
Supplementary Material for: Bounds on the computational complexity of neurons due to dendritic morphology

 **Anamika Agrawal***

Center for Data-Driven Discovery
Allen Institute
Seattle, WA 98109
Department of Neurobiology and Biophysics
University of Washington
Seattle, WA 98195
anamika.agrawal@alleninstitute.org

 **Michael A. Buice**

Center for Data-Driven Discovery
Allen Institute
Seattle, WA 98109
michaelbu@alleninstitute.org

1 Transformer analysis comparing attention-head architectures

To test whether the robustness-flexibility trade-off observed in single-neuron architectures extends to artificial systems, we adapted our *broad* (basal-like) and *hierarchical* (apical-like) motifs to the layout of attention heads within a transformer. The total number of heads was held constant across variants, but arranged in different structural patterns. Broad configurations implemented standard parallel multi-head attention (shallow and wide), while hierarchical configurations aggregated head groups in a tree structure (deep and narrow).

We evaluated these models on four sequence-learning tasks: **copying** (short-term memory), **reversal** (long-range dependency), **sorting** (comparison), and **pattern completion** (reasoning). Each model used $d_{\text{model}} = 128$, 3 layers, sequence length = 12, vocabulary size = 20, and was trained for 200 epochs with Adam ($\text{lr} = 5 \times 10^{-4}$). Convergence was measured as the number of epochs to stable validation loss. Transfer learning was assessed by fine-tuning pre-trained models on new tasks relative to training from scratch (negative values indicate degradation relative to baseline).

Architecture	Type	Avg. Loss ↓	Task Wins	Rank
[15, 1]	Broad	0.2004	2 / 4	1
[19, 1]	Broad	0.2296	0 / 4	2
[9, 1]	Broad	0.2394	1 / 4	3
[13, 1]	Broad	0.2410	1 / 4	4
[12, 6, 2]	Hierarchical	0.7617	0 / 4	5
[8, 4, 2]	Hierarchical	0.7625	1 / 4	6
[10, 5, 1]	Hierarchical	0.7687	0 / 4	7
[6, 3, 1]	Hierarchical	0.7796	0 / 4	8

Architecture-level comparison.

Architecture-type summary. Broad (basal-like) architectures achieved higher asymptotic accuracy across tasks, consistent with their greater parallel capacity and robustness to initialization. Hierarchical (apical-like) architectures converged faster and retained stronger performance under

*

Metric	Hierarchical	Broad	Winner
Average Loss ↓	0.768 ± 0.01	0.228 ± 0.02	Broad
Task Win Rate	25%	75%	Broad
Convergence Time ↓	149 ± 9 epochs	166 ± 11 epochs	Hierarchical
Transfer Benefit ($\Delta\%$) ↑	-11.5%	-21.5%	Hierarchical

fine-tuning, demonstrating improved adaptability and transfer. These trends mirror the robustness-flexibility trade-off identified in our neuron-level analyses: breadth favors stable task performance, while hierarchical depth promotes reusability and context-dependent adaptation. Although preliminary, these results illustrate how dendritic-inspired architectural diversity can enhance both stability and continual learning in artificial systems.

