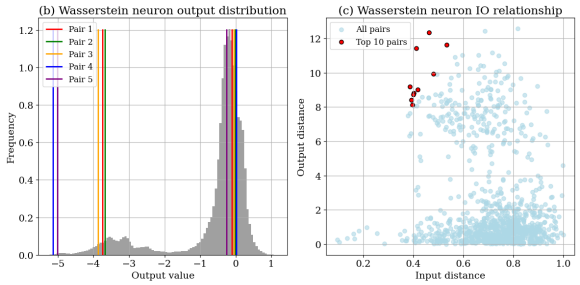

Figure 3: Semantic processing of entangled neurons. (a) Pairs of tokens with the greatest OI ratio (T1 and T2), as well as their next tokens (NT1 and NT2) for neuron 5176. Ġ indicates the start of a word. (b) Location of each pair of tokens in the output space. All but one token is mapped to a negative pre-activation. (c) The input-output (IO) relationship of this neuron.

At the individual neuron level, we find that highly entangled neurons appear to specialize in narrow syntactic roles. Neuron 5176 exhibits a high OI ratio for conjunctional and prepositional transitions such as "for," "the," "of," and "in" (Figure 3a), which are essential for maintaining coherence.

Notably, these distinctions manifest primarily in the negative portion of its pre-activation output space (Figure 3b), suggesting a richer use of the activation space than is often acknowledged. Because prior work frequently overlooks the functional significance of negative pre-activations, even those in non-ReLU models, we further investigate this phenomenon in Section 2.4.

2.4. Wasserstein neurons utilize negative computation to differentiate inputs

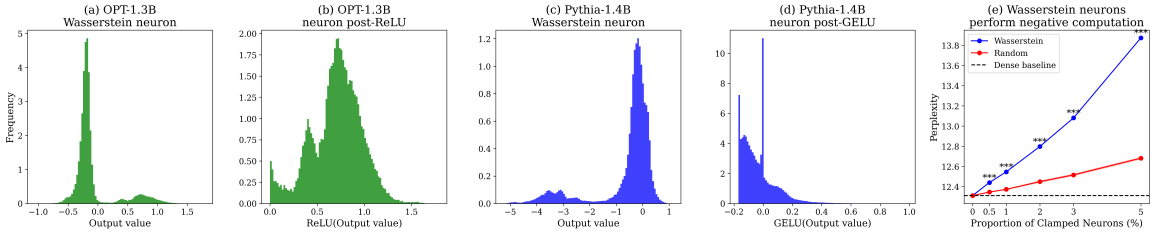

Figure 4: Wasserstein neurons use the negative activation space to perform meaningful computation. (a) A representative Wasserstein neuron in OPT-1.3B. (b) Post-ReLU activations of the same neuron in OPT-1.3B, with the zero values removed for visualization. (c) A representative Wasserstein neuron in Pythia-1.4B. (d) Post-GELU activations of the same neuron in Pythia-1.4B. (e) Removing the ability of Wasserstein neurons to conduct negative computation significantly hinders the model's performance. *** indicates $p \leq 10^{-4}$. Error bars represent ±1 standard error of the mean.

Finally, we investigate the functional importance of the negative pre-activation space in Wasserstein neurons. Compared to those in a ReLU-based model such as OPT-1.3B, which exhibit predominantly smooth, exponential-like negative pre-activation distributions, Wasserstein neurons in Pythia-1.4B (which uses GELU) frequently display rich, multimodal structure in their negative pre-activations (see Section C). These patterns persist even after activation (Figure 4a–d), indicating that meaningful computation is occurring in the negative activation space for Wasserstein neurons.

To probe this quantitatively, we conduct a targeted ablation experiment. In each up projection matrix in Pythia-1.4B, we identify the neurons with the highest Wasserstein distance. We then clamp the post-activation outputs of these neurons to be non-negative, effectively applying a ReLU after the GELU. We measure the resulting degradation in model performance using perplexity on the WikiText-2 validation set. The results are striking: ablating only the negative activations—a seemingly weak change—of the top 5% of Wasserstein neurons increases perplexity from 12.3 to nearly 14. In contrast, clamping an equal number of randomly selected neurons has a much smaller effect. This demonstrates that Wasserstein neurons rely heavily on their ability to compute in the negative activation space to perform their nuanced input differentiation.

