

Learning Libraries of Programmatic Policies

Levi H. S. Lelis

Assistant Professor

Canada CIFAR AI Chair, Amii

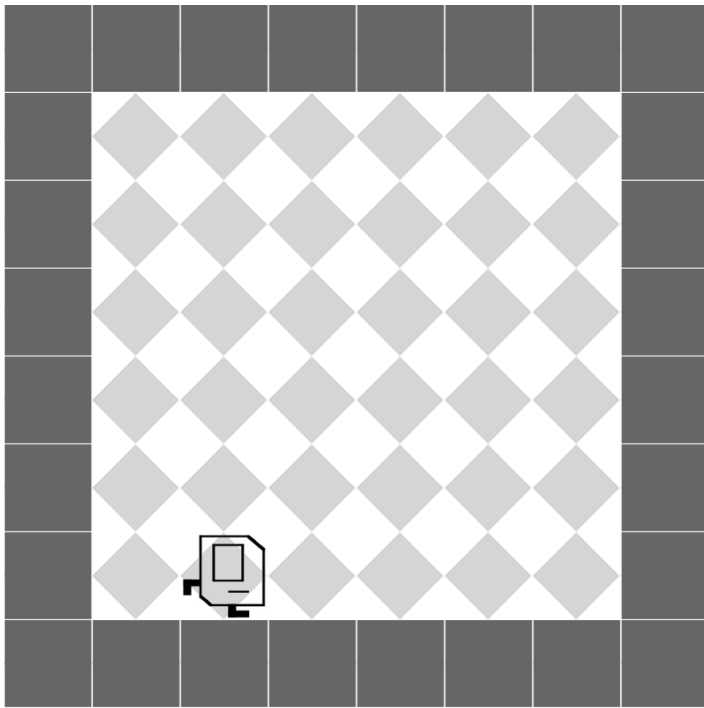
University of Alberta, Canada

Universidade Federal de Viçosa, Brazil

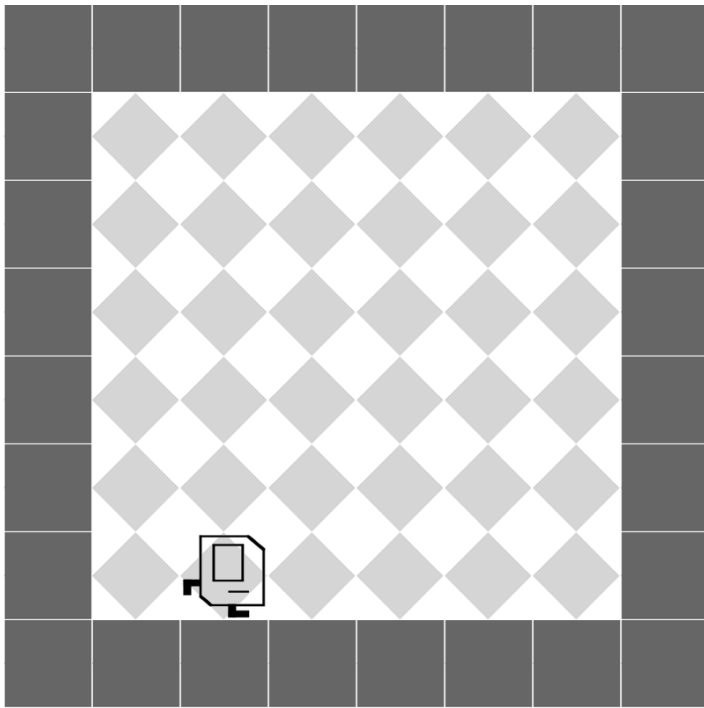
Mahdi Alikhasi, Ken Thjia, Tales Carvalho, Rubens Moraes

Programs are Everywhere

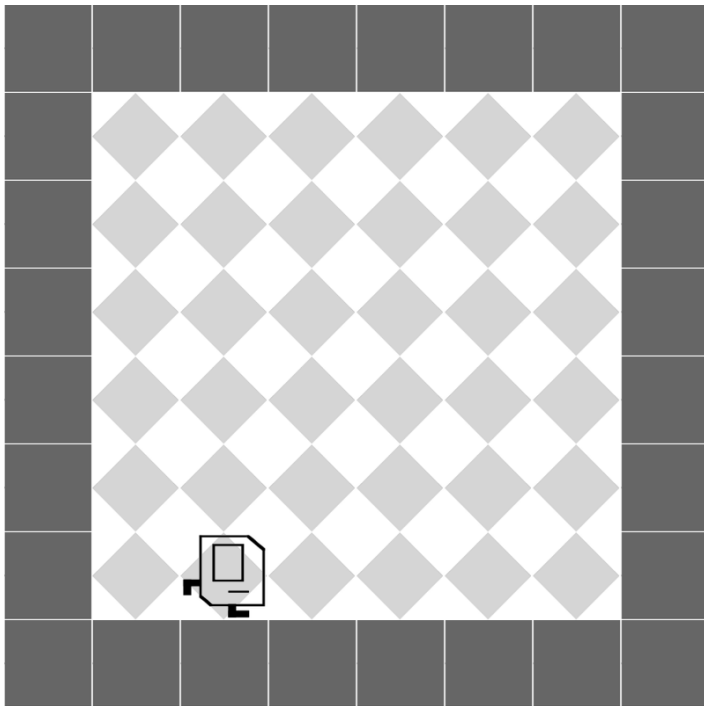
Sequential Decision Making



Sequential Decision Making

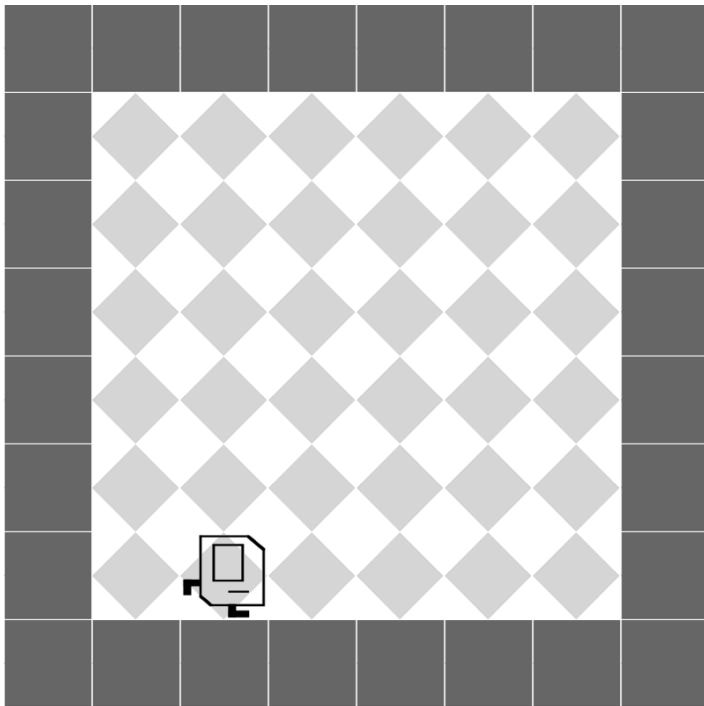


Sequential Decision Making



```
WHILE markerPresent  
  pickMarker  
  IF frontIsClear  
    move  
  ELSE  
    turnLeft
```

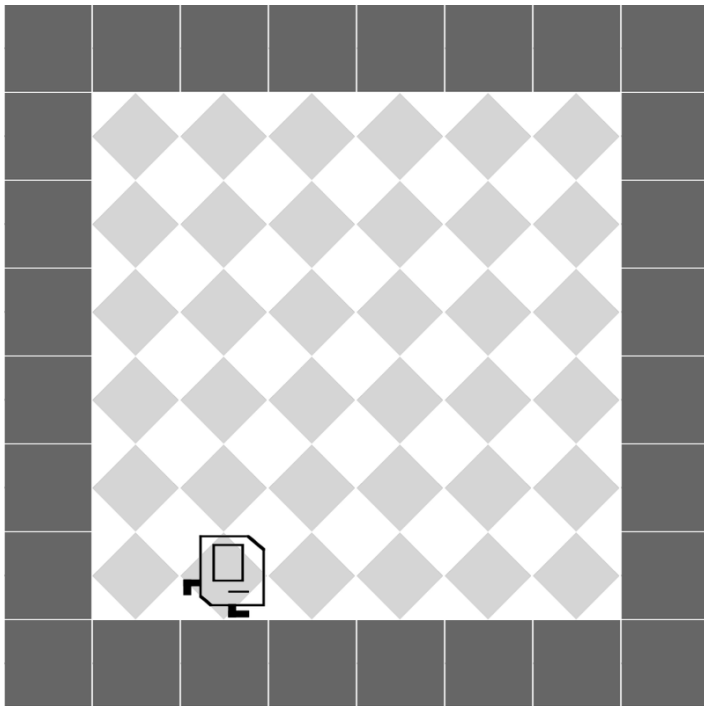
Sequential Decision Making



```
WHILE markerPresent  
  pickMarker  
  IF frontIsClear  
    move  
  ELSE  
    turnLeft
```

Programmatic policy

Sequential Decision Making



Reinforcement learning problem

```
WHILE markerPresent  
  pickMarker  
  IF frontIsClear  
    move  
  ELSE  
    turnLeft
```

Programmatic policy

Search for Programmatic Policies

Search for Programmatic Policies

- ❖ Search space.

Search for Programmatic Policies

- ❖ Search space.

