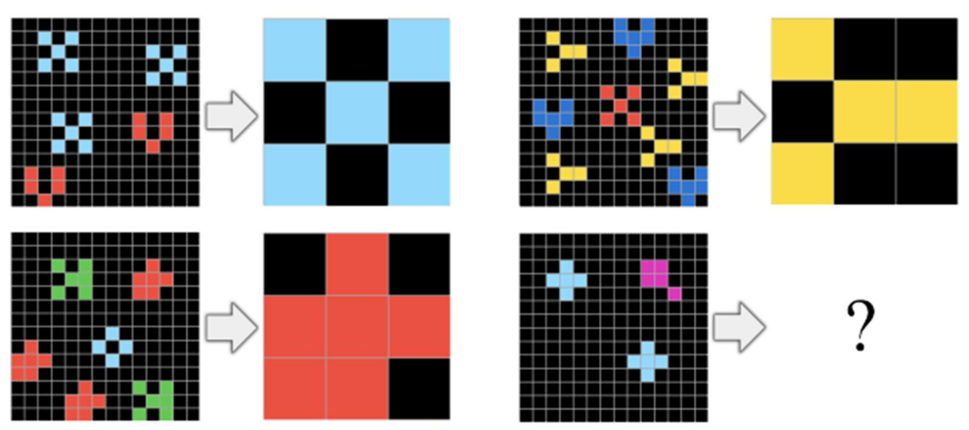
Improving Transformation Search In BEN

Daria Condratov | Supervised by Sebastijan Dumancic and Dekel Zak

contact: D.Condratov@student.tudelft.nl

1. Introduction

- **Program synthesis** => problem of automatic generation of programs based on a specification
- The **Abstraction and Reasoning Corpus** (ARC) => infer rule from a few examples (difficult)



- **BEN** (Divide, Align and Conquer) => program synthesis system for ARC
- **Transformation search** in BEN:
  - ❖ Finds programs that transform input in output
  - ❖ Grammar with 11 primitives
  - ❖ Uses exhaustive enumeration
  - ❖ Space **grows exponentially** with depth

2. Research Question

How can we better find the necessary transformation programs?

- Can grammar pruning make the transformation search more efficient?