To our knowledge, this is the first study to directly demonstrate the mechanistic importance of the negative activation space in large language models. While previous work has highlighted the benefits of smooth activation functions like GELU in terms of gradient flow and convergence [8, 12, 18], our findings show that the negative region itself enables qualitatively different forms of computation. In particular, Wasserstein neurons appear to specialize in exploiting this region, offering new insights into how non-ReLU activations support rich, multimodal representations. These results suggest that the negative activation space is not merely a byproduct of smooth activations but a functional substrate for distinct computational roles. Future work may further uncover how models allocate representational capacity across the activation space, and how such allocation arises.

References

[1] Micah Adler, Dan Alistarh, and Nir Shavit. Towards combinatorial interpretability of neural computation. *arXiv preprint arXiv:2504.08842*, 2025.

[2] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

[3] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[4] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[5] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3): e30, 2021.

[6] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

[7] Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

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

[9] Adam S Jermyn, Nicholas Schiefer, and Evan Hubinger. Engineering monosemanticity in toy models. *arXiv preprint arXiv:2211.09169*, 2022.

[10] Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes. *arXiv preprint arXiv:2312.03096*, 2023.

[11] Donghyun Lee, Je-Yong Lee, Genghan Zhang, Mo Tiwari, and Azalia Mirhoseini. Cats: Contextually-aware thresholding for sparsity in large language models. *arXiv preprint arXiv:2404.08763*, 2024.

[12] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

[13] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[14] Iman Mirzadeh, Keivan Alizadeh, Sachin Mehta, Carlo C Del Mundo, Oncel Tuzel, Golnoosh Samei, Mohammad Rastegari, and Mehrdad Farajtabar. Relu strikes back: Exploiting activation sparsity in large language models. *arXiv preprint arXiv:2310.04564*, 2023.

[15] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

[16] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[17] Shashata Sawmya, Linghao Kong, Ilia Markov, Dan Alistarh, and Nir N Shavit. Wasserstein distances, neuronal entanglement, and sparsity. In *The Thirteenth International Conference on Learning Representations*, 2025.

[18] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[19] Alessandro Stolfo, Ben Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. Confidence regulation neurons in language models. *Advances in Neural Information Processing Systems*, 37:125019–125049, 2024.

[20] Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic, 2024.

Appendix A. Details on layer-wise input mapping

Because neurons with high WD are those that disproportionately amplify the distance between similar input pairs, we refer to them as entangled neurons, following the convention set by previous work [17].

Our exact setup is as follows: the model processes an input of 140 token sequences, created from the Wikitext-2 test dataset, each consisting of 2048 tokens. We aim to identify the most entangled neurons in each layer and analyze the token pairs that these neurons are most sensitive to, across different layers.

We begin by generating 1000 random pairs of tokens from the flattened array of input IDs. Token pairs were selected without replacement, ensuring that all 2000 token IDs were unique. Using these token pairs, we calculated the mapping difficulty (MD) of all 8192 neurons in each layer.

For each of the 24 layers, we identified the top 10 entangled neurons based on their MD values. We considered these neurons to do the most "work" in processing the relationship between tokens in a pair—i.e., those whose outputs are most separated despite relatively small input differences. For each of the selected neurons, we identified the 10 token pairs with the highest output difference to input difference ratio (referred to as "output-input (OI) ratios" in other sections), resulting in a total of 100 token pairs per layer. While some pairs could be repeated across neurons, these repetitions help highlight the token pairs that are most "effortful" to handle in a given layer.

For each of the 100 identified token pairs in a given layer, we computed the average OI ratio across the top 10 most entangled neurons, considering not only the current layer but also previous and successive layers. This resulted in a (23, 100) array of average output-input ratios, where 23 represents the layers being considered, excluding the first layer due to its abnormal behavior. Next, we normalized these values by calculating the z-score for each token pair relative to the distribution of output-input ratios for all 1000 pairs in each layer for each entangled neuron. We then normalize the average z-score to the source layer for better visualization.

Appendix B. Empirical investigation of other neurons

In addition to details previously mentioned in Section 2.3, in Wasserstein neurons, many pairs with the highest OI ratio are near-duplicates—e.g., “the” vs. “The,” “to” vs. “the,” and “and” vs. “to,” suggesting the grammar and low-level functional importance of the role of Wasserstein neurons. Additionally, the appearance of punctuation-like tokens such as “-”, “.”, “(”, “@”, and “,” underscores layer two’s involvement in processing syntactic boundaries—marking clause breaks, parentheticals, enumerations, or formatting cues. Pairs like “The” vs. “the” and “A” vs. “a” further indicate sensitivity to casing and sentence position, implying that layer two distinguishes between identical tokens placed at the beginning of a sentence versus within it.

