

Appendix

A More on the Starting Symbol

We have observed a significant difference in the generalization and interpretability of SA^+ and SA^- models. But it is not obvious how the addition of a starting symbol can cause such a change. We first point out that recognizing \mathcal{D}_1 is trivial, and a

one-layer SA (with or without the starting symbol) achieves perfect accuracy on this task. In addition, a two-layer SA^+ is not interpretable for \mathcal{D}_1 , as the task does not require learning to attend to the correct preceding token, but rather a simple counting mechanism suffices. Further, we found that two layers of SA are necessary for the recognition of $\mathcal{D}_{n>1}$ and the addition of more layers does not improve generalization, but rather degrades it.

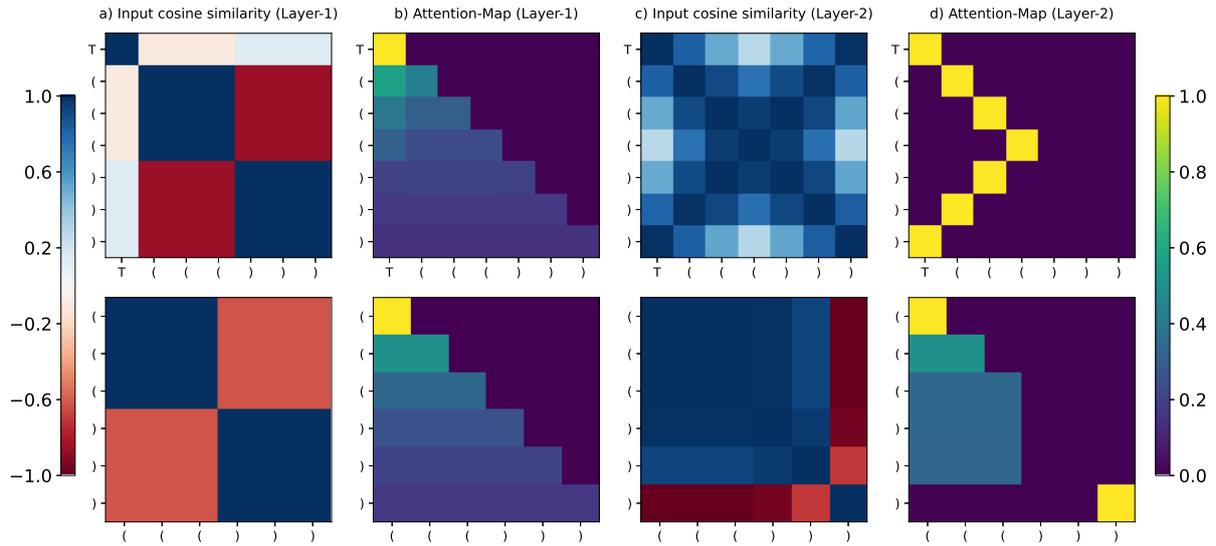


Figure 1: Contrasting SA^+ and SA^- on different parts of the network for the sequence “((((()))”. In a and c, we present the pair-wise cosine similarity of the symbols at layers 1 and 2. In b and d, we present the weights given by one of the attention heads.

We take our SA^+ and SA^- models trained on \mathcal{D}_4 , and contrast the representations learned across the two layers of the network. For presentational purposes, we use a simple string, “((((()))”. Figure 1 shows the cosine similarity between pairs of symbols in the sequence and the attention-maps for the first (out of 4) heads at the two layers of the networks. The input sequence at the first layer is simply the looked-up embeddings for each symbol, which are identical for all opening parentheses, and similarly identical for all closing parentheses. The embeddings for the opening and closing parentheses have negative cosine similarity: -0.77 for SA^- and -0.95 for SA^+ . Further, the starting symbol in SA^+ has a negative cosine similarity (-0.19) with the opening parenthesis and a positive cosine similarity (0.18) with the closing parenthesis. For both models, attention weights at the first layer are almost uniformly distributed across the preceding parentheses, opening or closing. This occurs because the input sequence to the first layer contains several identical representation for opening and

closing parentheses.

Beyond the first layer, the two networks behave radically differently. For SA^- , the the input representations to second layer have a cosine similarity close to or exactly 1.0, except for the last symbol. In contrast, the input representation for SA^+ is based on the head-dependency relationship. For instance, each opening parenthesis has the highest cosine similarity with its opening counterpart and the last closing parenthesis is matched with the starting symbol. Crucially, the starting symbol has enabled SA^+ to differentiate among the opening parentheses, which remain identical at layer-2 for the SA^- model. Both SA^- and SA^+ maintain opposite representations (negative cosine similarity) for opening and closing parentheses, which helps them emulate push/pop operations. But only SA^+ is able to refine representations at the second layer, such that it can match the correct pair of opening and closing parentheses.