Search for Programmatic Policies

- ❖ Search space.

Search for Programmatic Policies

- ❖ Search space.
- ❖ Search procedure.

Programming Language

Programming Language

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Programming Language

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Quiz: Can the program `if x <= y then 0 else 1` be written in this language?

Programming Language

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Quiz: Can the program `if x <= y then 0 else 1` be written in this language?

Answer: Yes.

Programming Language

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Quiz: Can the program `if x <= y then 0 else 1` be written in this language?

Answer: Yes.

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

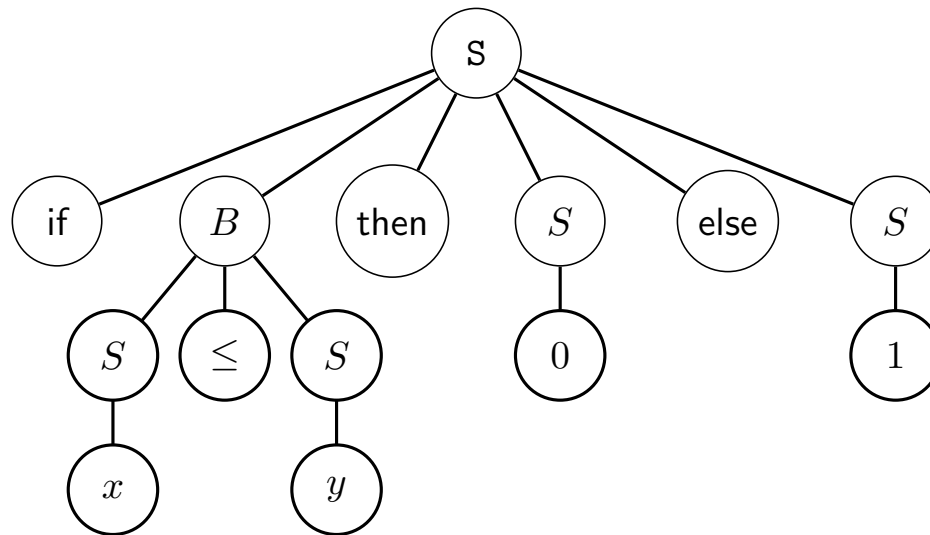
$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

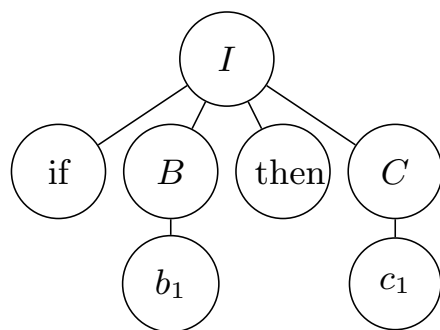
$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$

Representing Programs in Memory

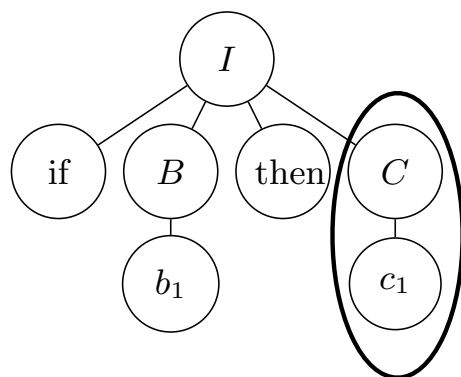
```
if x <= y then 0 else 1
```



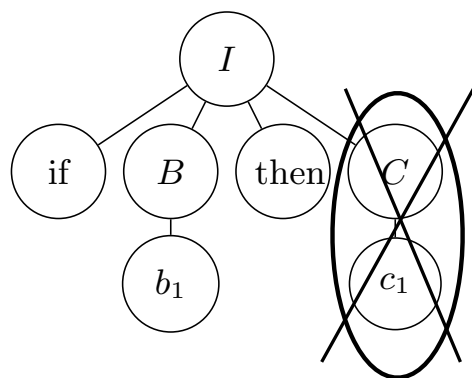
Stochastic Neighborhood Function



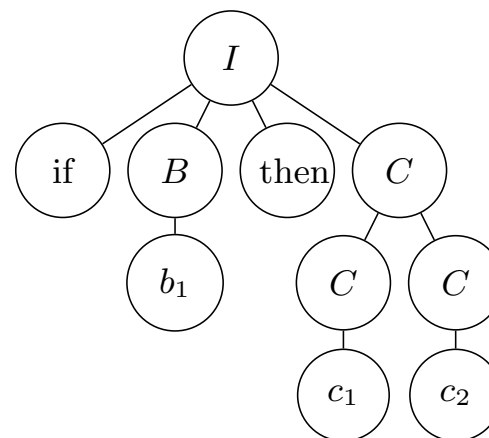
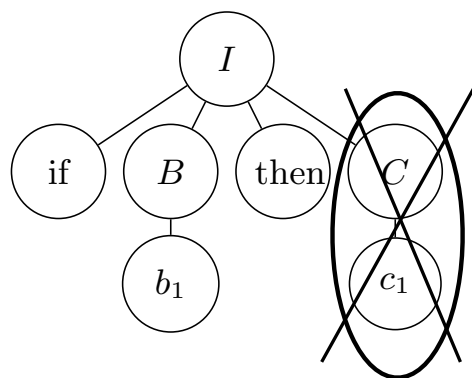
Stochastic Neighborhood Function



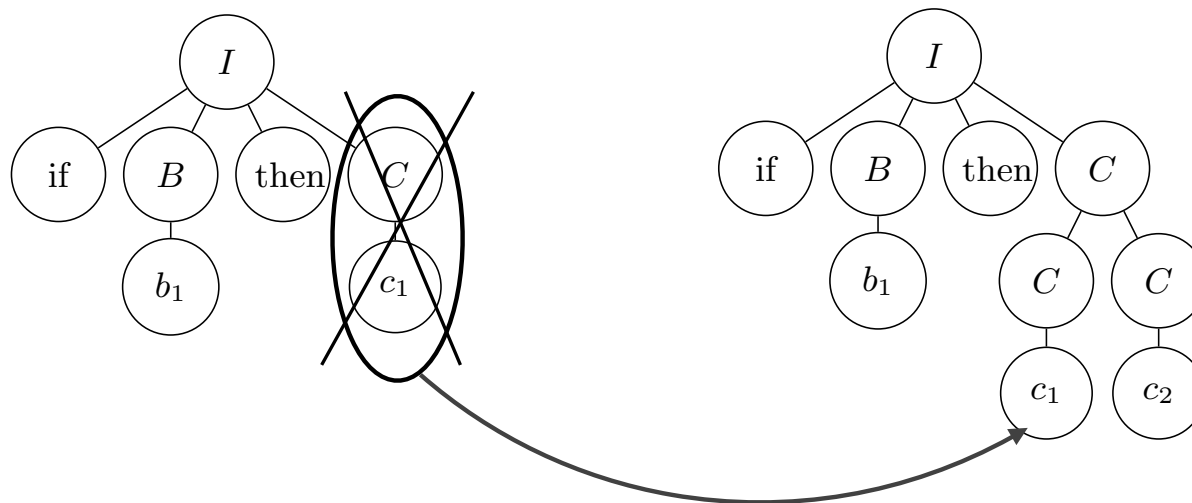
Stochastic Neighborhood Function



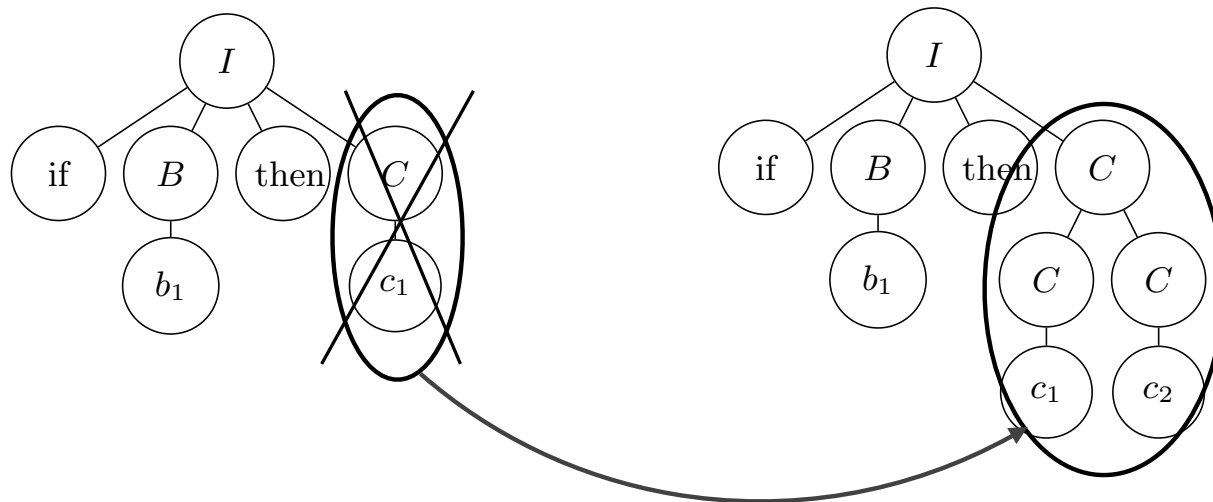
Stochastic Neighborhood Function



Stochastic Neighborhood Function



Stochastic Neighborhood Function



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$

How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1 \bullet$

How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

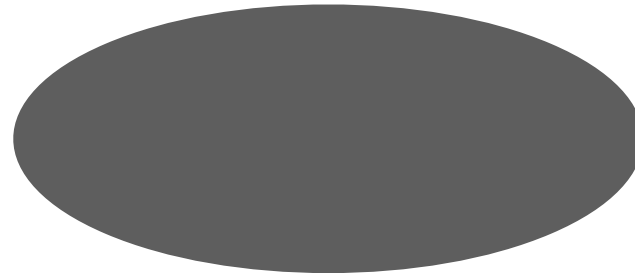
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1 \bullet$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

