*lil*Gym: Natural Language Visual Reasoning with Reinforcement Learning

Anne Wu, Kianté Brantley, Noriyuki Kojima, and Yoav Artzi

https://lil.nlp.cornell.edu/lilgym

***lil*Gym is a new RL benchmark for studying natural language visual reasoning with interaction**

- **Human-written highly-compositional** natural language with diverse reasoning challenges
- Each statement describes a **large set of valid goal states**, creating a reward computation challenge
- Policy must learn **complex language-conditioned goal equivalence** to generalize well
- Contemporary methods show non-trivial performance, but remain far from human performance

Goal: Manipulate **an image** to satisfy **a target boolean value** with respect to an **input statement**

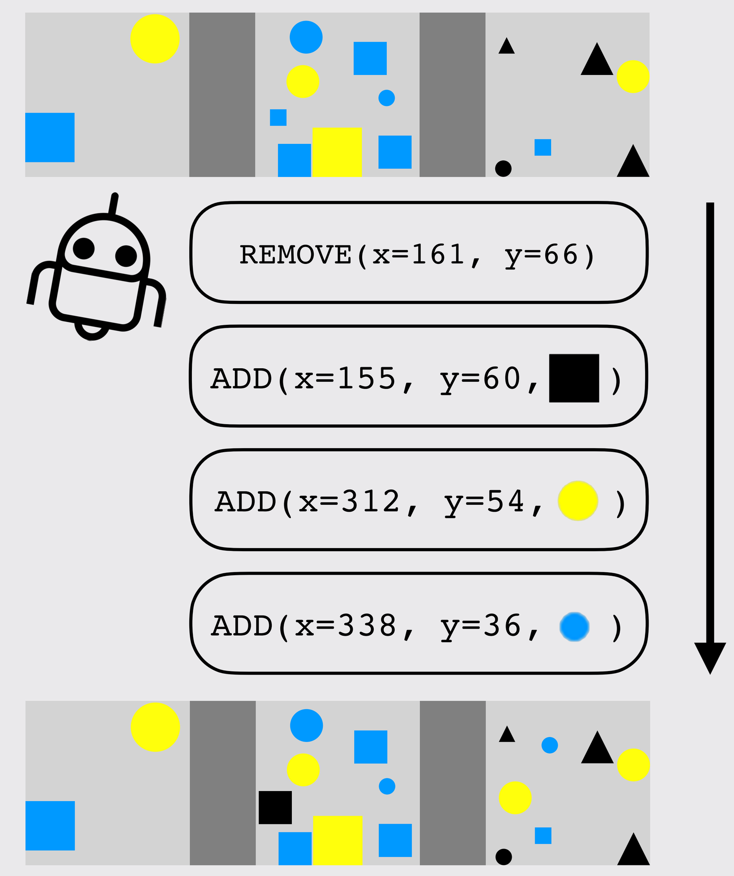
Statement

There are two boxes which has the same number of objects and three kinds of colors.

Target boolean

TRUE

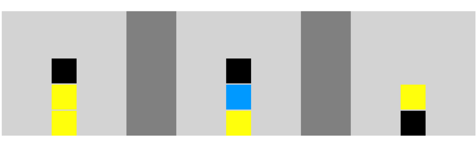
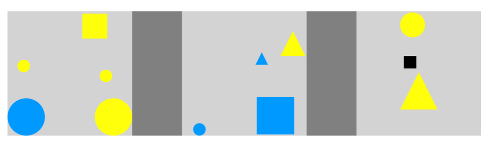
Rollout



Designing *lil*Gym: Making NLVR Interactive

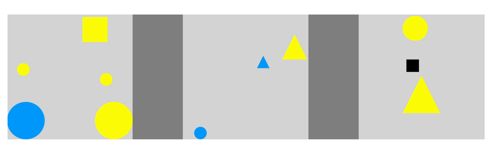
NLVR (Suhr et al. 2017) is a supervised benchmark for visual reasoning with rich compositional reasoning

Two Types of Environments

Tower	Scatter
	
ADD(BOX, COLOR) REMOVE(BOX) STOP	ADD(x, y, SHAPE, COLOR, SIZE) REMOVE(x, y) STOP
13 actions	1,064,001 actions Optional grid simplification

×

Two Types of Starting States

Scratch	FlipIt
One of the grey box has exactly six objects **TRUE**	*There is no square closely touching the bottom of a box* **FALSE**
	

- Start from an empty state
- Goal: satisfy statement
- Target boolean: always true

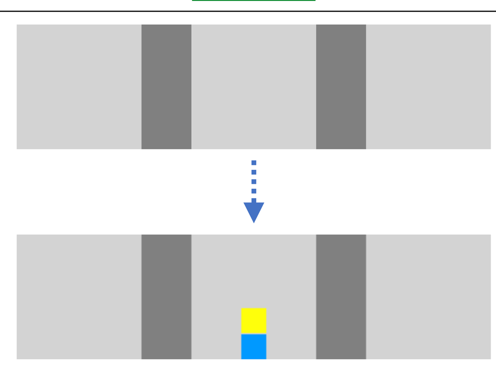
- Start from an NLVR image
- Goal: flip boolean value
- Target boolean: true or false

