A Appendix: Compositional generalization through meta seq2seq learning
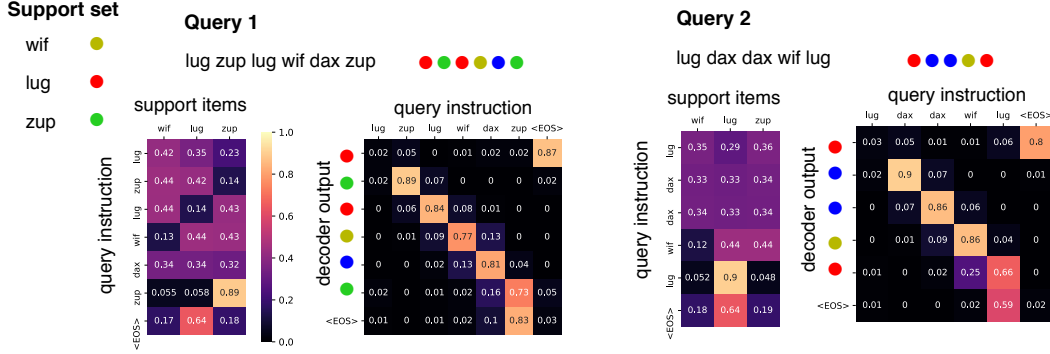

Figure A.1: During a test episode of the ME task, the support set (top left) and two queries are shown. The ME inference is that "dax" maps to "blue." The key-value memory attention A for each query is shown in the left matrix, with rows as encoder steps and columns as support items. The decoder attention for each query is shown in the right matrix, with rows as the decoder steps and columns as encoder steps. <EOS> marks the end-of-sequence.

Mutual exclusivity (ME). Attention is visualized in Figure A.1 for a test episode in the ME task (Experiment 4.2) with two queries "lug zup lug wif dax zup" and "lug dax dax wif lug." When passing a query symbol-by-symbol through the key-value memory (Figure A.1 left), the network allocates attention to all of the cells that *do not* contain the current query symbol, a counterintuitive but valid encoding strategy. This pattern is reversed in the last step before the end-of-sequence symbol (<EOS>), where more intuitively the input symbol activates the memory cell that contains its corresponding support item. The withheld ME symbol "dax" leads to a broad, uniform pattern of attention spread across the support items, indicating its novelty.

The RNN decoder attention is more straightforward. The diagonal pattern indicates strong alignment between each output symbol from the decoder (color; row) and its corresponding input symbol in the encoder (pseudoword; column). The first decoder step is an exception because the decoder hidden state is initialized with the last context step C_T (Section 3). The attention vectors do not sum to 1 because of padded elements from the batched decoder.
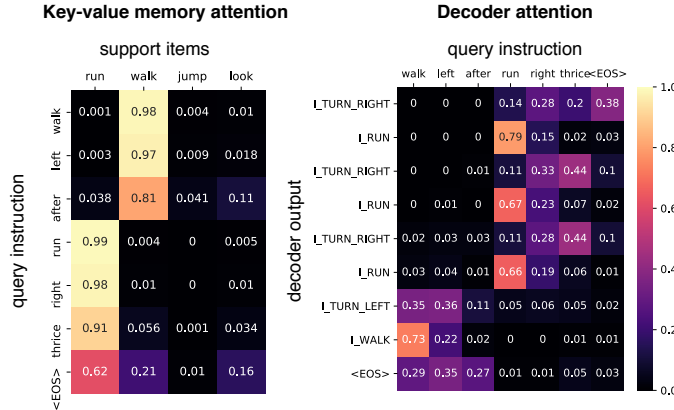

Figure A.2: Attention in meta seq2seq learning on the SCAN task. During test, the network is evaluated on the query "walk left after run right thrice." <EOS> marks the end-of-sequence.

SCAN. Attention is visualized in Figure A.2 for a test episode in the "add jump" task (Experiment 4.4). The key-value memory attention provides a lookup mechanism for retrieving the response for each input primitive, including "walk" and "run" (Figure A.2 left). The decoder attention also provides an intuitive alignment, attending to "run", "right," and "thrice" in alternation while executing "run right thrice" (Figure A.2 right).