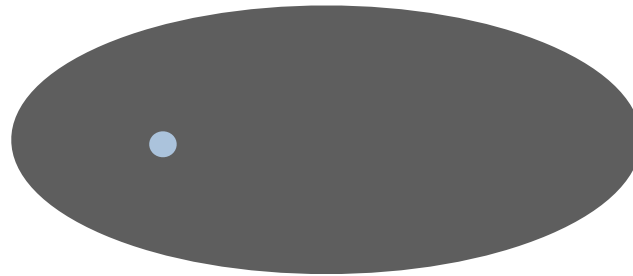
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

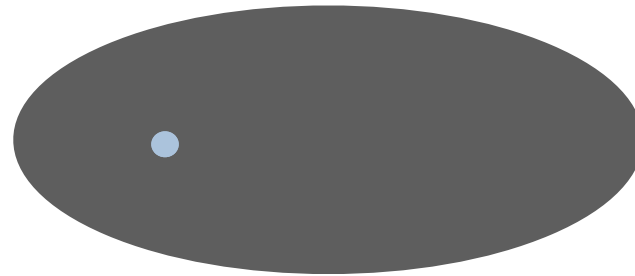
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

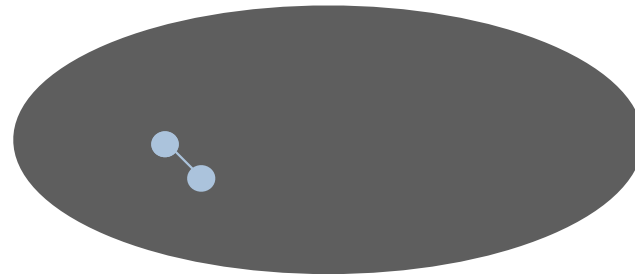
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

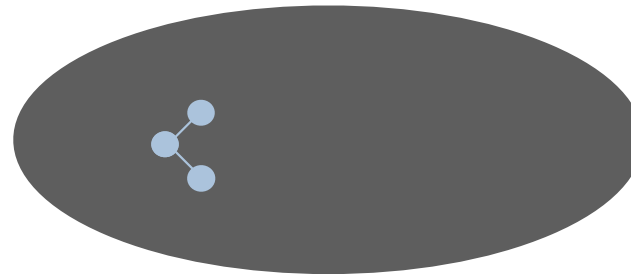
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

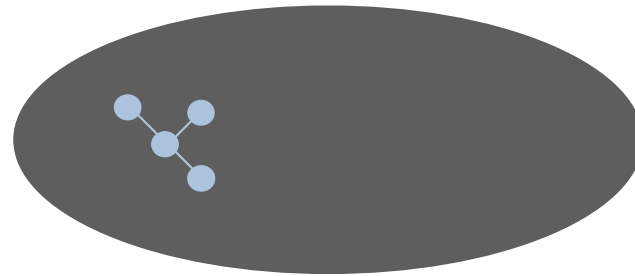
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

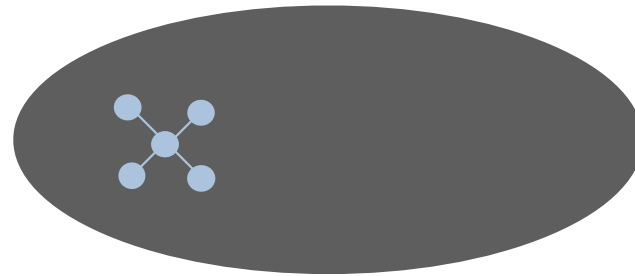
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

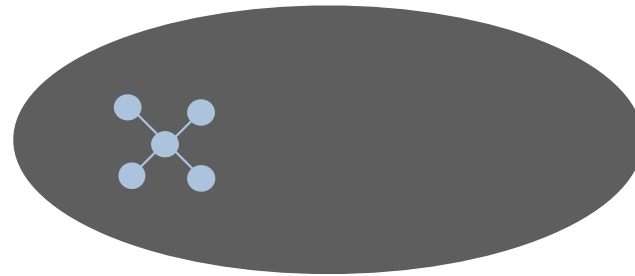
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

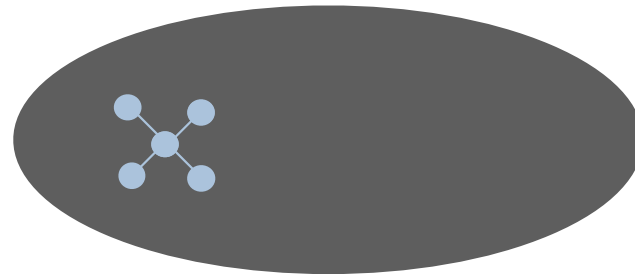
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

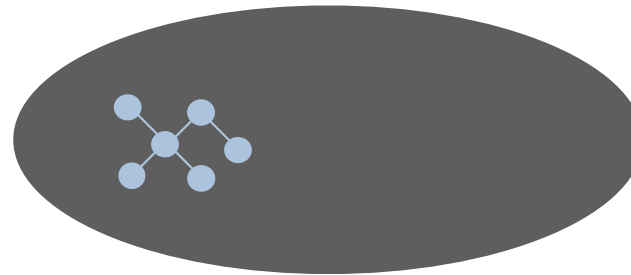
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

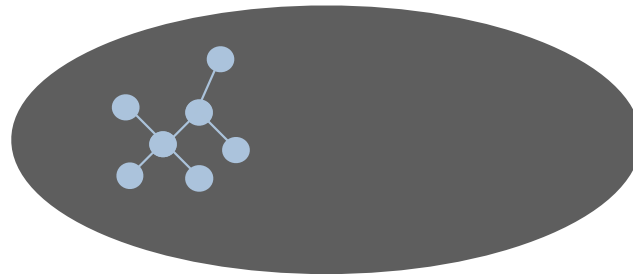
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

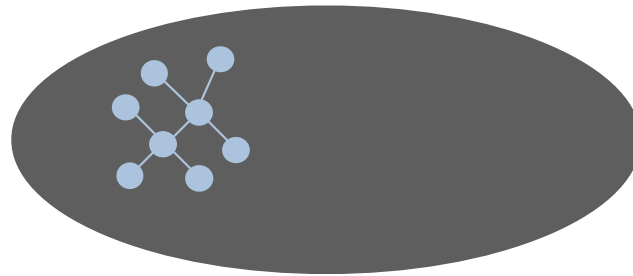
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

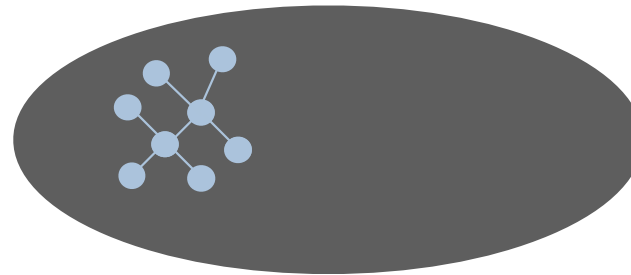
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

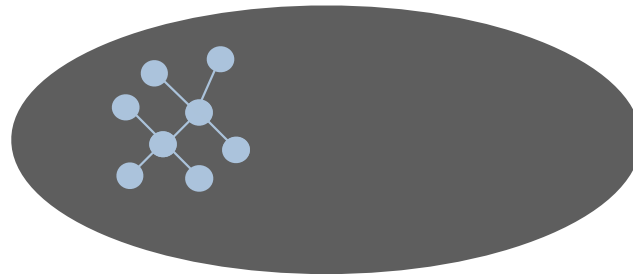
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

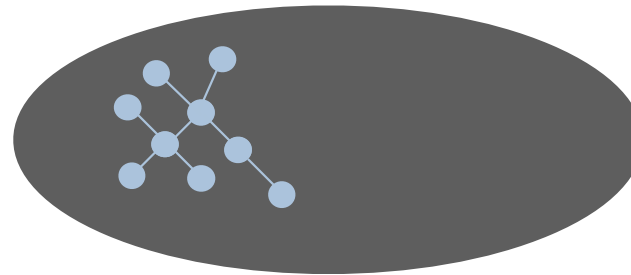
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

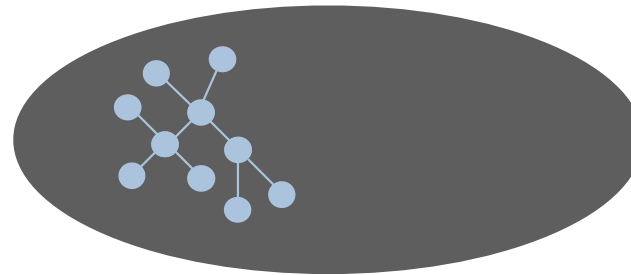
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

