RecT: A Recursive Transformer Architecture for Generalizable Mathematical Reasoning

Rohan Deshpande, Jerry Chen, Isabelle Lee

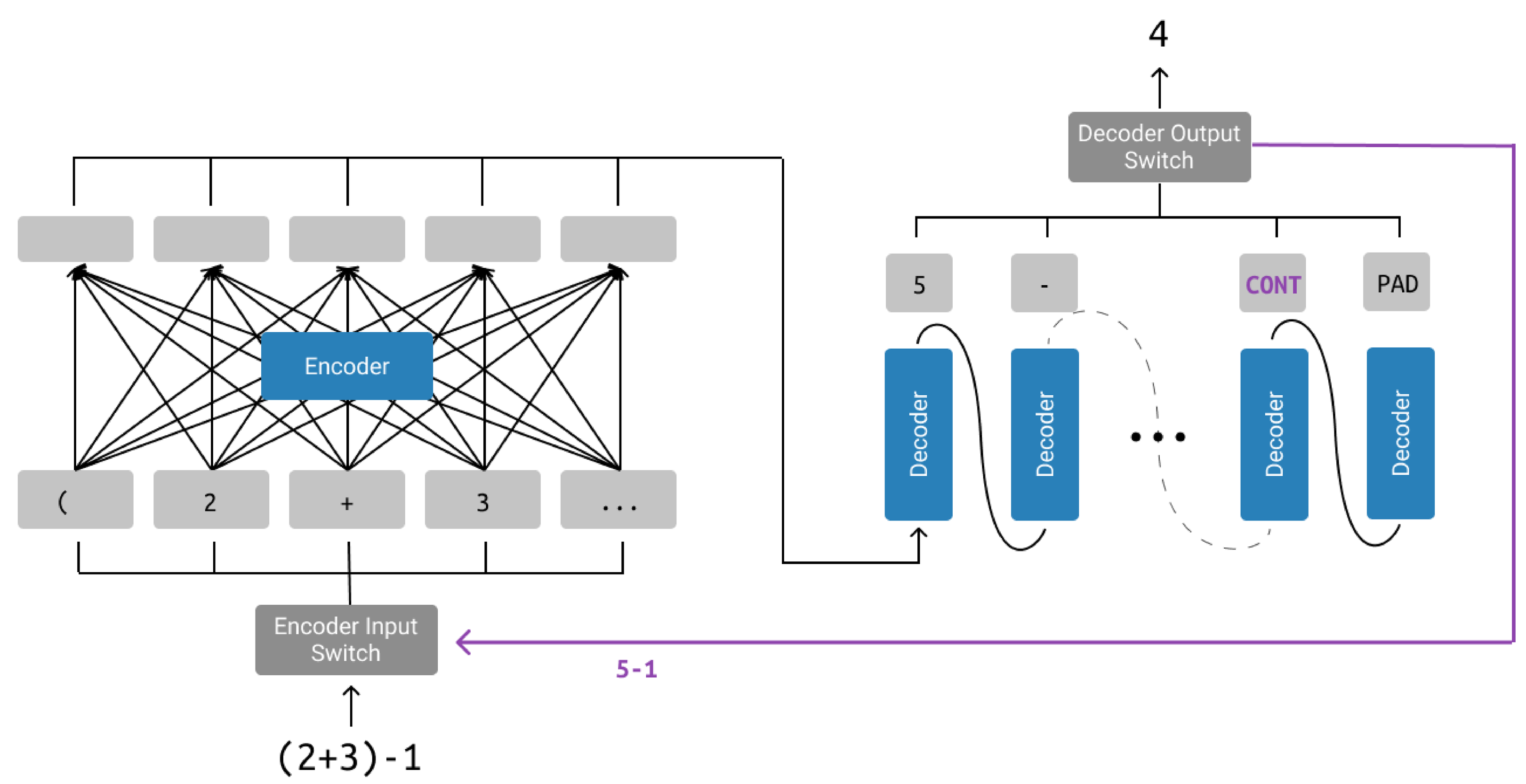
Stanford University Department of Electrical Engineering & Computer Science

Background

- Goal: Training neural language models to learn mathematical reasoning
- Past sequence-to-sequence models perform well when evaluated on test samples from the same distribution, but lack in extrapolation capabilities
- We aim to tackle generalization issues encountered by existing models by explicitly modeling recursion.

Approach

- RecT inherits properties from both recursive and transformer based language models
- RecT treats each recursive step independently, allowing us to teacher-force intermediate steps (leading to efficient computation).
- We introduce two special tokens, “Loop Continue” and “Loop End”, which allow the model to signal whether or not it should recurse.