2 Supplemental Figures

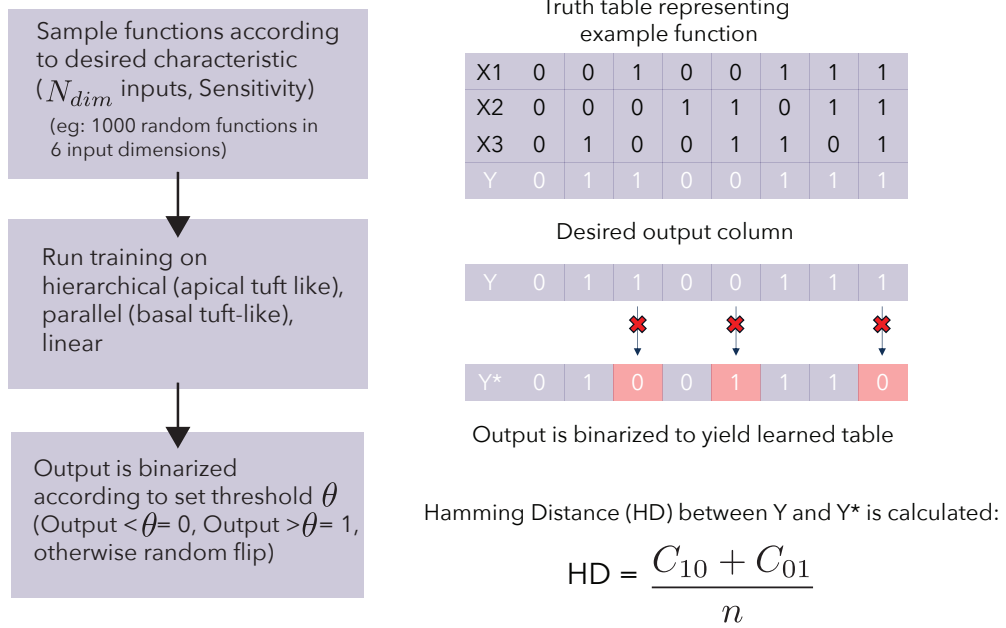


Figure 1: Schematic describing calculation of function learnability by an architecture through hamming distance of input truth table Y_{in} and binarized output Y_{out} . The hamming distance (HD) quantifying the mismatch between the objective Y and the learned output Y^* is defined as the sum of the number of 1-bits misclassified as 0 (C_{10}) and the number of 0-bits misclassified as 1 (C_{01}) divided by the total number of output bits.

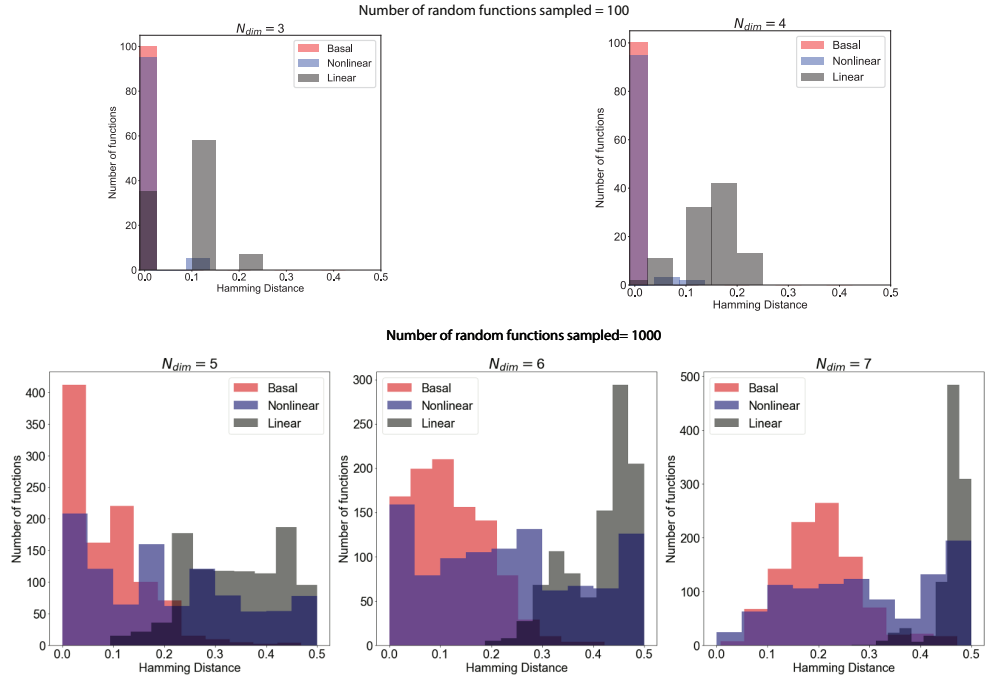


Figure 2: Performance Across input dimensions, highlights how linear neuron can't solve the typical functions. For $N_{dim} = 3, 4$; 100 random functions were sampled to be tested, for $N_{dim} = 5, 6, 7$, 1000 functions were sampled.