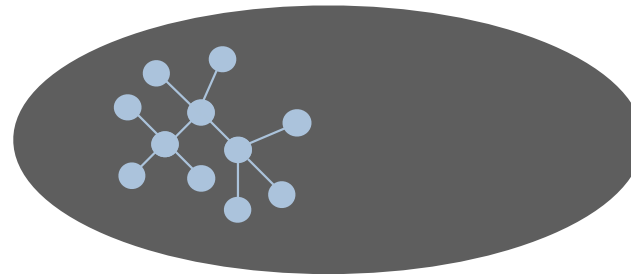
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

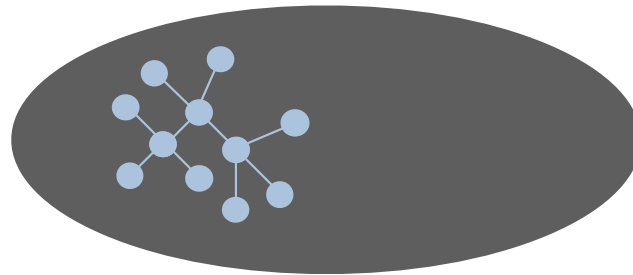
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

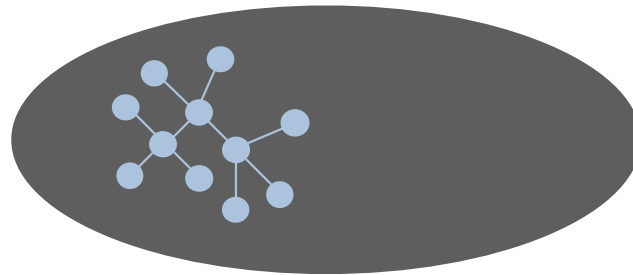
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

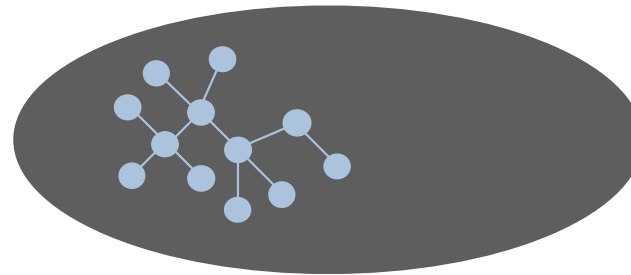
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

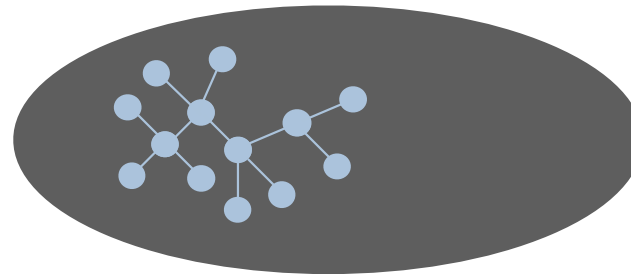
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

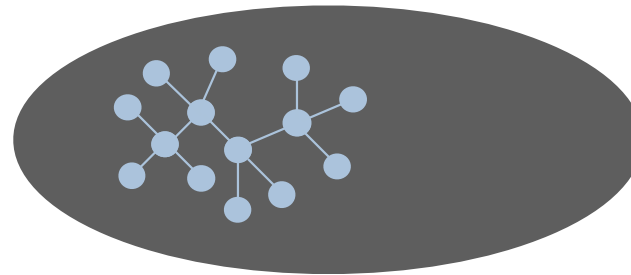
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

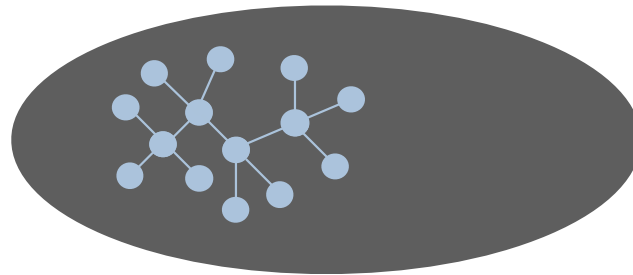
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

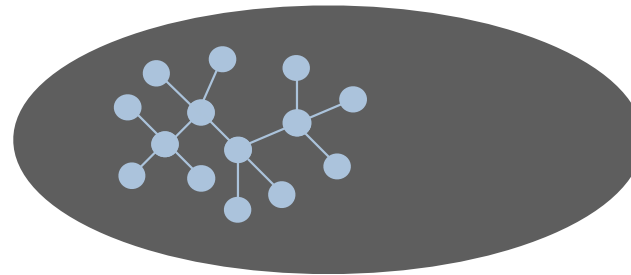
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



How to Search?

Hill-Climbing

$S \rightarrow x \mid y \mid 0 \mid 1 \mid (S + S) \mid \text{if } B \text{ then } S \text{ else } S$

$B \rightarrow (S \leq S) \mid (S == S) \mid (S \geq S)$

Generate "random" program

$S \rightarrow \text{if } B \text{ then } S \text{ else } S$

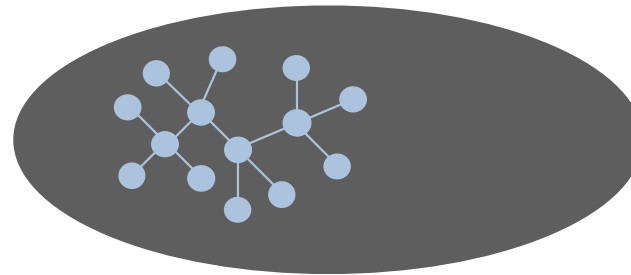
$\rightarrow \text{if } S \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq S \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } S \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } S$

$\rightarrow \text{if } x \leq y \text{ then } 0 \text{ else } 1$



Restarts, restarts, restarts...

Searching for Programs

- ❖ Search space.
- ❖ Search procedure.

Searching for Programs

- ❖ Search space.
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

- ❖ Search space.
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

- ❖ Search space.
 - ❖ Search algorithm must be simple: stochastic hill climbing.

- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

- ❖ Search space.
 - ❖ Search algorithm must be simple: stochastic hill climbing.
 - ❖ Learning semantic spaces (Morales & Lelis, IJCAI 2024)
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

- ❖ Search space.
 - ❖ Search algorithm must be simple: stochastic hill climbing.
 - ❖ Learning semantic spaces (Moraes & Lelis, IJCAI 2024)
 - ❖ Learning programs from neural networks (Alikhasi & Lelis, ICLR 2024).
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

- ❖ Search space.
 - ❖ Search algorithm must be simple: stochastic hill climbing.
 - ❖ Learning semantic spaces (Moraes & Lelis, IJCAI 2024)
 - ❖ Learning programs from neural networks (Alikhasi & Lelis, ICLR 2024).
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Searching for Programs

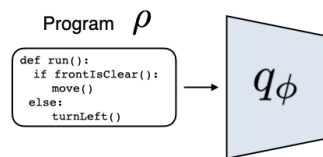
- ❖ Search space.
 - ❖ Search algorithm must be simple: stochastic hill climbing.
 - ❖ Learning semantic spaces (Moraes & Lelis, IJCAI 2024)
 - ❖ Learning programs from neural networks (Alikhasi & Lelis, ICLR 2024).
- ❖ Search procedure.
 - ❖ Bee Search (Ameen & Lelis, JAIR 2023).

Spaces Conducive to Search - LEAPS

Program ρ

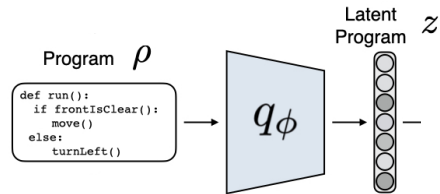
```
def run():  
  if frontIsClear():  
    move()  
  else:  
    turnLeft()
```

Spaces Conducive to Search - LEAPS



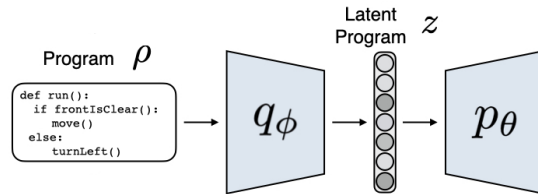
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



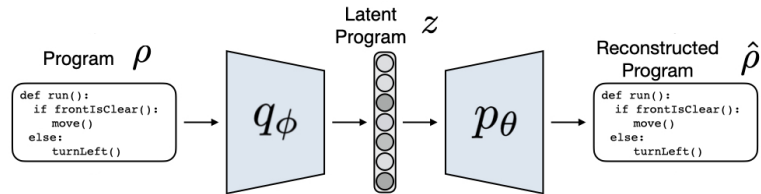
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



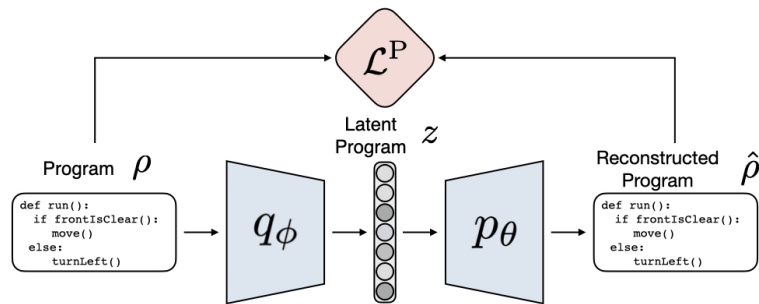
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