We provide additional examples of entangled neurons in layer two. Neuron 5224 is involved in processing basic grammatical structures, particularly with functional words like “the,” “and,” and “to.” Neuron 7723 focuses on token pairs involving numbers and specialized content. For example, it handles pairs like “the” and “6” and “16” and “be,” which likely relate to numerical, temporal, or domain-specific information. Neurons 1168 and 851 both seem to be focused on special characters and punctuation marks, with Neuron 1168 processing several pairs involving hyphens (“-”), highlighting a focus on segmentation of text, and Neuron 5 processing several pairs involving commas (“,”), periods (“.”), and parentheses (“(”), indicating a focus on sentence boundaries.

Neuron 5224

T1	T2	N1	N2
Ġthe	ĠThe	ĠT	Ġvice
ĠD	Ġthe	ia	Ġproblematic
ĠG	Ġof	ins	Ġ7
ĠA	Ġwas	Ġ70	Ġpushed
ĠO	Ġa	ya	Ġlist

Neuron 7723

T1	T2	N1	N2
Ġthe	Ġ6	ĠCross	Ġkm
Ġ	Ġ3	Ġthe	Ġrecord
Ġit	Ġ3	Ġmakes	Ġinnings
Ġ2	ĠIt	Ġkm	Ġis
Ġas	Ġ19	ĠTra	Ġ

Neuron 1168

T1	T2	N1	N2
-	Ġthe	@	Ġthreat
Ġa	-	Ġstudent	@
Ġis	-	Ġmore	@
Ġto	-	Ġmarch	@
-	ed	@	Ġ.

Neuron 851

T1	T2	N1	N2
Ġand	Ġto	Ġcrashed	Ġuse
Ġand	Ġto	Ġthe	Ġmove
Ġof	Ġand	Ġthe	Ġcommenced
Ġ(	ĠThe	Ġ9	ĠUnder
Ġ(	Ġthe	Ġ19	ĠDan

Figure 5: The highest OI ratio token pairs for entangled neurons.

Additionally, we show the input-output (IO) relationship for these and other entangled neurons in layer two, demonstrating the degree to which these pairs of tokens are mapped further than expected for the relative input distance.

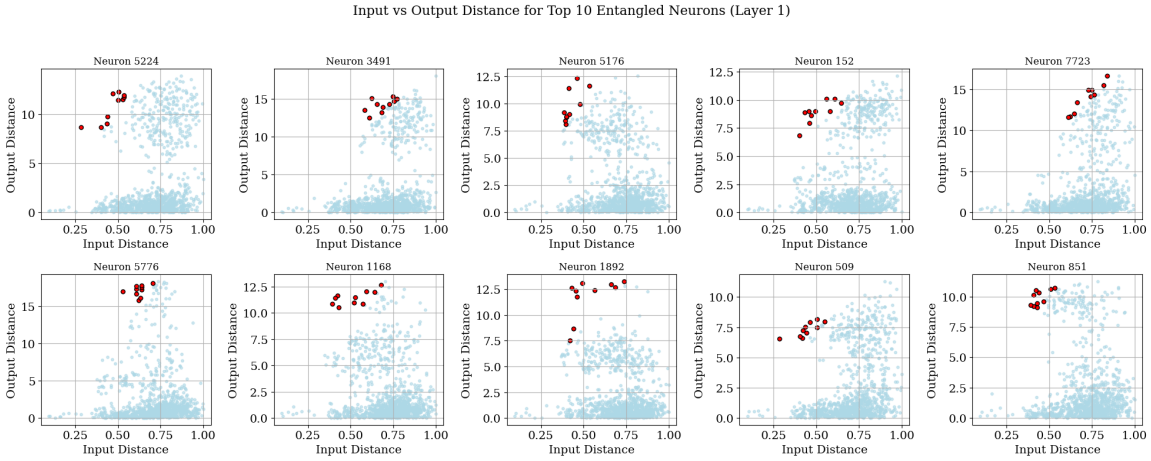

Figure 6: IO relationship of entangled neurons.