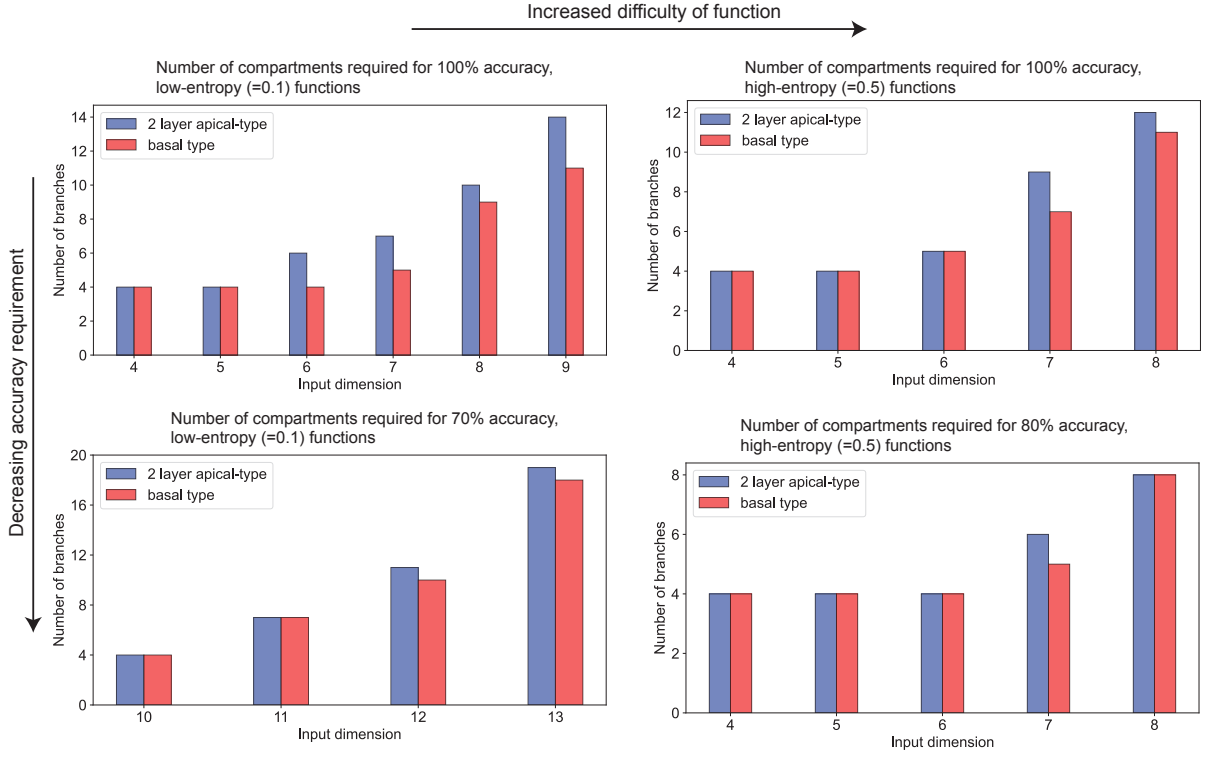


Figure 3: Compartment requirements scale exponentially. We tested the realizability of $n_{func} = 5$ randomly sampled functions in a given input dimension, with $n_{trials} = 5$ allowed to learn for each function. If an architecture is unable to learn any of the 5 functions in all of the trials, then the architecture is rejected and the compartment size is incremented by 1. There are 4 conditions tested: (i) low-entropy (low-difficulty) with high accuracy (ii) low-entropy with low accuracy (iii) high entropy, high accuracy (iv) high entropy, low accuracy