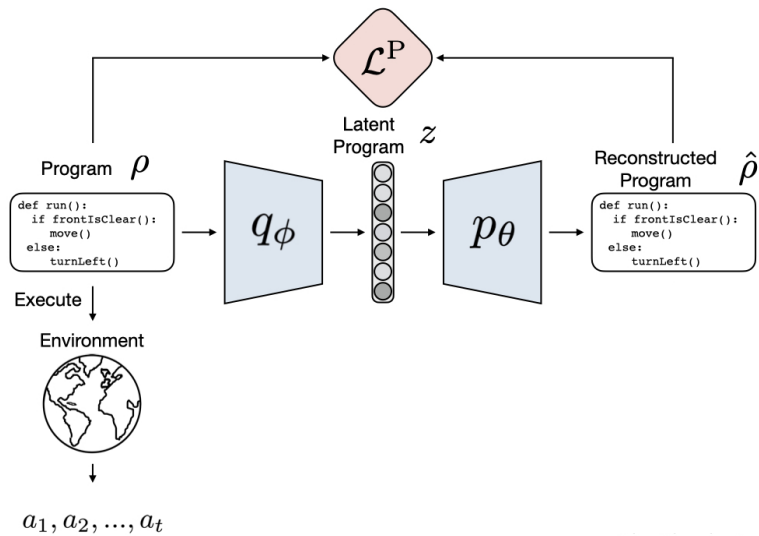
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



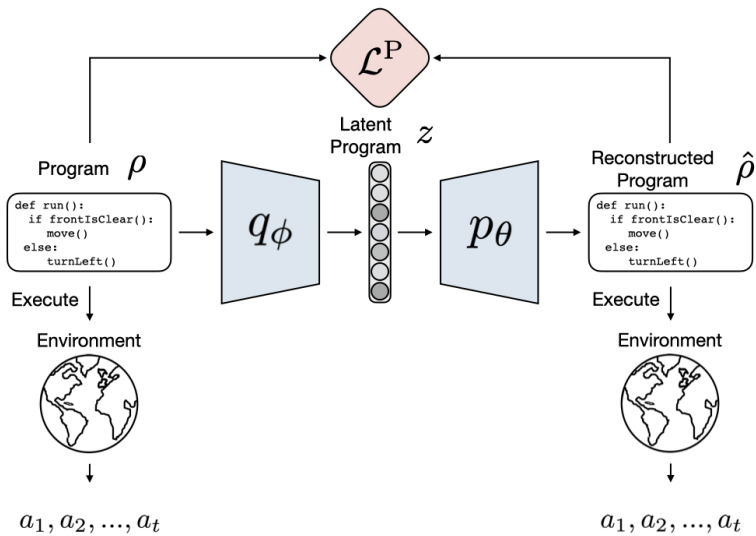
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



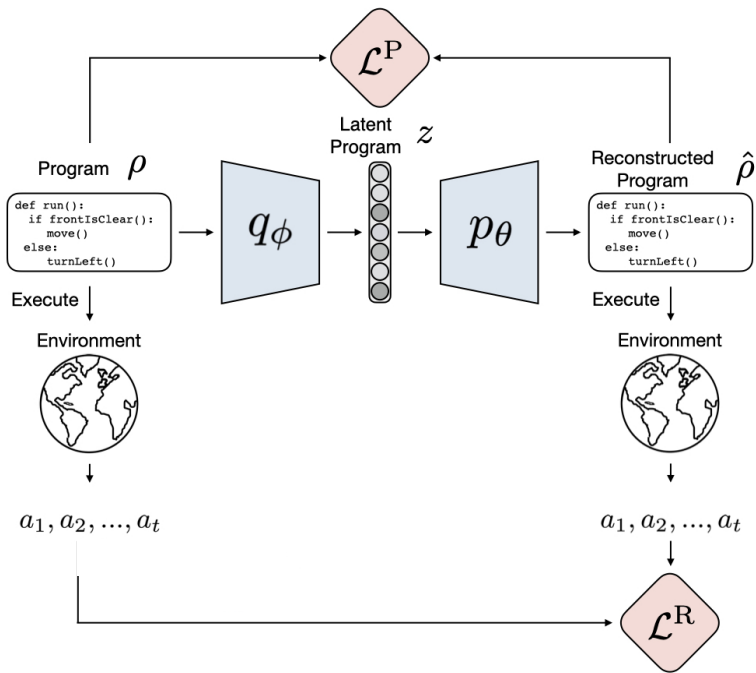
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



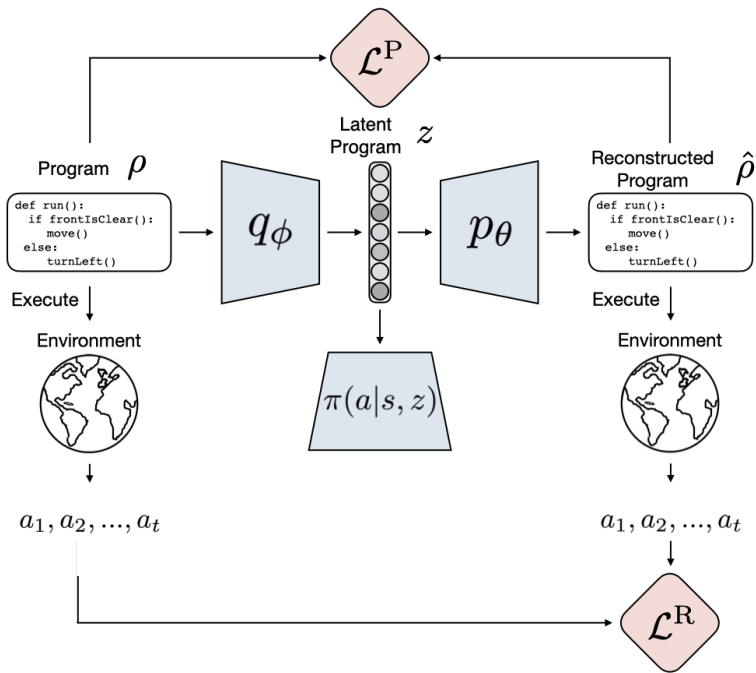
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



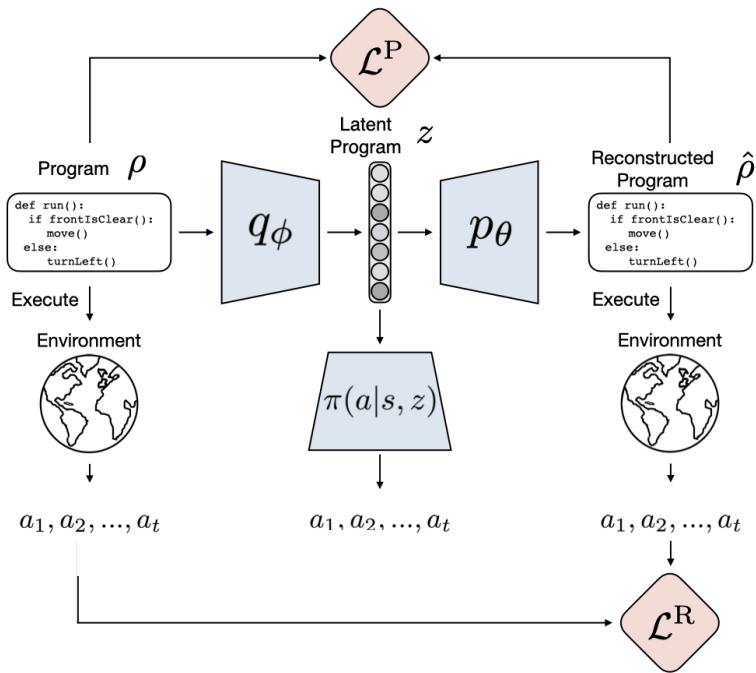
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



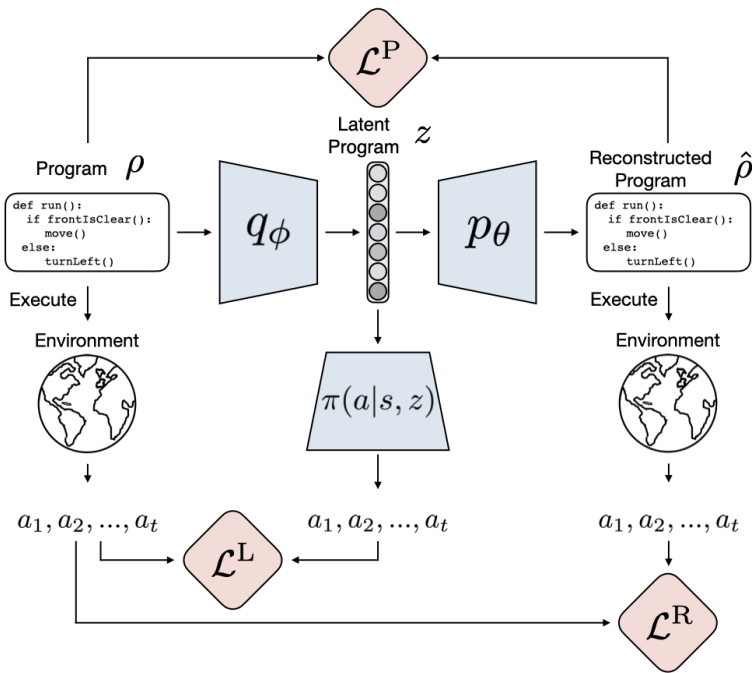
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