Four Configurations

Tower-Scratch

There is a blue block as the base of a tower with only two blocks

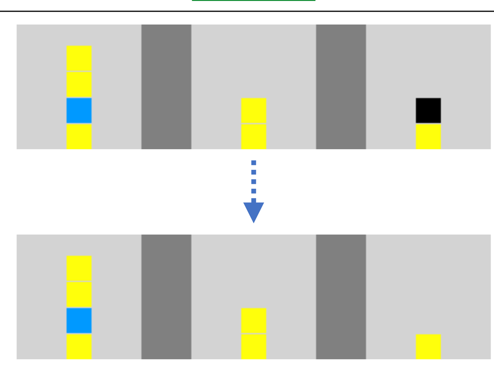
TRUE



Tower-FlipIt

There is no black block as the top of a tower with at most three blocks

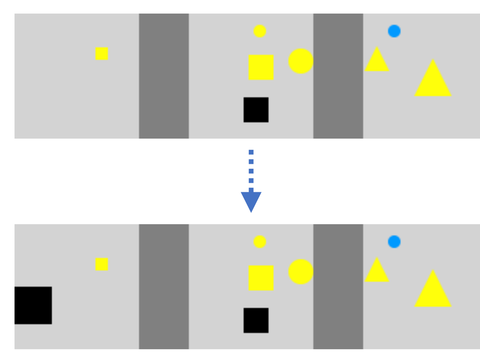
TRUE



Scatter-FlipIt

There is a grey box where none of the black objects are touching the edge

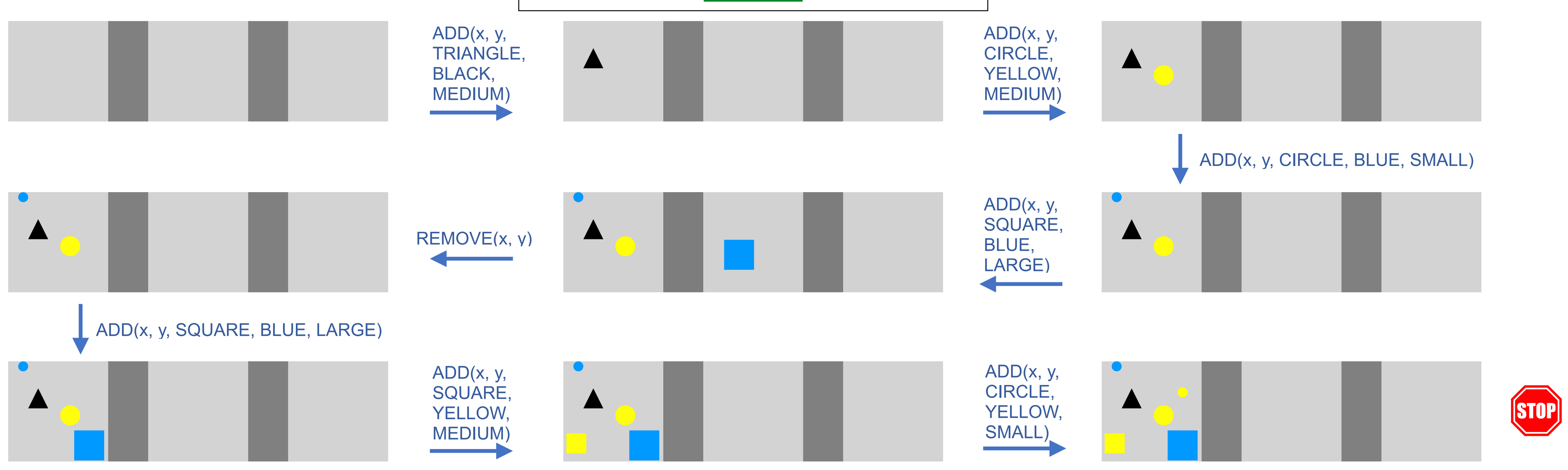
FALSE



Scatter-Scratch

One of the grey box has exactly six objects

TRUE



Annotation for Reward Computation

Key challenge: evaluating a state's correctness requires resolving language meaning

Solution: annotate all 2,661 statements with executable formal meaning representations

Example of Language Statement Annotation

Statement:

the grey box with least number of objects contains only one black object

Python program annotation:

```
count(filter_obj(min(
        all_boxes,
        key=lambda x:
            count(x)).all_items_in_box(),
    lambda y: is_black(y))) == 1
```

