A SCAN Tasks

For the sequence to sequence architecture, we use bidirectional LSTM as encoder, and unidirectional LSTM with attention as decoder. The first and last states of encoder are concatenated as initial state of decoder. The state size is $m = 16$ for encoder, and $2m = 32$ for decoder. For all SCAN tasks, the primitive embedding size k_p and function embedding size k_f are both 8. The weight for L_2 norm regularization λ is 0.01, and noise weight α is 1. We use Adam (Kingma and Ba, 2014) for optimization. We ran 10,000 training steps. Each step has a mini-batch of 64 samples randomly and uniformly selected from training data with replacement. We clip gradient by global norm of 1. Initial learning rate is 0.01 and it exponentially decays by a factor of 0.96 every 100 steps. We use TensorFlow (Abadi et al., 2016) for implementation.

SCAN dataset contains four sub datasets. Jump task contains 14,670 training and 7,706 test samples. TurnLeft task contains 21,890 training and 1,208 test samples. Simple task contains 16,728 training and 4,182 test samples. Length task contains 16,990 training and 3,920 test samples. We aim at Jump and TurnLeft tasks in the main experiments. Since Simple task does not require compositional generalization, and Length task requires syntactic generalization, they are beyond the scope of this paper. However, we still evaluate them to show that their performance is not significantly reduced.

B SCAN Template-matching

We extend experiments to SCAN template-matching task (Loula et al., 2018b). There are four tasks in this dataset. In *jump around right* task, the test set includes all samples containing "jump around right" (1,173 samples), and training set consists of the remaining samples (18,528 samples). In *primitive right* task, the test set includes all samples containing "Primitive right" (4,476 samples), and the training set consists of the remaining templates (15,225 samples). In *primitive opposite right* task, the test set includes all samples containing templates in the form "Primitive opposite right" (4,476 samples), and the training set consists of remaining templates (including their conjunctions and quantifications) (15,225 samples). In *primitive around right* task, the test set includes all samples containing templates in the form "Primitive around right" (4,476 samples), and the training set consists of remaining templates (15,225 samples).

For *primitive around right* task, we set $m = 8, k_p = k_f = 128, \lambda = 0.1$, and $\alpha = 0.1$. For other tasks, we use the same model configurations as SCAN Jump task.

C Primitive and Functional Information Exist in One Word

We constructed a dataset with both primitive and functional information contained in one word using the grammar in Table 9. The training data contains 2,560 samples, and test data 1,151 samples. For the proposed approach, we set $m = 32, \lambda = 0.1, \alpha = 0.3$, and we run 5,000 training steps. We keep other configurations the same as SCAN task. For comparison, we use standard LSTM with attention. The hyper parameters are the same as the proposed approach.

D Few-shot Learning task

For few-shot learning task, we set $k_p = k_f = 16$ and $\lambda = 0.1$. We keep other configurations the same as SCAN task.

The dataset (Lake et al., 2019) is shown in Figure 4. It contains 14 training and 10 test samples. Among them, 8 test samples correspond to Primitive task. The other 2 test samples requires syntactic generalization which is beyond the scope of this paper.

E Machine Translation

The experimental setting of machine translation follows Lake and Baroni (2018). The training data contains 10,000 English-French sentence pairs. The sentences are selected to be less than 10 words in length, and starting from English phrases such as "I am", "he is" and their contractions. The training data also contains 1,000 repetition of sentence pair ("I am daxy", "je suis daxiste"). Note that "dax" does not appear in the first set of training data. The two sets of training data are mixed and randomized. The test data contains 8 pairs of sentences that contain "daxy" in different patterns from training data, for example ("you are not daxy", "tu n'es pas daxiste").

The model configurations are similar to SCAN tasks, except that we set $m = 32, k_p = k_f = 32$, $\lambda = 1$ and $\alpha = 1$ in experiments. The result shows that some predicted outputs differ from the reference, but they are correct translations. Please see Table 10 for details.

S	→	V A N
A	→	C R M | C M R | R C M | R M C | M C R | M R C
V	→	push | pull | raise | spin
R	→	small | large
C	→	yellow | purple | brown | blue | red | gray | green | cyan
M	→	metal | plastic | rubber
N	→	sphere | cylinder | cube

Table 9: Commands for grammar in the extended experiment. The material of rubber only appears with other fixed words in training "push small yellow rubber sphere". However, it appears with other combinations of words in test.

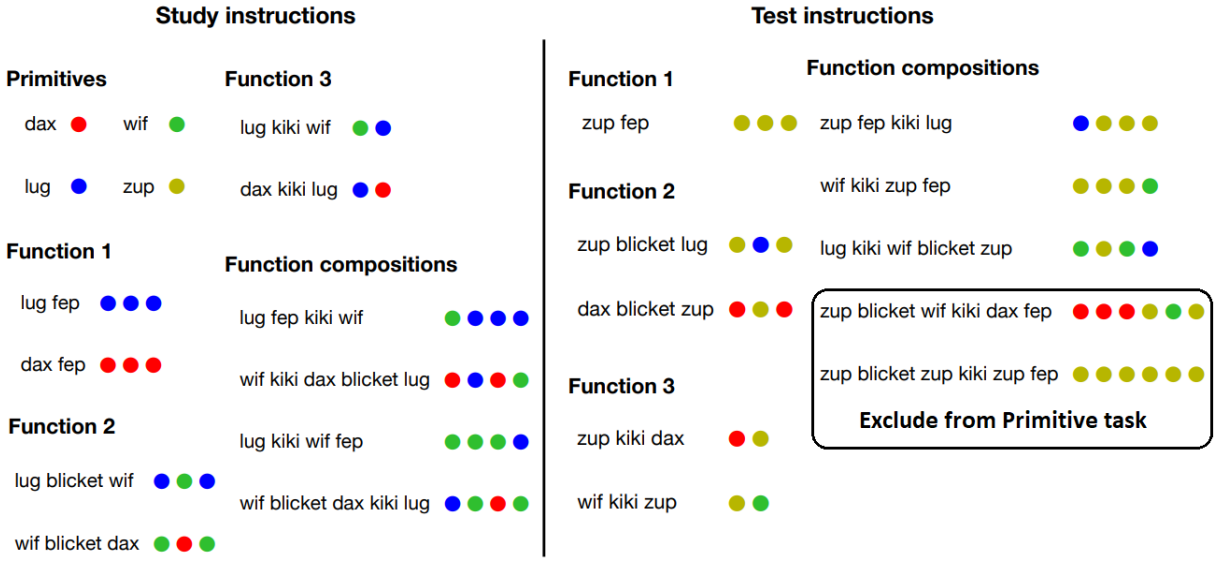

Figure 4: Few-shot instruction learning dataset. Participants learn to produce abstract outputs (colored circles) from instructions in pseudowords. Some pseudowords are primitives corresponding to a single output symbol, while others are function words that process items. A primitive ("zup") is presented only in isolation during training and is evaluated with other words during test. Participants are expected to learn each function from limited number of examples, and generalize compositionally. This figure is modified from Lake et al. (2019).

Input	Output
you are daxy .	tu es daxiste . vous etes daxiste .
you are not daxy .	tu n es pas daxiste . vous n etes pas daxiste .
you are very daxy .	tu es tres daxiste . vous etes tres daxiste .

Table 10: Test prediction mistakes in machine translation task (normalized). All mistakes are actually correct. Left part is input. Right part is output for reference (upper) and hypothesis (lower).