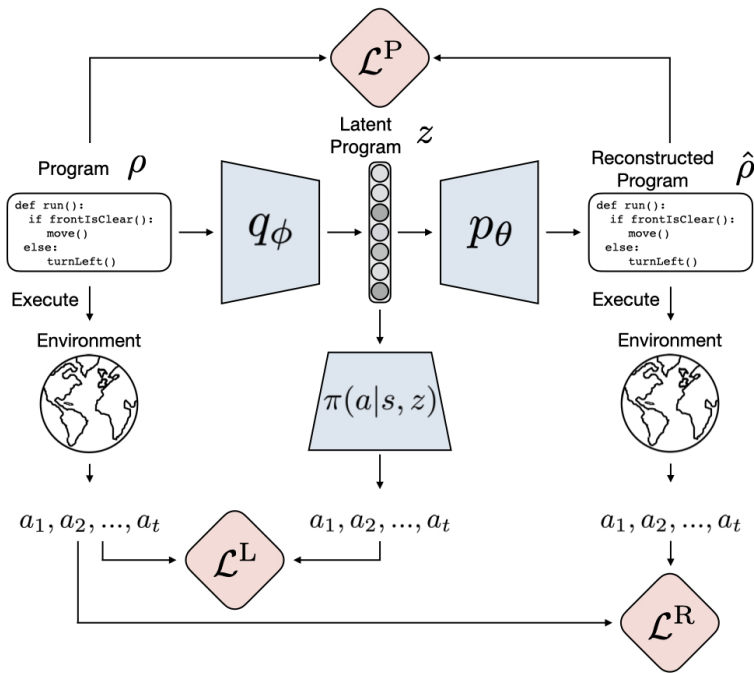
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



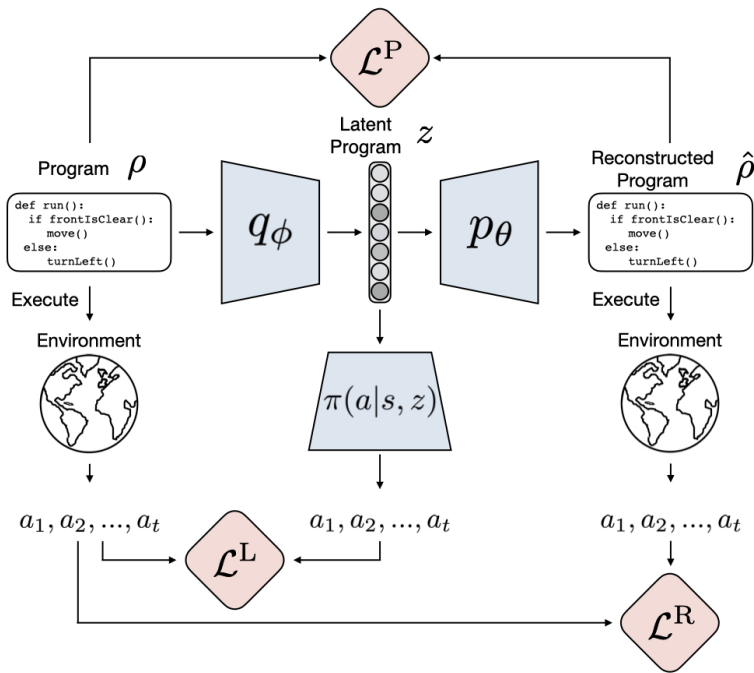
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS

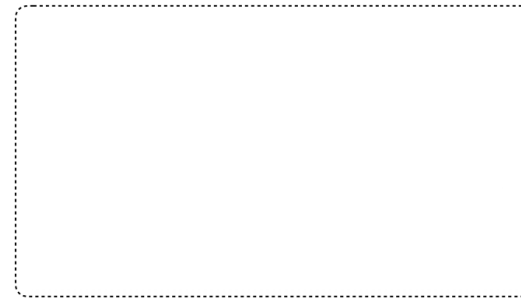


D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS

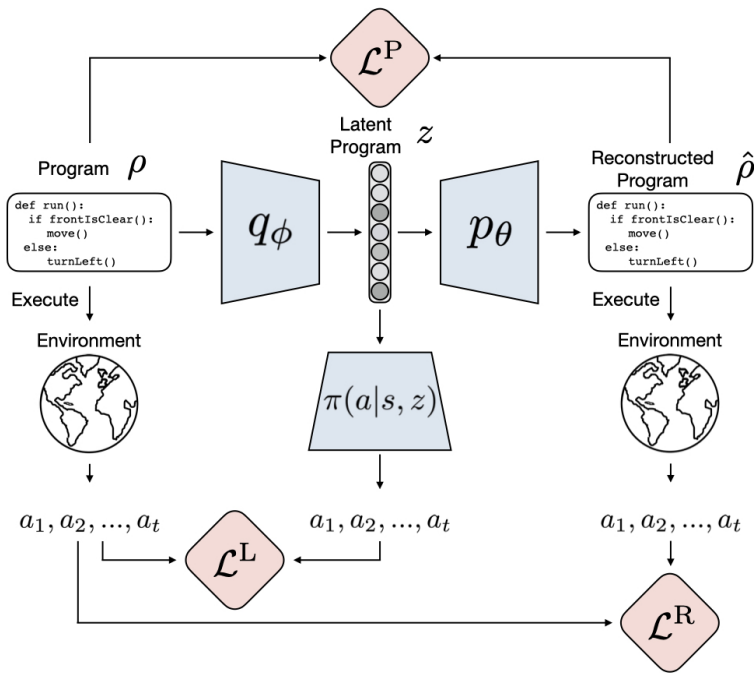


Cross Entropy Method



D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS

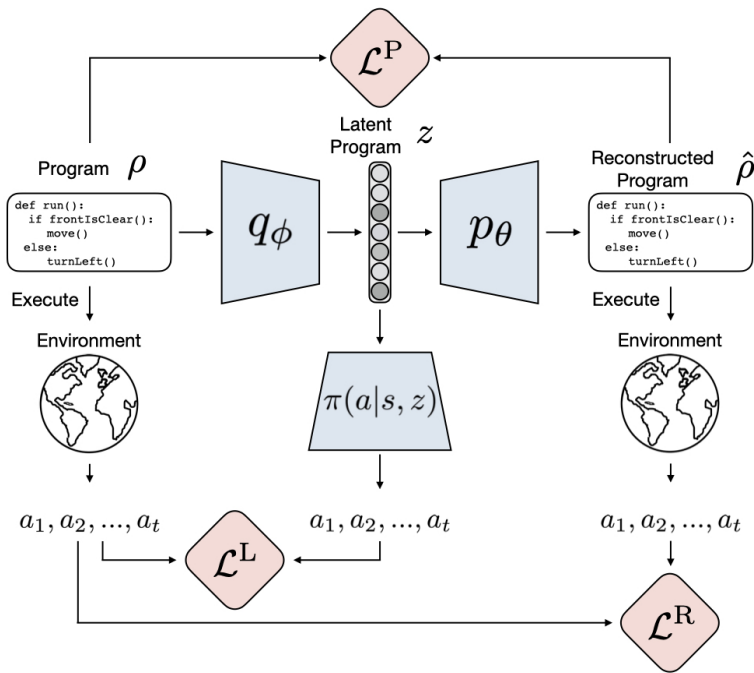


Cross Entropy Method

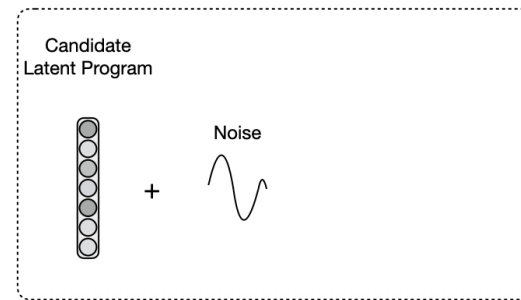


D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS

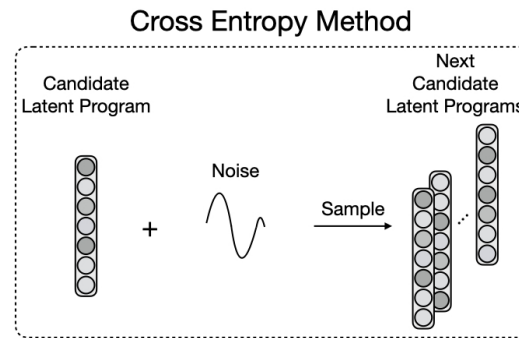
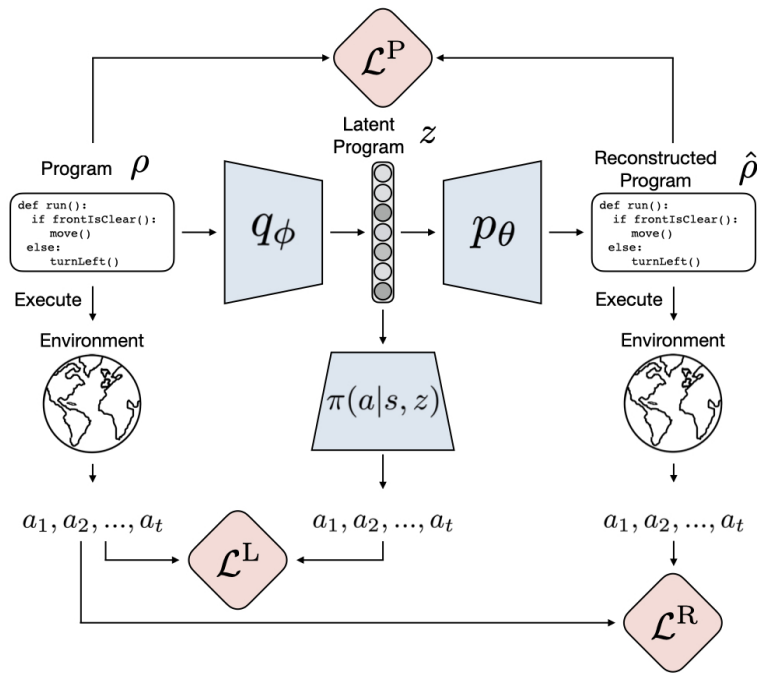