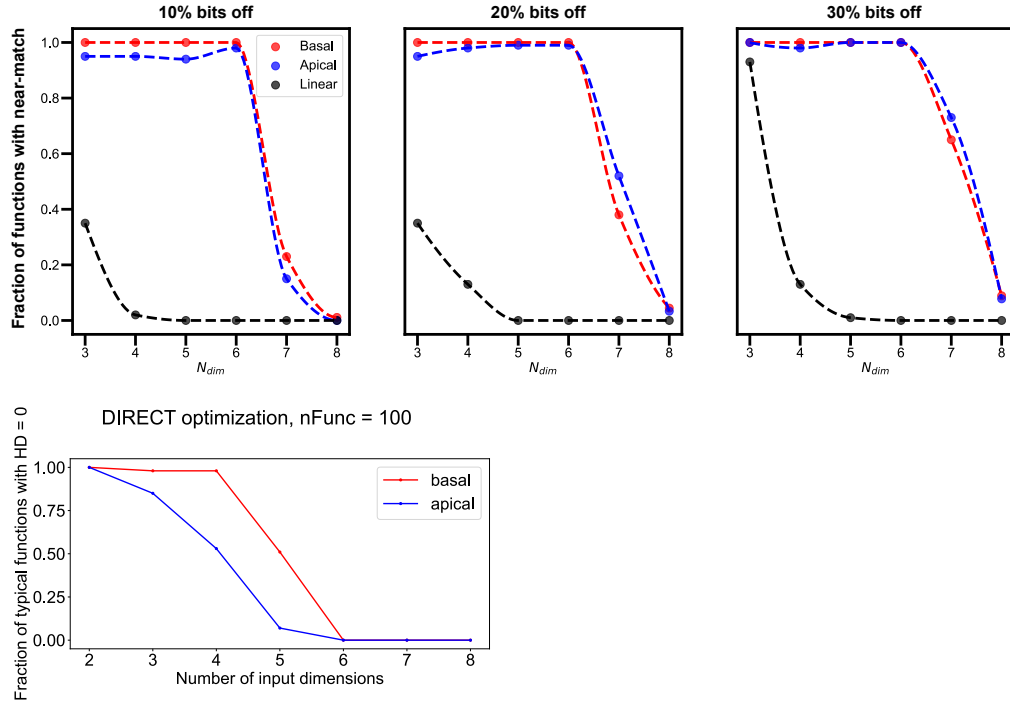


Figure 4: First row: The best case function realization fraction for each of the cell types, but with reduced requirements on accuracy. With reduced accuracy requirements. Second row: DIRECT based verification that there is a critical dimension beyond which functions become hard to realize, for both of the cell types.

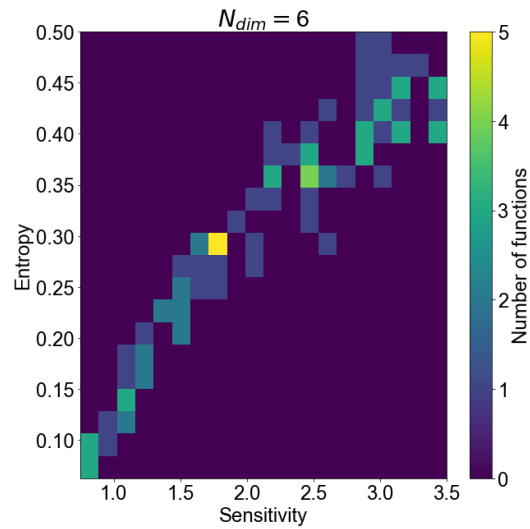


Figure 5: Entropy and Sensitivity have a monotonic relationship.

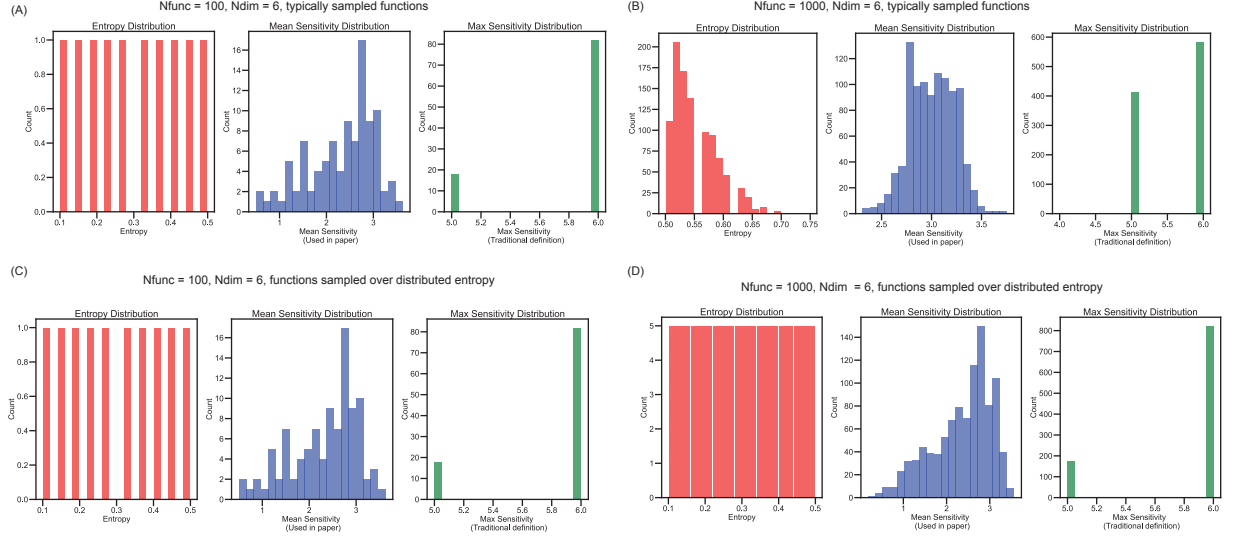


Figure 6: Figure demonstrates how the distribution of sensitivity changes from 100 typical functions to 1000 functions with distributed values of sensitivity, showing that the entropy-based sampling scheme successfully samples functions with a varying degree of sensitivity

