We thank all reviewers for constructive comments. We first address the common issue, then the individual questions.

Common issue: the importance of category prediction. Category prediction is required to determine the order to apply different parsing rules. We added an ablation study on the SCAN length split to demonstrate its importance. With category predictors, NeSS achieves 100% test accuracy in 5 independent runs. Without category predictors, NeSS still achieves 100% test accuracy in 2 runs; however, the accuracy is around 20% in the other 3 runs. The main reason is that without category predictors, the model may not learn that the parsing rule for "thrice" has the same priority as "twice". For example, in the test set, there is a new pattern "jump around right thrice" that does not appear in the training set. The correct translation is to parse "jump around right" first, then repeat the action sequence thrice, resulting in 24 actions. Without category prediction, a model could mistakenly parse "right thrice" first, concatenate the action sequences of "jump" and "right thrice", then repeat it for four times, resulting in 16 actions. This model could still achieve 100% training accuracy, because this pattern does not occur in the training set, but the test accuracy drops dramatically due to the wrong order for applying rules.

However, **category prediction alone doesn't guarantee length generalization.** A model can still fail by parsing rules incorrectly on longer test samples, e.g., they may predict "X Y" as the action sequence when the input is "X after Y". Recursion and sequence manipulation supported by NeSS are critical to learn such parsing rules to generalize. Based on the reviewers' suggestions, we will add more discussion and ablation studies.

R1. [Q: Few-shot learning and error bars?] We evaluated NeSS on the few-shot learning task in (Lake et al, 2019), and it again achieved 100% accuracy. Regarding error bars: for SCAN and context-free grammar parsing, the test accuracy of NeSS is 100% in 5 independent runs; for compositional machine translation, the exact match accuracy of NeSS is 100% in 2 runs, and 62.5% in 3 runs. As discussed in [29], only one reference translation is provided for each test sample, but there are 2 different French translations of "you are" that appear frequently in the training set, which are both valid translations. When the model predicts the alternative translation, the exact match accuracy becomes lower.

[Q: More discussion of related work?] Neural-symbolic methods for VQA consider the generalization to new composition of visual concepts, and scenes with more objects than training images. Compared to VQA, our tasks do not require visual understanding, but need much longer execution traces. Compared to Synth, which searches for >1000 programs on SCAN, models that do not need an extensive search during inference could be more efficient, especially when the task requires more examples for test-time input specification. We will add more discussion in our revision.

R2. [Q: All Components needed?] See the common response above for the importance of category prediction, and we will incorporate other writing suggestions in our revision. All operators are required for sequence manipulation using the stack machine. Curriculum learning is required for the model to find correct traces for long training inputs. When the encoder receives more information, e.g., the entire input sequence, the model does not utilize the recursion property of the stack machine anymore, and thus the generalization accuracy becomes similar to a seq2seq model.

R3. [Q: Experiments on naturalistic domains?] Our goal is to address the compositional generalization problem. However, most existing natural language benchmarks are not designed for this purpose. The challenge in those datasets is to handle the inherently ambiguous and potentially noisy natural language inputs. Their training and test sets are usually from the same distribution, and thus do not evaluate compositional generalization. Therefore, we did not run experiments on these datasets. Instead, our evaluation focuses on the standard sequence-to-sequence generation benchmarks used in previous works on compositional generalization. Such benchmarks are typically constructed with synthetic grammars, so that it is easier to change training and test distributions. We consider improving compositional generalization for more natural inputs as future work. We will clarify these points in our revision.

[Q: Trace search details?] When the current machine status is included in the rule set extracted from previous lessons, NeSS directly applies the rule, and only searches for other operations when it cannot find any consistent trace. We will discuss more details about the trace search and open-source the code in the final version.

R4. [Q: Comparison with differentiable data structures?] In Appendix E, we quoted the results of differentiable data structures on context-free grammar parsing from [9], denoted as Stack LSTM, Queue LSTM and DeQue LSTM. These results show that a stack alone is insufficient to obtain good results. We also evaluated these models on other benchmarks, and the results are similar to seq2seq. For example, stack LSTM achieves 100%/17%/0%/0.3% test accuracy on Simple/Length/Jump/Around Right splits of SCAN, though the training accuracies are always 100%. These results showed that without enhancing the stack machine with more operators for sequence manipulation, simply augmenting neural networks with a stack alone is not enough to achieve good generalization on these tasks. We will provide a more detailed discussion of the related works and the new results in our revision.

[Q: Data is over-sampled?] See common response for the discussion of accuracy improvement. Compared against prior works, we used the original training set, and we didn't do any additional sampling during training. The mapping from L_s to L_t is over sequences, and we will incorporate other writing suggestions in our revision.