Cross Entropy Method



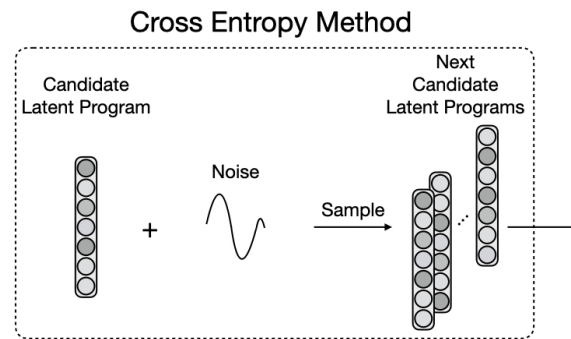
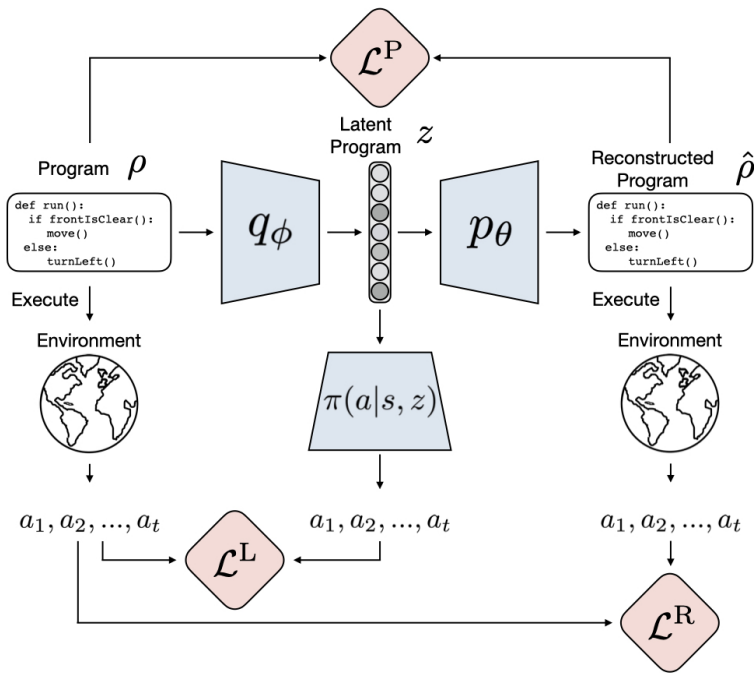
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



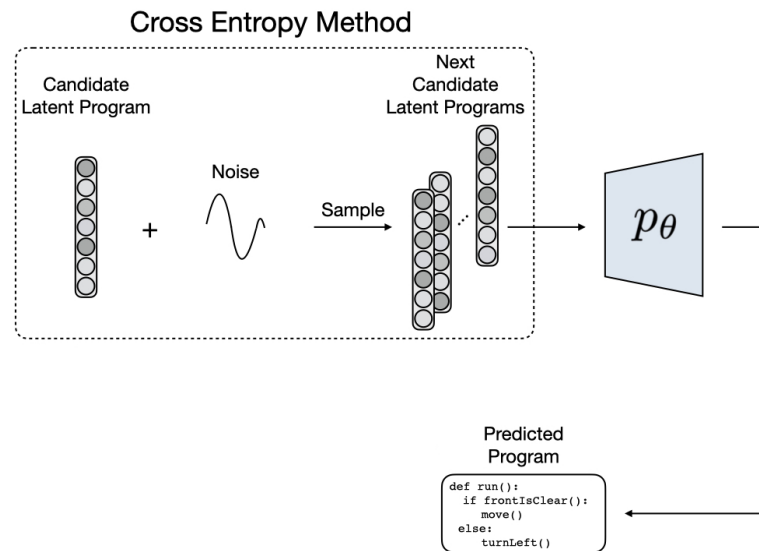
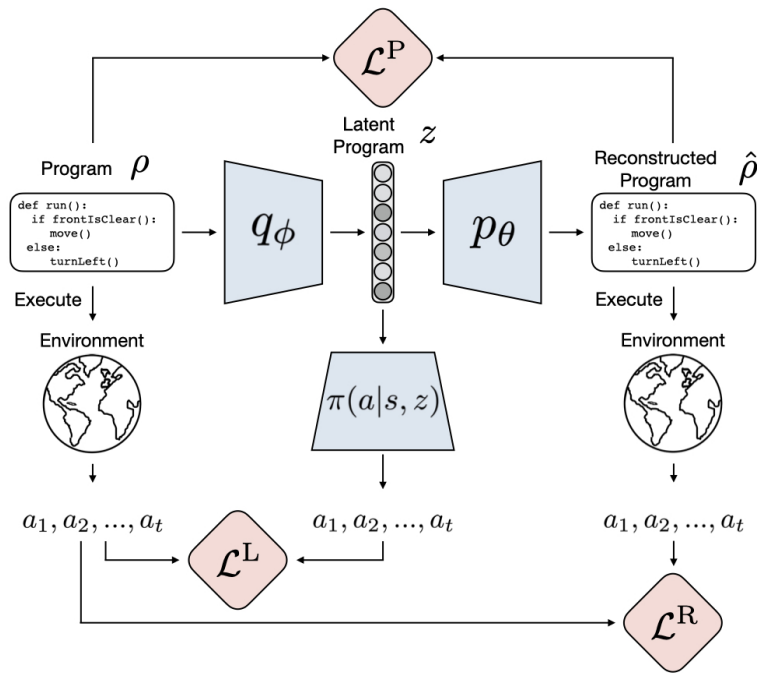
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



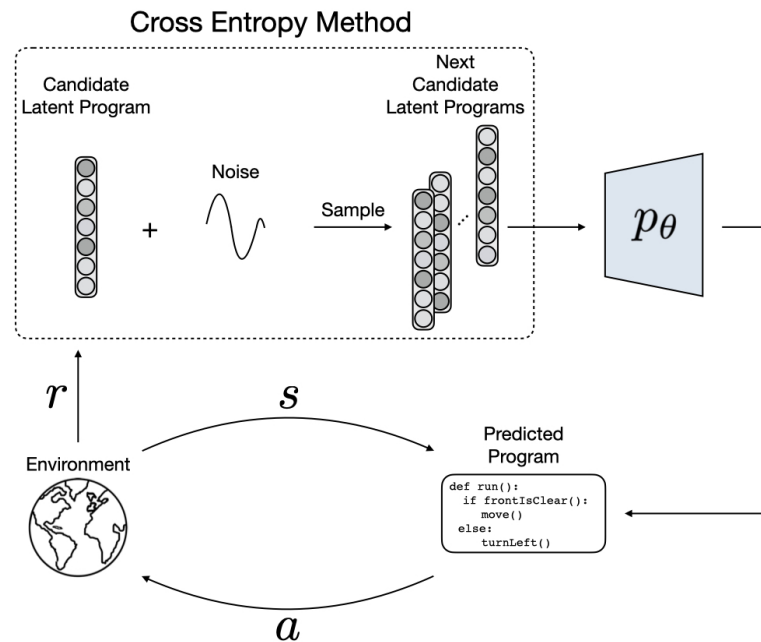
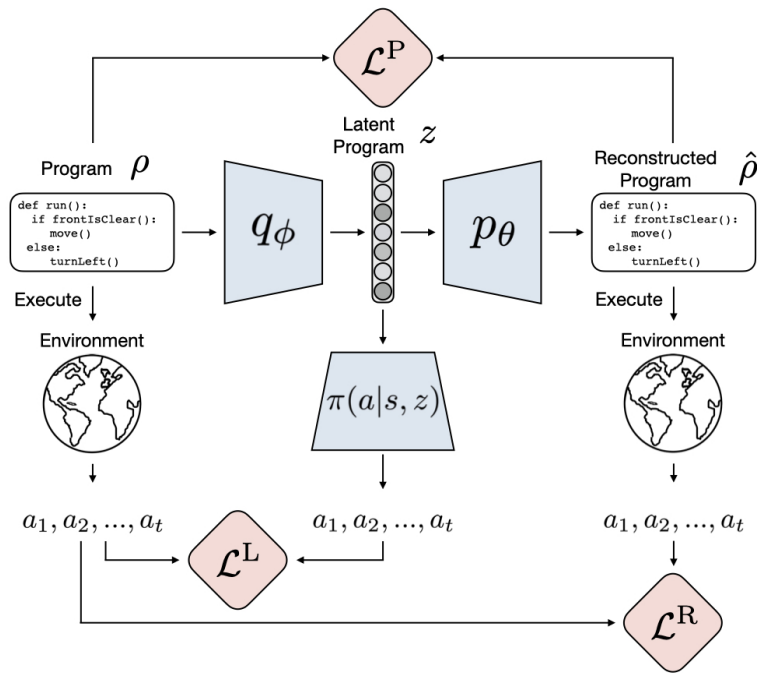
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