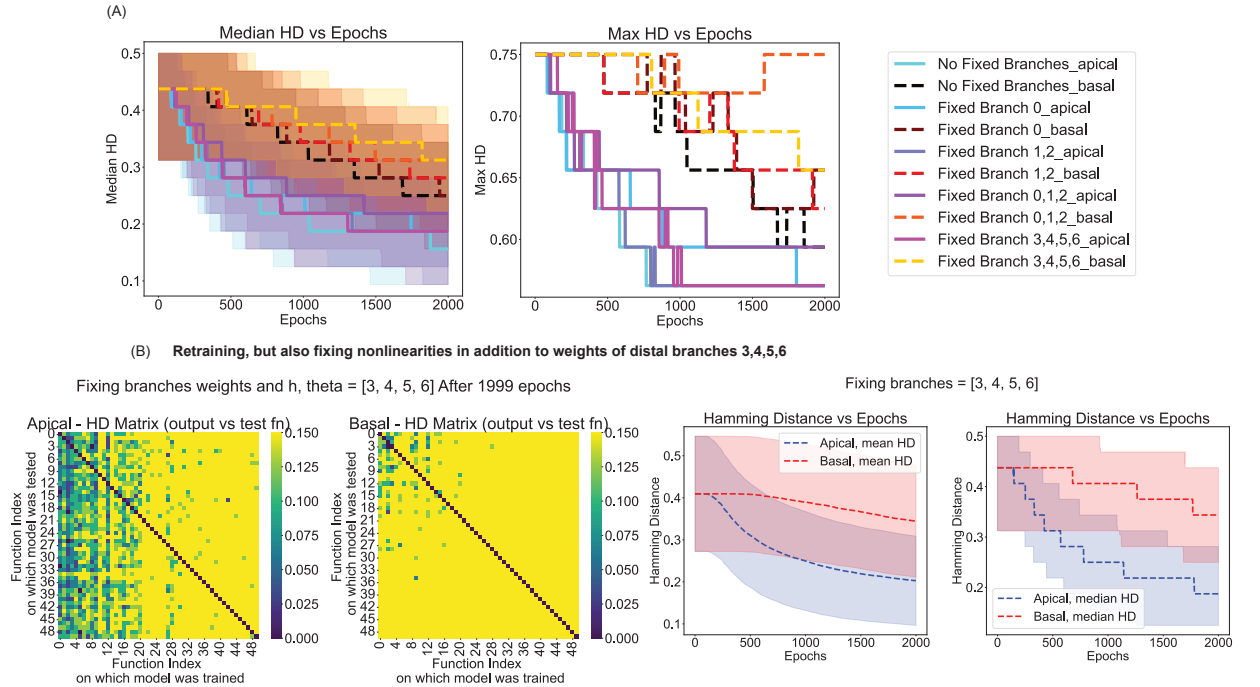


Figure 7: (A) Median and Max across j for $HD_{j \rightarrow i}(t)$ across t . (B) Flexible relearning is exhibited by apical type even when the non-linearities h_b and θ_b are held fixed for branches $b = 3, 4, 5, 6$ (most distal branches/shallowest layer) during retraining.