Appendix C. Wasserstein neurons in OPT-1.3B vs. Pythia-1.4B

Here, we show more examples of the neurons with the largest Wasserstein distances in the second up projection matrix in both Pythia-1.4B and OPT-1.3B. Note how much more complex the output distributions are for Wasserstein neurons in Pythia-1.4B than they are in OPT-1.3B, in particular in the negative pre-activation space.

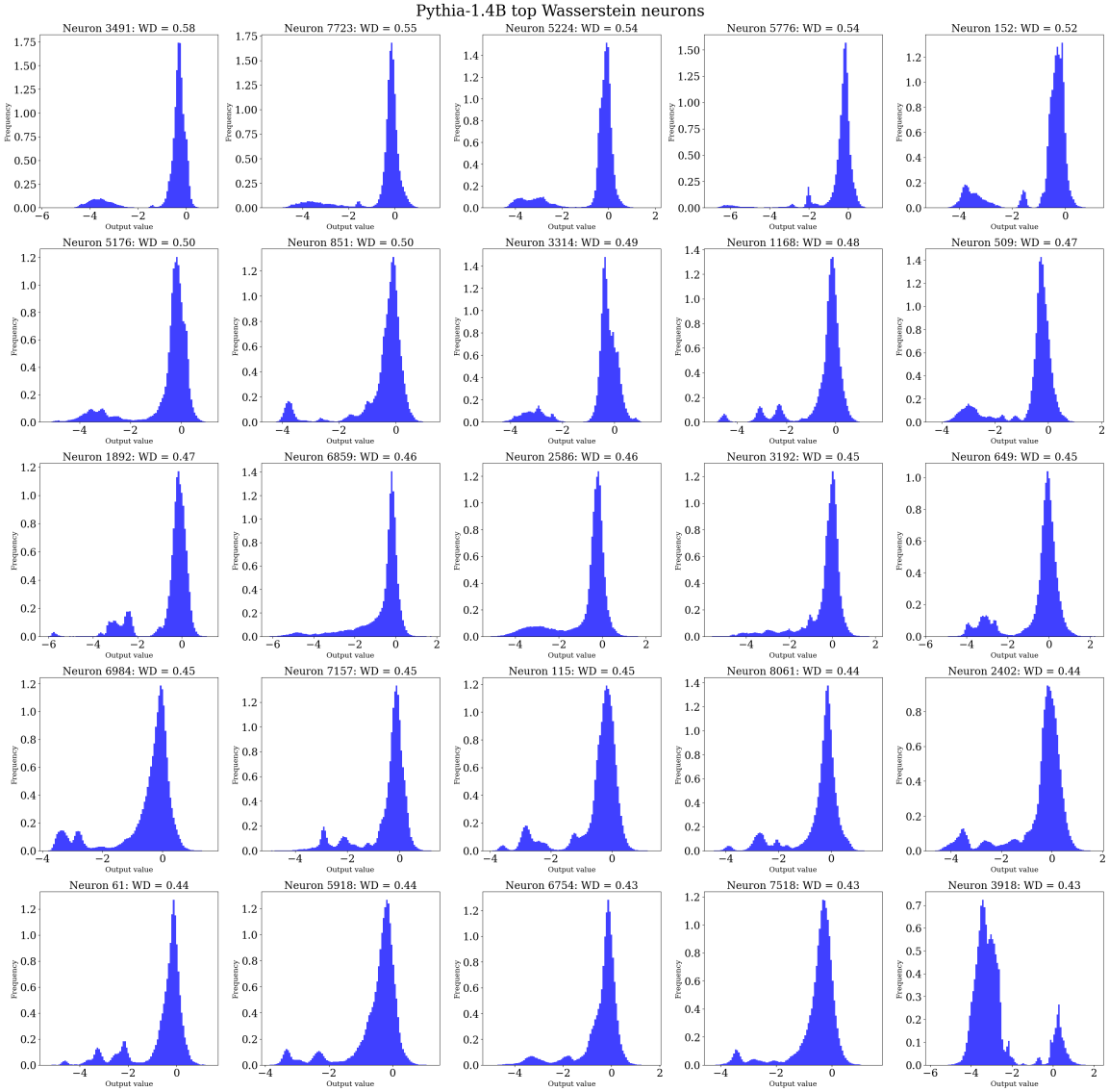

Figure 7: Top 25 Wasserstein neurons in Pythia-1.4B. Almost all neurons have complex and multimodal distributions exclusively within the negative pre-activation output space. Figure partially reproduced from [17] with permission from the authors.

