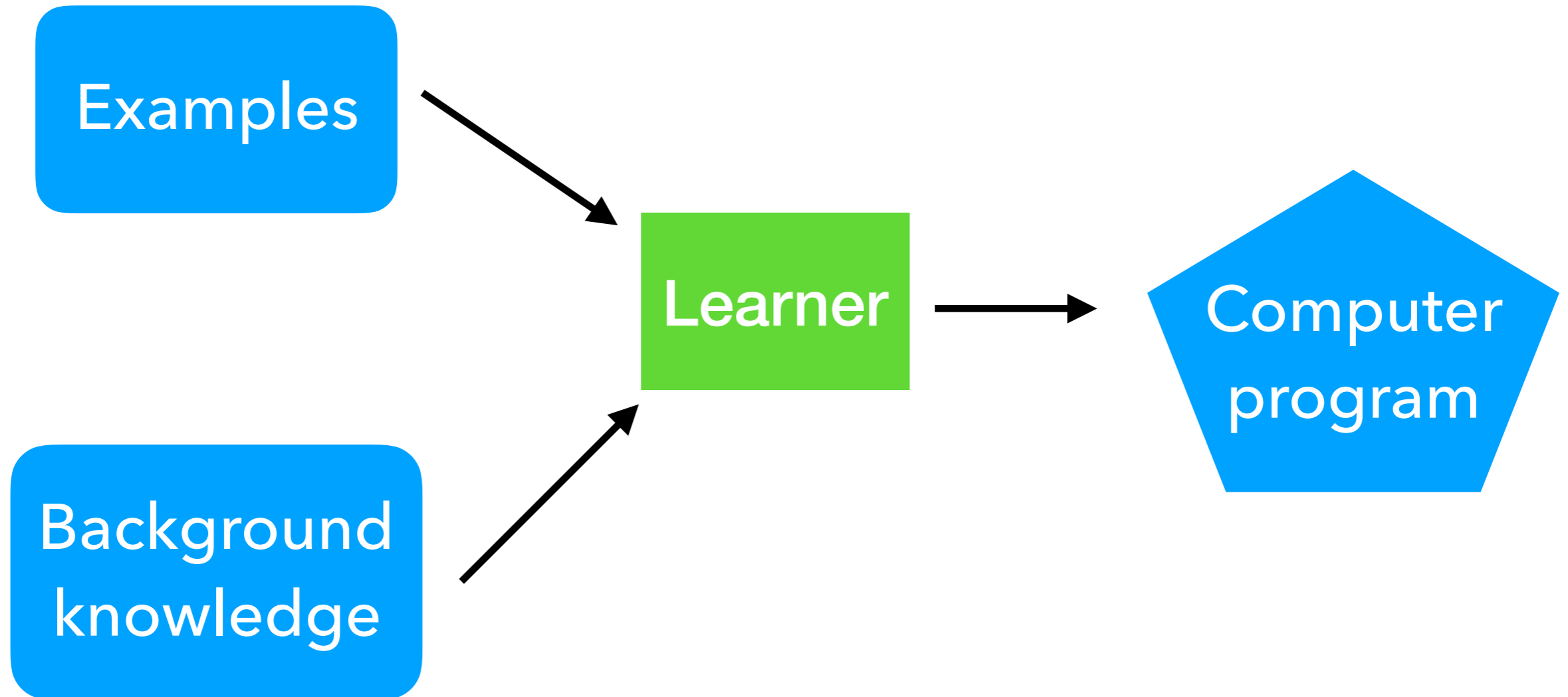


Learning programs through play

Andrew Cropper

Program induction



Examples

input	output
dog	g
sheep	p
chicken	?

Examples

input	output
dog	g
sheep	p
chicken	?

Background knowledge

- head/2
- tail/2
- empty/1

Examples

input	output
dog	g
sheep	p
chicken	n

Background knowledge

- head/2
- tail/2
- empty/1

```
def f(a):  
    t = tail(a)  
    if empty(t):  
        return head(a)  
    return f(t)
```

Examples

input	output
dog	g
sheep	p
chicken	n

Background knowledge

- head/2
- tail/2
- empty/1

$f(A, B) : \text{-tail}(A, C), \text{empty}(C), \text{head}(A, B).$
 $f(A, B) : \text{-tail}(A, C), f(C, B).$

**Where do we get background
knowledge from?**

Where do we get background knowledge from?