Experiment Setup

- Curriculum-based approach where the models are first pre-trained on simple arithmetic.
- Leveraged the DeepMind Mathematics Dataset and complex multi-step arithmetic problems into increasingly “reduced” representations

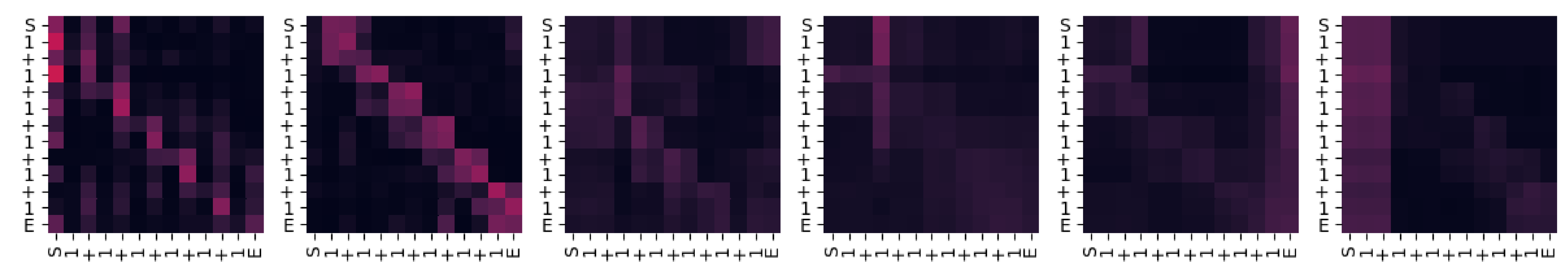
Randomized Padding

- Since we will be measuring model extrapolation, the length of problems at test time may exceed the length during training.
- This poses a challenge to models because they may have never learned to use positions beyond what was necessary in training.
- We resolve this issue by prefixing problems with a random number of padding tokens.

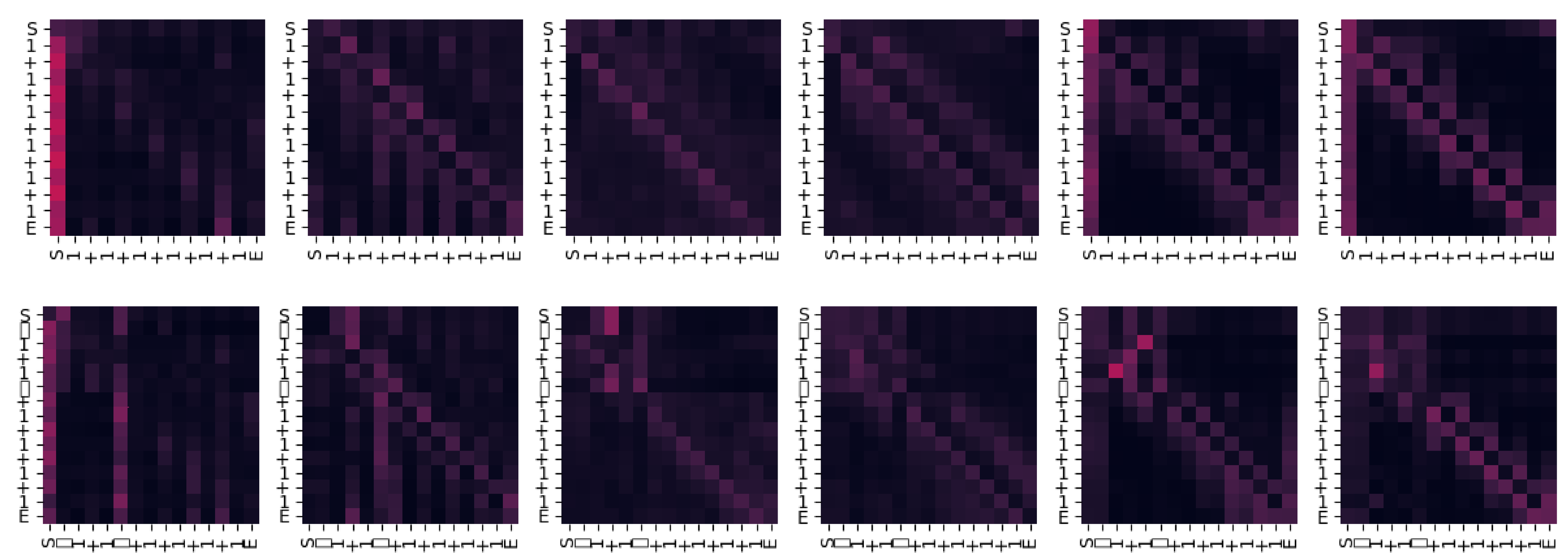
Subproblem Marking

- Originally, each teacher-forced intermediate step involved performing 1 computation and copying the rest of the arithmetic problem.
- However, it is difficult for models to differentiate between the computation and copy portions in this scheme when using cross-attention.
- We solve this by augmenting the token space: we create tokens which mark the beginning and end of a sub-problem. By doing this RecT, is taught to alternate between “marking” a valid sub-problem to solve and performing the computation.