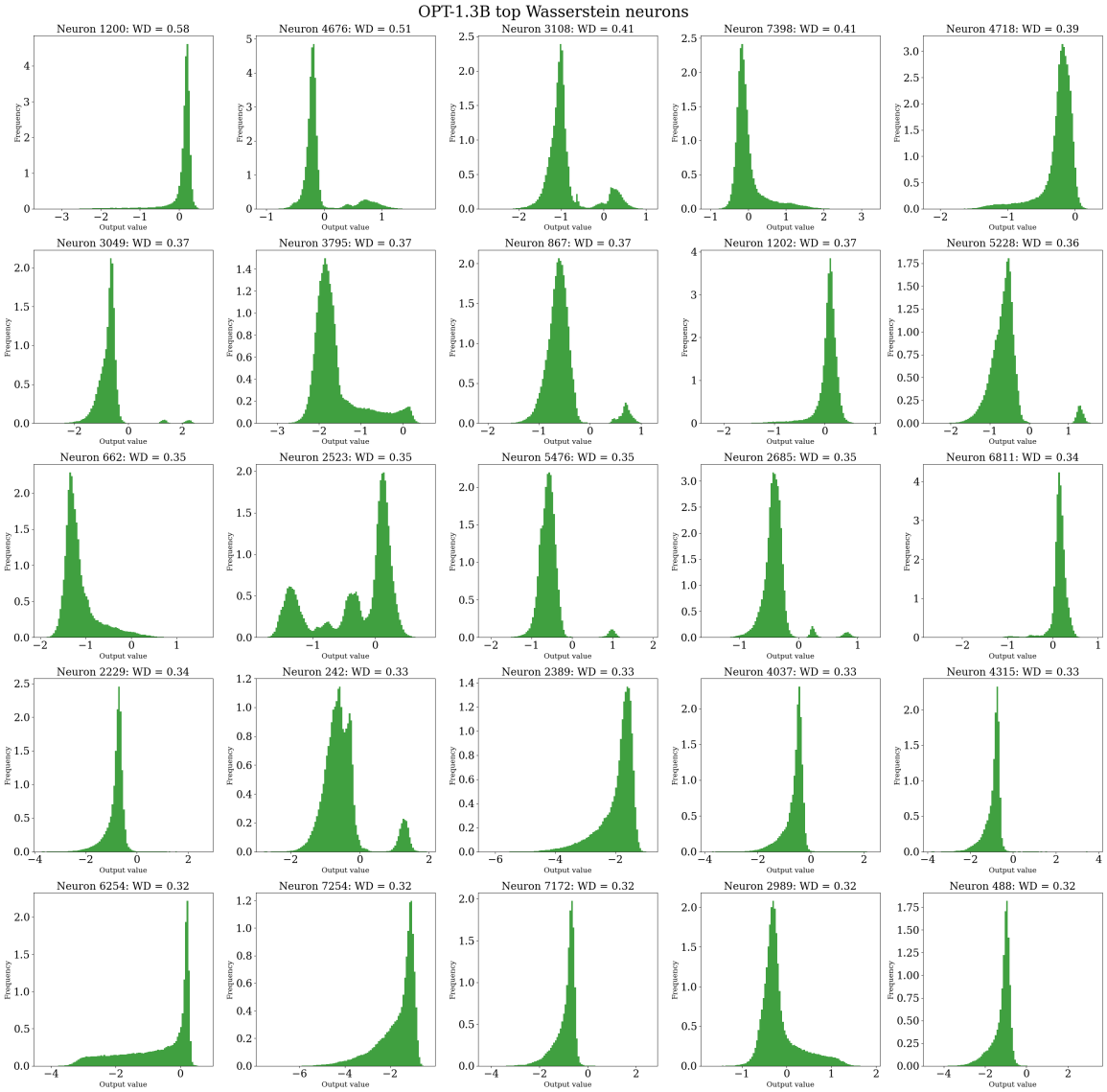

Figure 8: Top 25 Wasserstein neurons in OPT-1.3B. With one exception, there is no multimodality in the negative computation space of any Wasserstein neuron pre-activation.