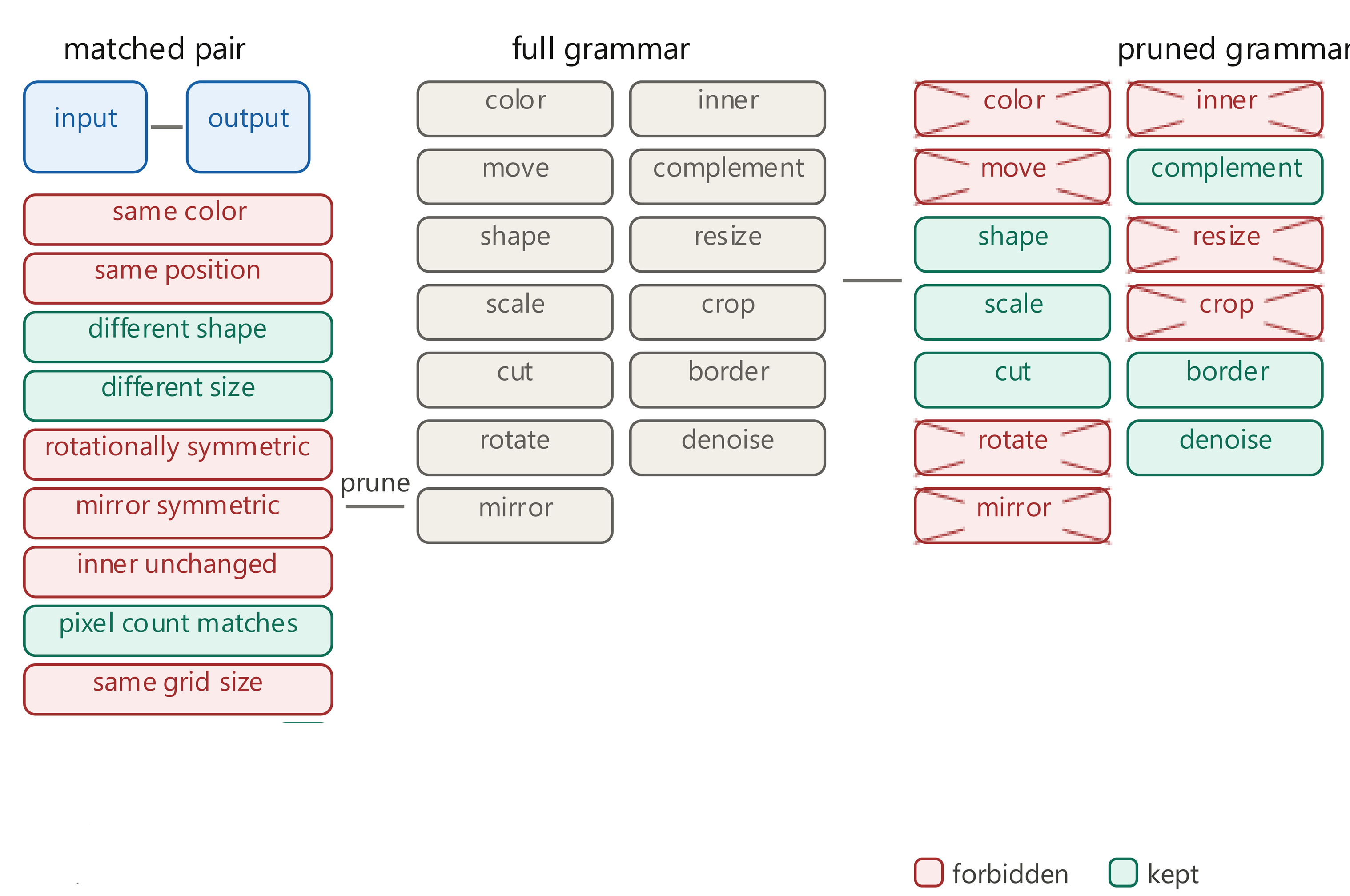
3. Methodology

Transformation search infrastructure:

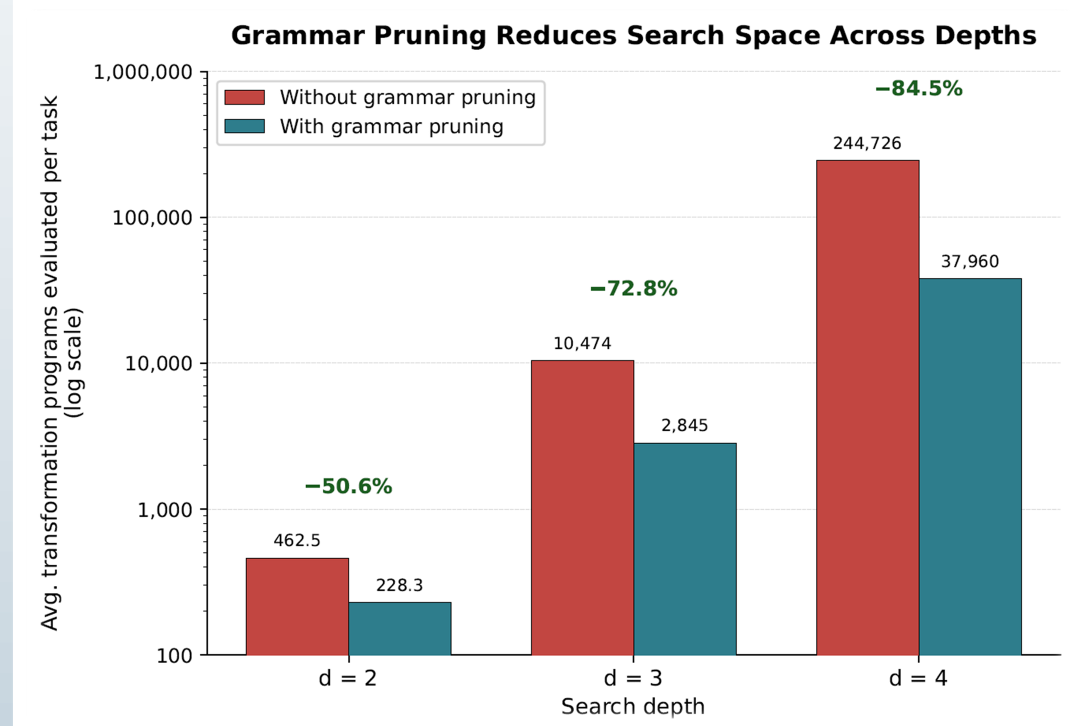
- Transformation search step from BEN in Julia using the Herb library
- Added 2 extra primitives for grid size change
- Breadth-first **exhaustive enumeration** over 13 primitives
- Color, move, shape and resize parameters added dynamically to the grammar
- Sends **all viable programs** to the concept learner