- Hand-crafted rules

Where do we get background knowledge from?

- Hand-crafted rules
- Supervised multitask learning

Where do we get background knowledge from?

- Hand-crafted rules
- Supervised multitask learning
- **Self-supervised learning**



Playgol

1. Play
2. Build

Playing

1. Generate random tasks
2. Learn solutions to them
3. Add solutions to the BK

Building

Solve user-supplied problems using
the augmented BK

Why should it work?

We increase branching but reduce depth

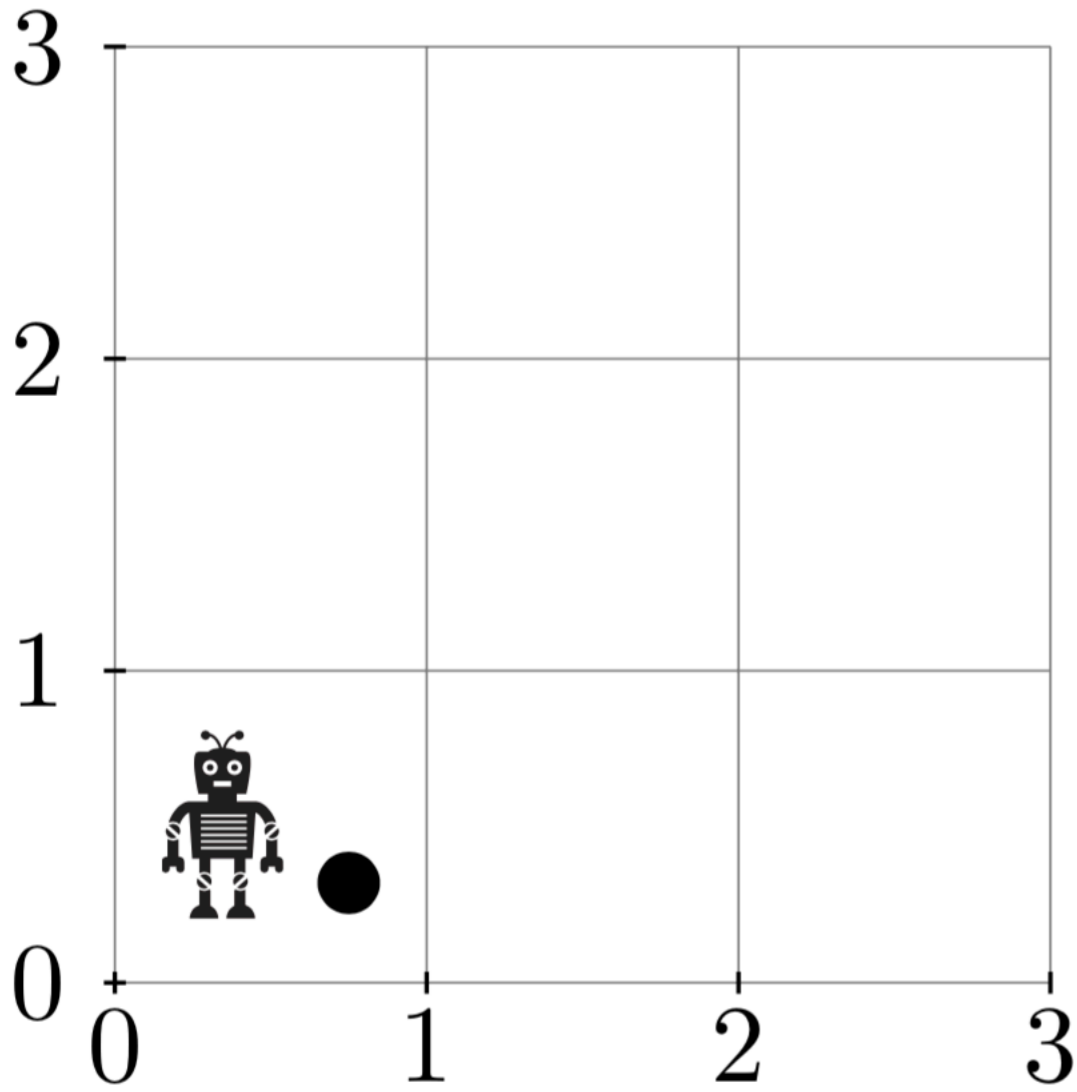
Does it work in practice?