Attention without subproblem marking



Attention with subproblem marking: marking and reduction steps



Results

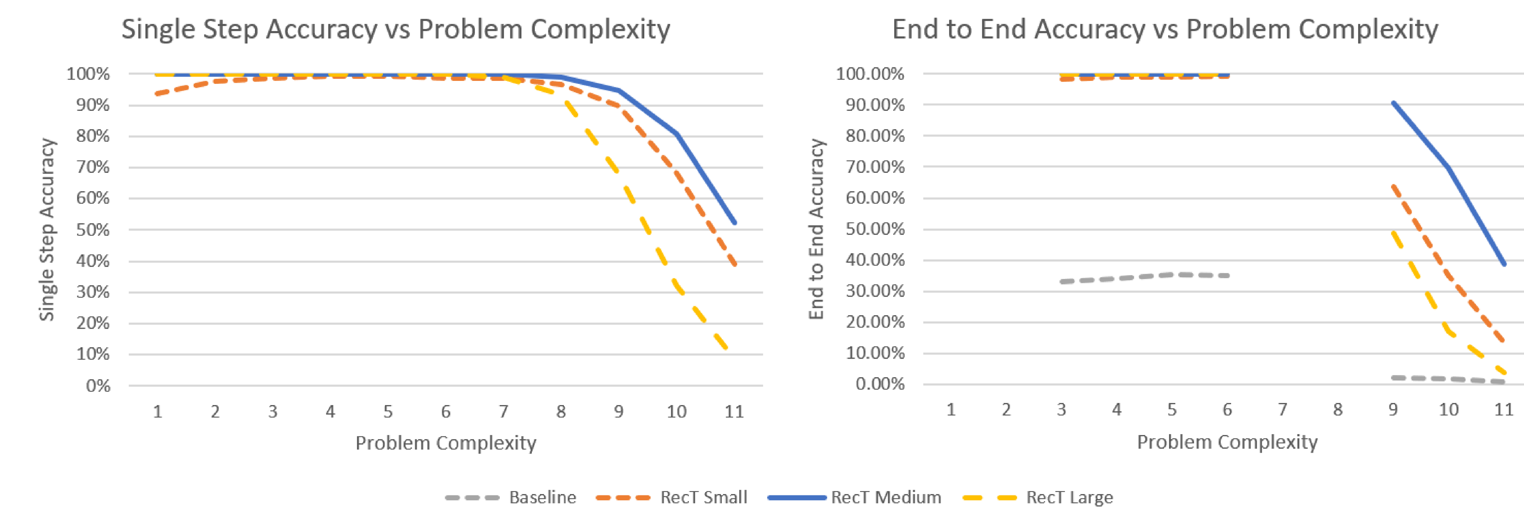
- Average accuracy improvement of 22.65% on extrapolation tasks when models are trained with sub-problem marking.
- Additional 20.53% improvement in extrapolation with randomized padding.

Results of RecT model variants.

Test Data	Variant	Per-Step Accuracy	End-to-End Accuracy
Simple addition/subtraction of two numbers, each up to 9 digits	RecT_Small	N/A	97.210%
	RecT_Medium	N/A	99.220%
	RecT_Large	N/A	99.820%
Interpolation: Multi-step addition/subtraction (between 3 and 6 operators)	RecT_Small	99.875%	98.890%
	RecT_Medium	100%	100%
	RecT_Large	99.998%	99.980%
Extrapolation: Multi-step addition/subtraction (between 9 and 11 operators)	RecT_Small	88.153%	37.350%
	RecT_Medium	96.426%	66.260%
	RecT_Large	88.372%	23.110%

Analysis

- RecT models significantly outperform the baseline (single-shot) transformer model.
- RecT_Medium achieves 90% E2E accuracy on 9-operator problems despite not having seen problems with more than 6 operators during training.
- There is a significant drop-off in accuracy for 11-operator problems. We hypothesize that increasing diversity in problem complexity during training would yield even better generalization capabilities.



Common Errors Made By RecT Models:

- Carrying mistakes leading to off-by-one errors in some digits
- Dropping parentheses that are unrelated to the current sub-problem
- Double negatives sometimes causing the model to subtract rather than add

Conclusion

- We have developed and validated a strongly supervised recursive transformer which was trained on complex arithmetic computation.
- The RecT model acheives 100% accuracy on out-of-sample interpolation dataset and 66.26% on extrapolation (previous studies report ~0% on extrapolation).
- RecT provides increased interpretability by outputting a proof of work (since it it explicitly computes each intermediate computation step)

Future Work

- Investigate weakly supervised approach (where intermediate steps are not teacher-forced)
- Experiment with other mathematical problem types