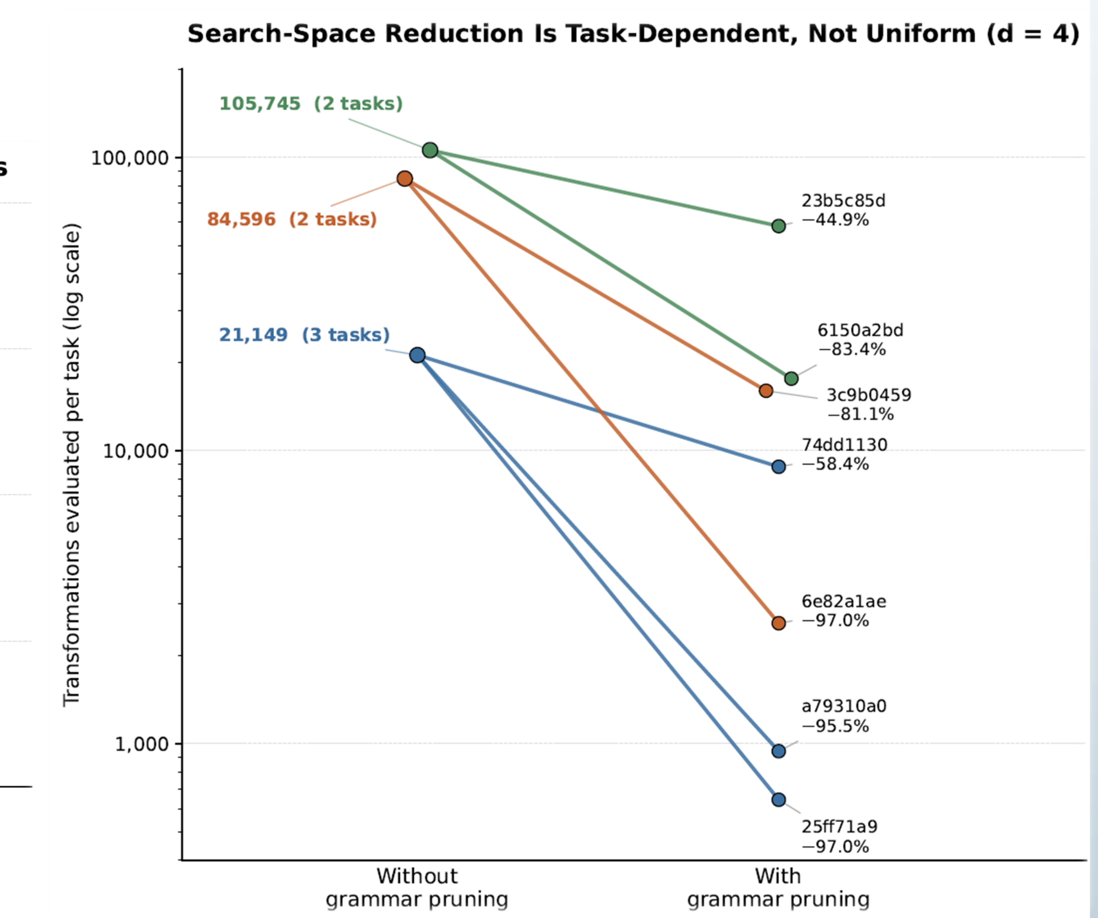
Optimizations

- Prune the grammar based on the similarities before the search begins
- Pruning rules for 11/13 primitives