Q1. Can playing improve performance?

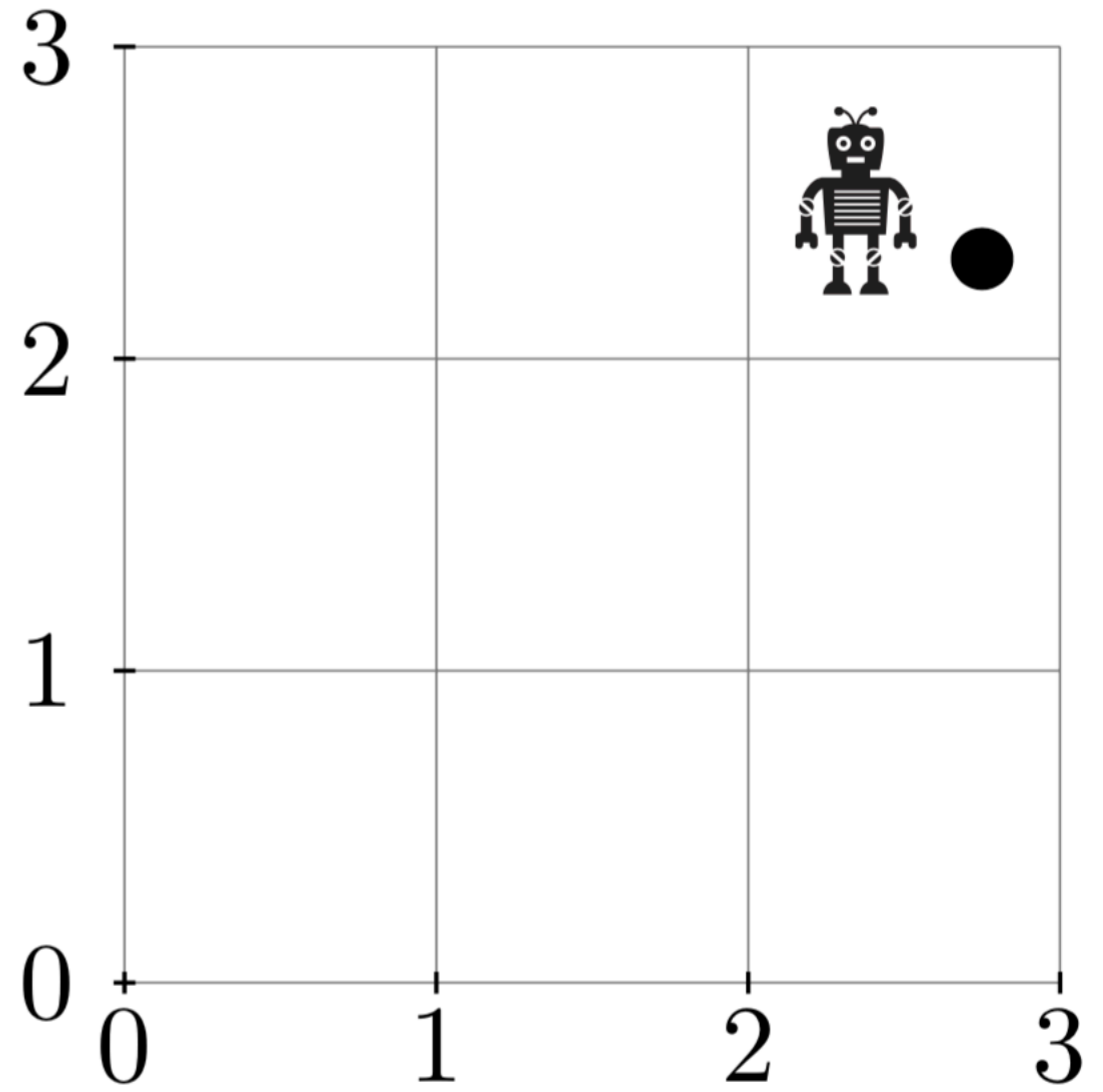
Q2. Can playing improve performance without many play tasks?

Q3. Can predicate invention improve performance?