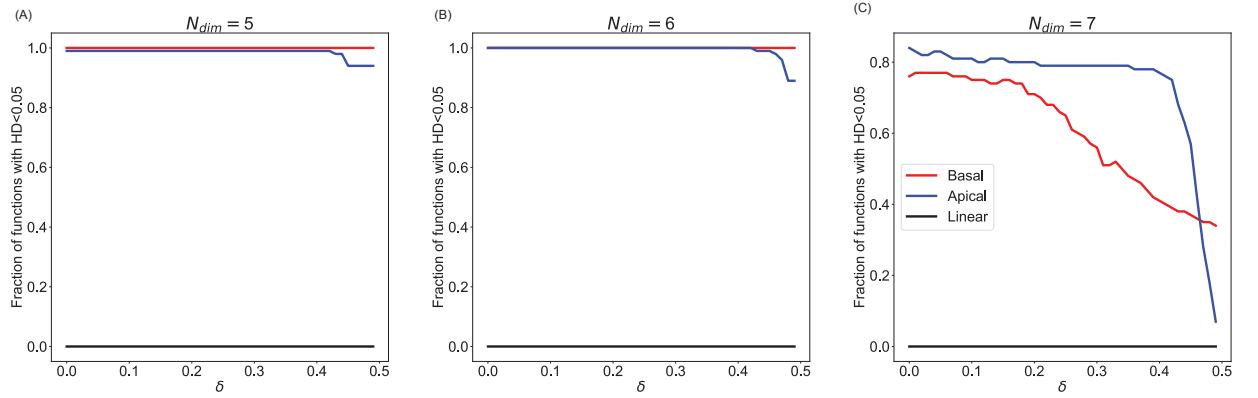


Figure 8: Changing δ (threshold for converting continuous output to binary output) changes the fraction of realized functions across dimensions. This threshold is referred to as θ in 1. At $N_{dim} = 7$ (the critical dimension for learning for both cells), the basal cell type is more sensitive to the threshold than the apical cell, highlighting two different methods of learning the boolean function mapping.

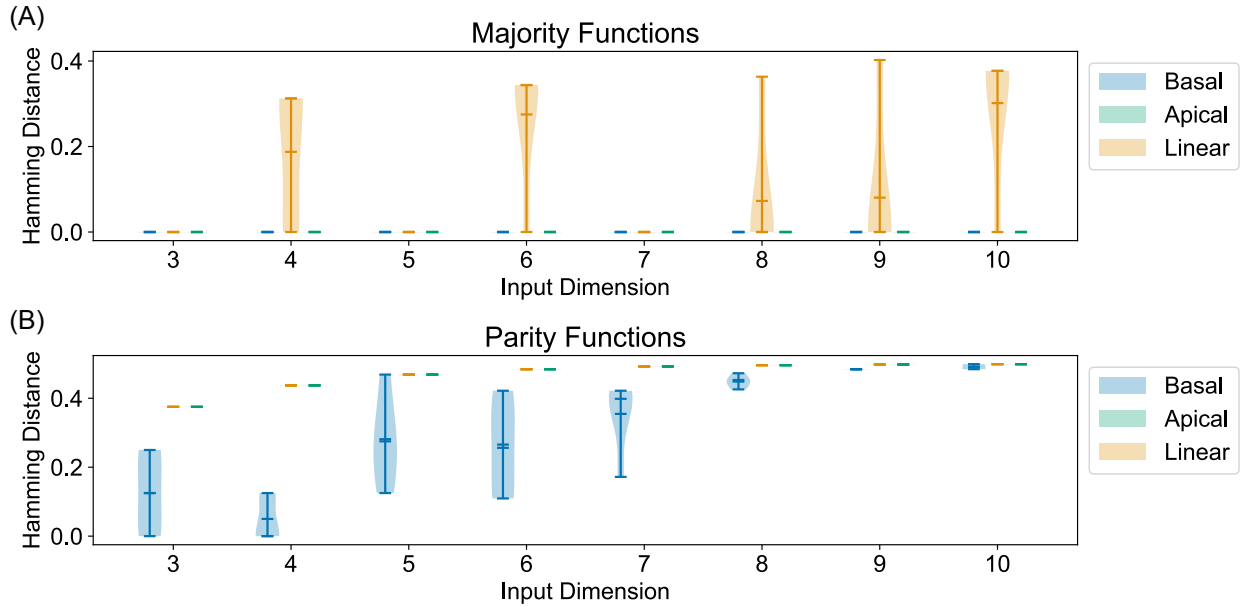


Figure 9: (A) Hamming distance while learning majority function (5 examples) in N_{dim} (x axis) (B) Hamming distance while learning parity function in N_{dim} (x axis) Performance on (A) majority function is disproportionately good and (B) parity is disproportionately bad - highlighting inability to directly relate boolean circuit complexity classes to neural network circuit complexity.