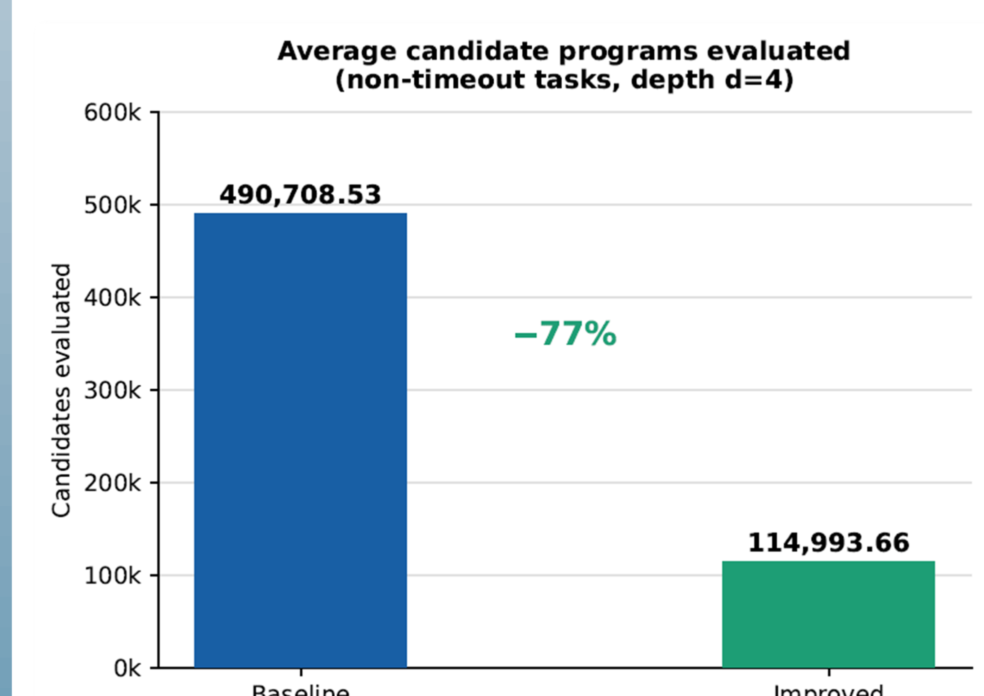
4. Results

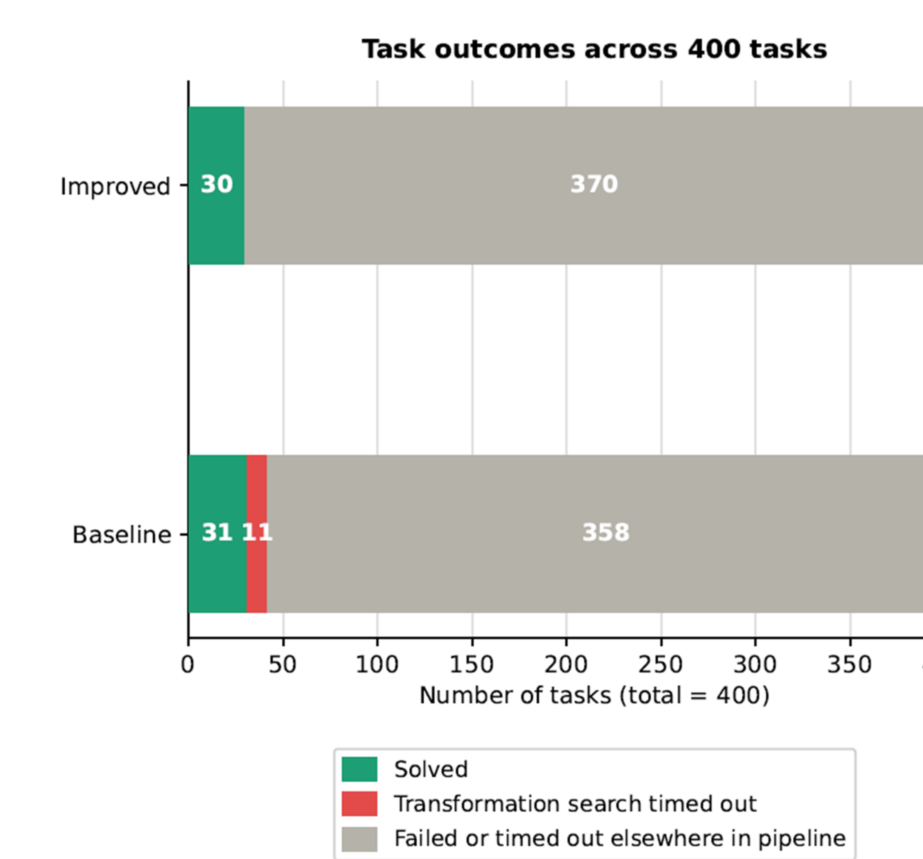




Rules contribute unequally:

- Consistent with the grammar space and how strict the rules are





- The correct programs are not pruned

5. Conclusion and Future Work

- Pruning **substantially reduces** search space
- Effect grows with depth
- **Why** the **regression** in solve rate?
- More pruning rules and/or primitives