More in the paper

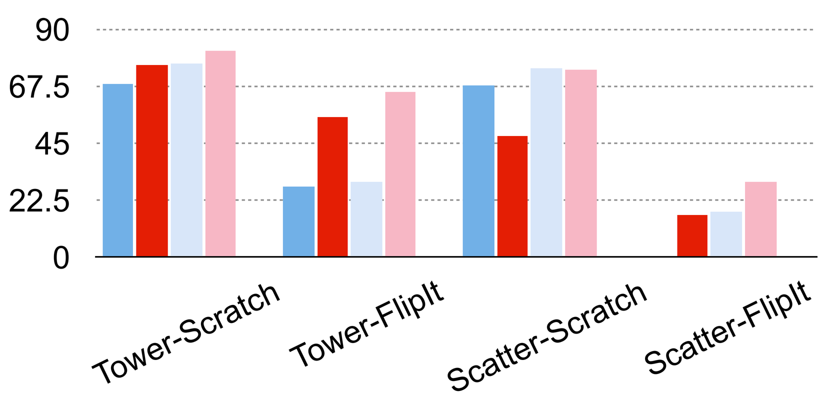
- Finer-grained analysis and rollout statistics (trajectory length, action types, etc.)
- Semantic and syntactic analysis
- Analysis of error patterns

Experiments and Results

- **Models**: C3+BERT (CNN + BERT) and ViLT
- **Learning algorithms**: PPO and PPO+SF
- **Scatter**: simplified 5×19 grid

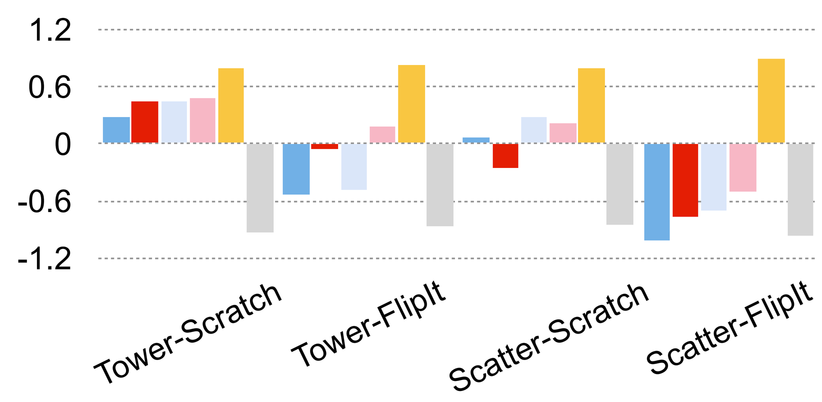
PPO w/C3+BERT · PPO w/ViLT · PPO+SF w/C3+BERT · PPO+SF w/ViLT · Expert · Random

Accuracy



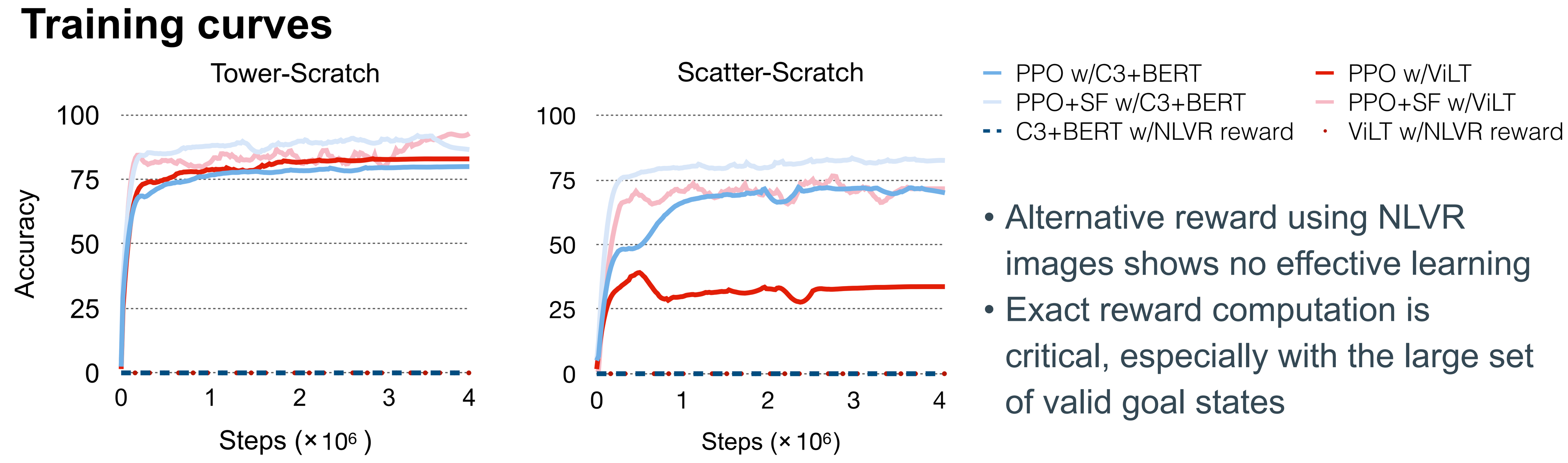
- Various levels of difficulty
- Overall: non-trivial performance
- Stop forcing (PPO+SF) helps: illustrates the exploration challenge

Mean Reward



- Reward shows similar trends to accuracy
- Gap to expert illustrates a long way to go
- Random policy: shows task difficulty

Training curves



- Alternative reward using NLVR images shows no effective learning
- Exact reward computation is critical, especially with the large set of valid goal states