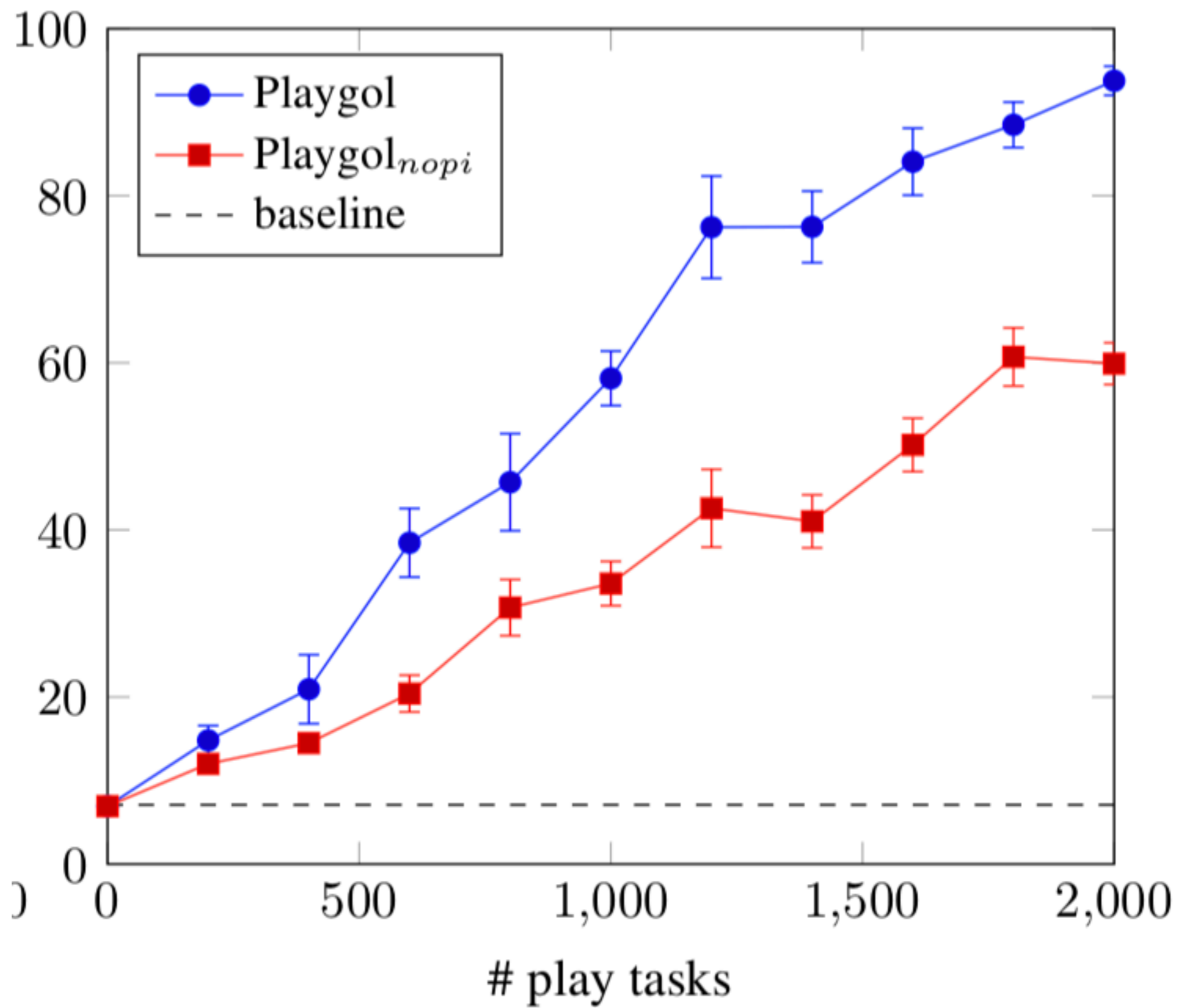
Robot planning

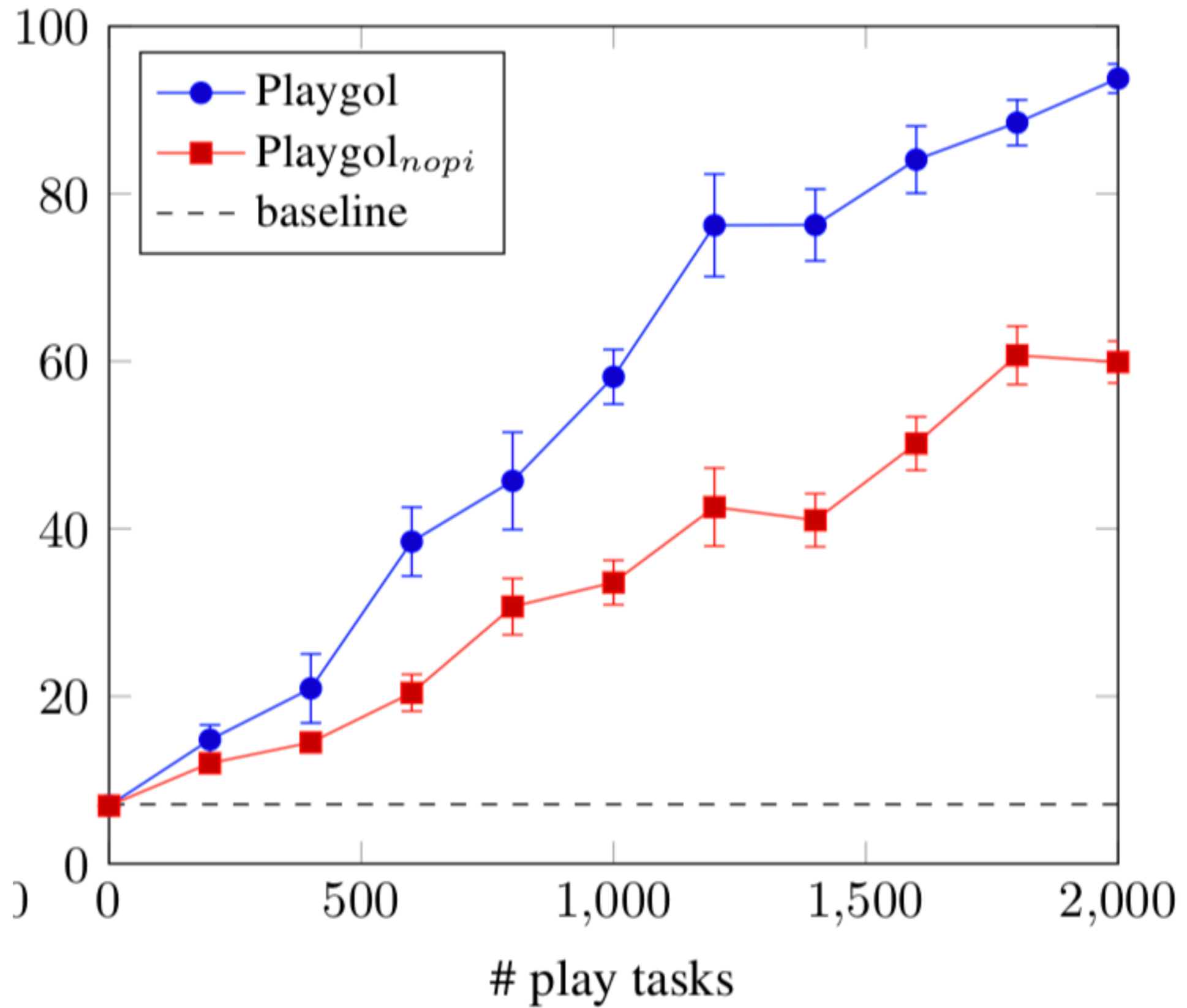


(a) Initial state



(b) Final state





We should need to sample 24,000,000 play tasks

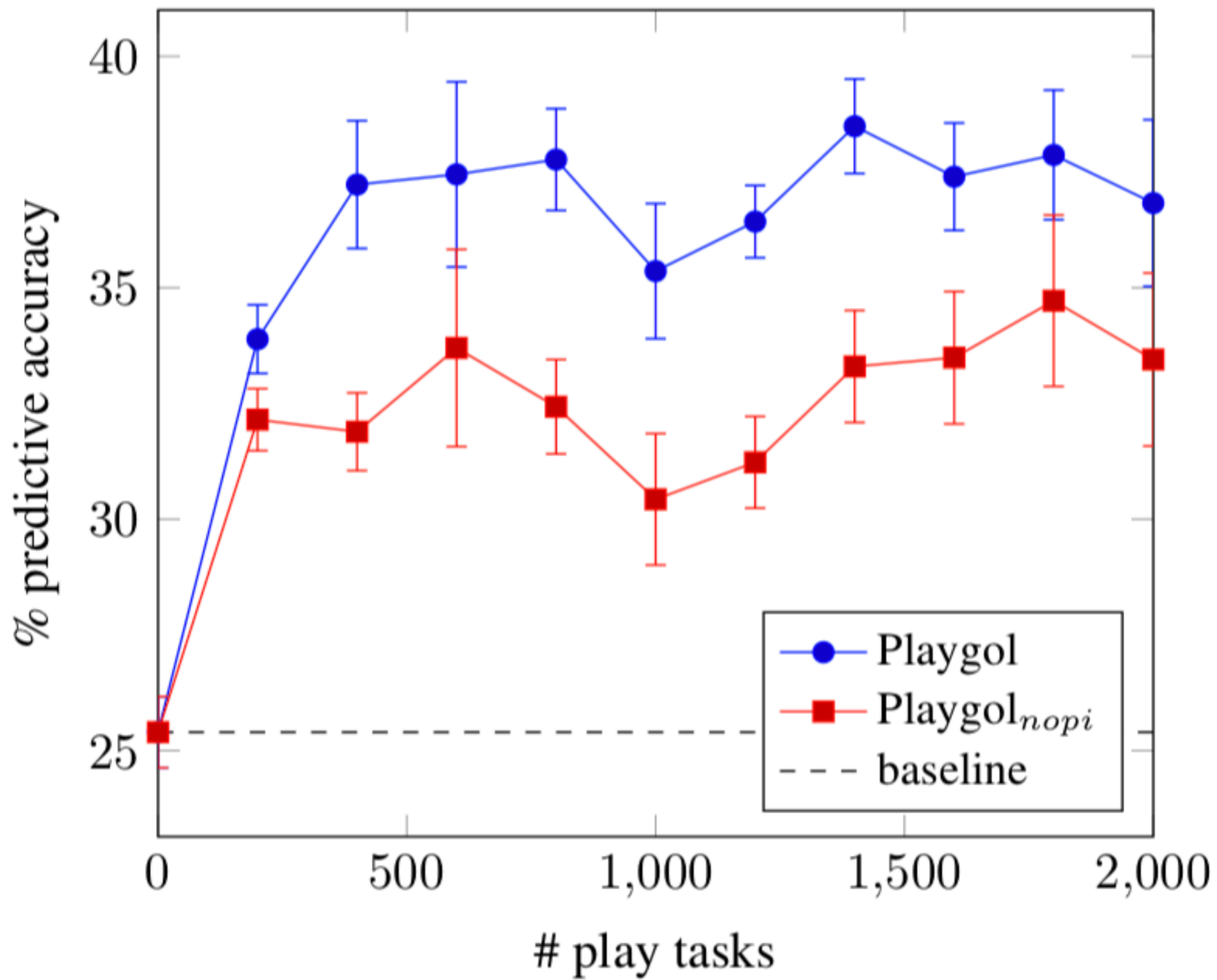
String transformations

Real-world build tasks

Input	Output
22 July, 1983 (35 years old)	JUL
30 October, 1955 (63 years old)	OCT
2 November, 1954 (64 years old)	NOV

Play tasks

Task	Input	Output
play_9	.f\73\R)	F
play_52	@B4\X;3MjKdyZzC	B
play_136	9pfy'ktfbS1v	99PF
play_228	I6zihQk-	Q



Input

Output

22 July, 1983 (35 years old)

JUL

30 October, 1955 (63 years old)

OCT

2 November, 1954 (64 years old)

NOV

```
build_95(A,B):-play_228(A,C),play_136_1(C,B).
play_228(A,B):-play_52(A,B),uppercase(B).
play_228(A,B):-skip1(A,C),play_228(C,B).
play_136_1(A,B):-play_9(A,C),mk_uppercase(C,B).
play_9(A,B):-skip1(A,C),mk_uppercase(C,B).
play_52(A,B):-skip1(A,C),copy1(C,B).
```

Todo

1. Better sampling
2. When does it work?

References

- Andrew Cropper, Stephen Muggleton. Metagol system. <https://github.com/metagol/metagol>
- Andrew Cropper, Stephen Muggleton. Learning Higher-Order Logic Programs through Abstraction and Invention. IJCAI 2016.
- Dianhuan Lin, Eyal Dechter, Kevin Ellis, Joshua B. Tenenbaum, Stephen Muggleton. Bias reformulation for one-shot function induction. ECAI 2014.