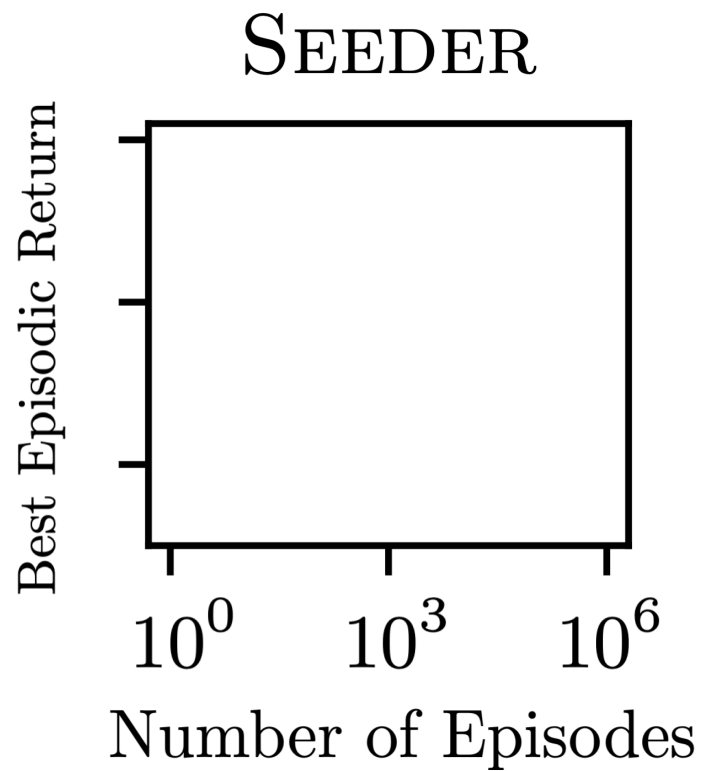
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Spaces Conducive to Search - LEAPS



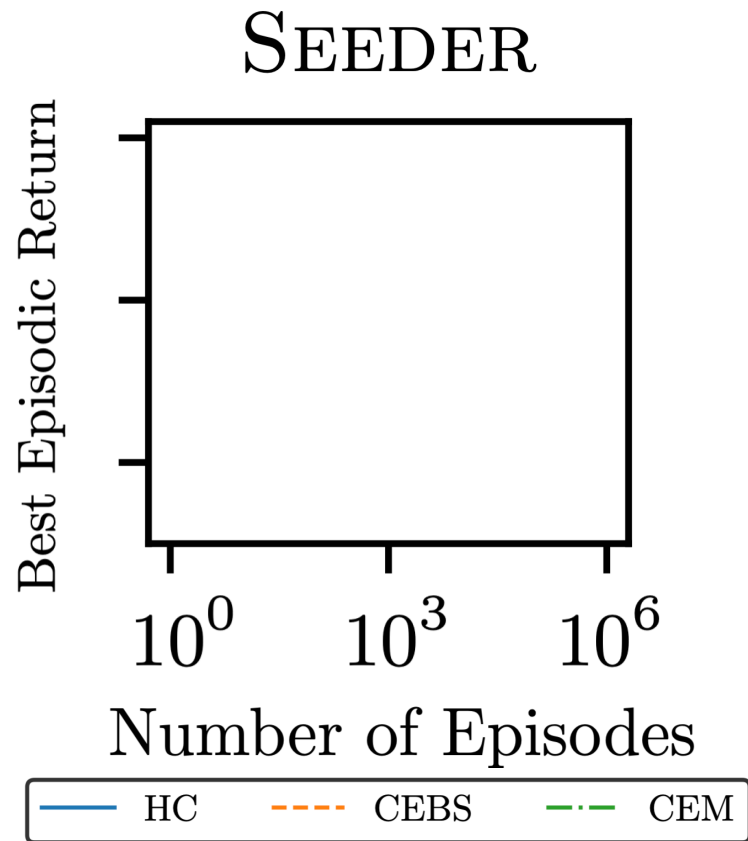
D. Trivedi, J. Zhang, S. Sun, J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies.

Empirical Results



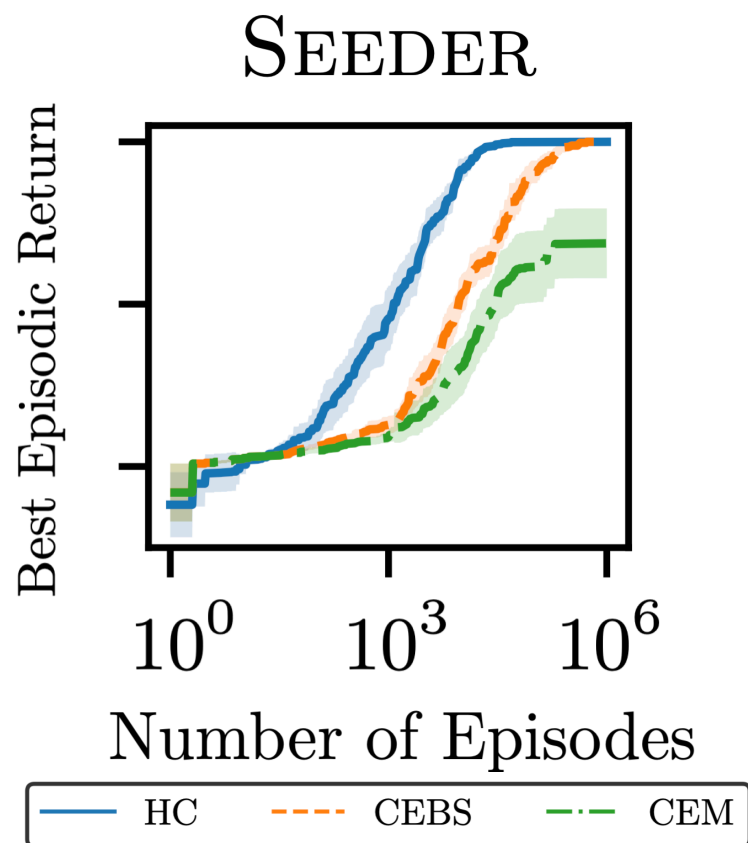
(Carvalho, Thjia & Lelis, ICLR 2024)

Empirical Results



(Carvalho, Thjia & Lelis, ICLR 2024)

Empirical Results



(Carvalho, Thjia & Lelis, ICLR 2024)

Semantic Spaces

Semantic Spaces

Key idea: Neighbor programs should be semantically different.

Semantic Spaces

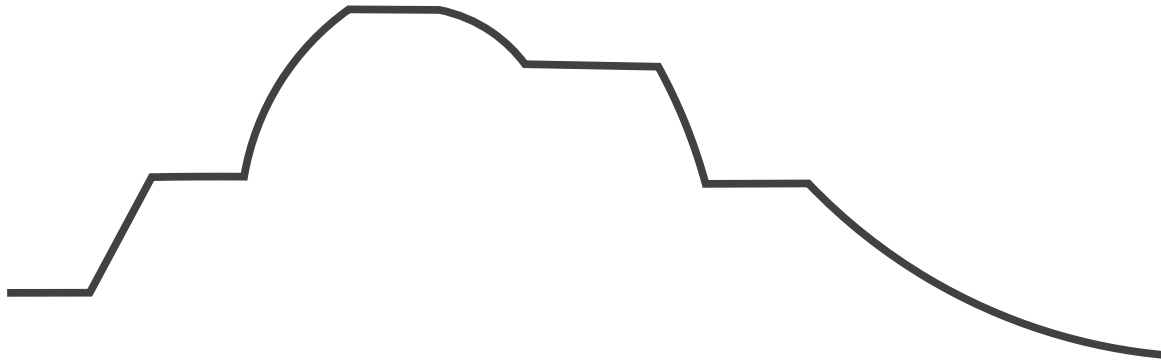
Key idea: Neighbor programs should be semantically different.

Syntax Space

Semantic Spaces

Key idea: Neighbor programs should be semantically different.

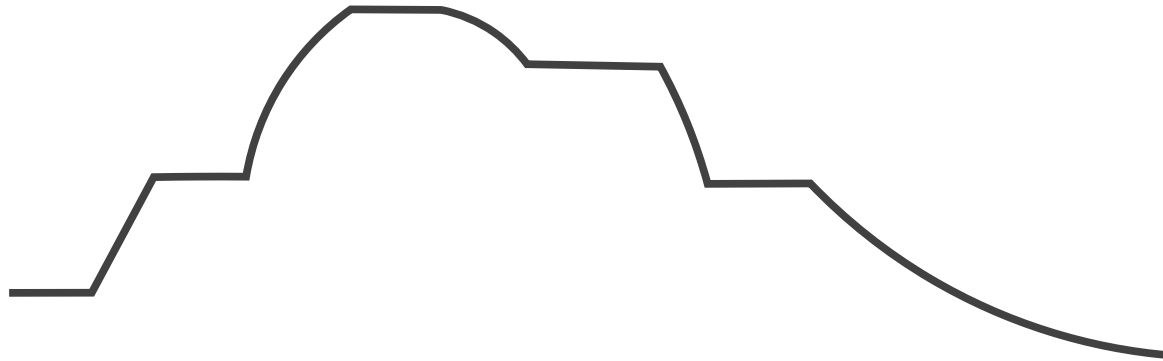
Syntax Space



Semantic Spaces

Key idea: Neighbor programs should be semantically different.

Syntax Space

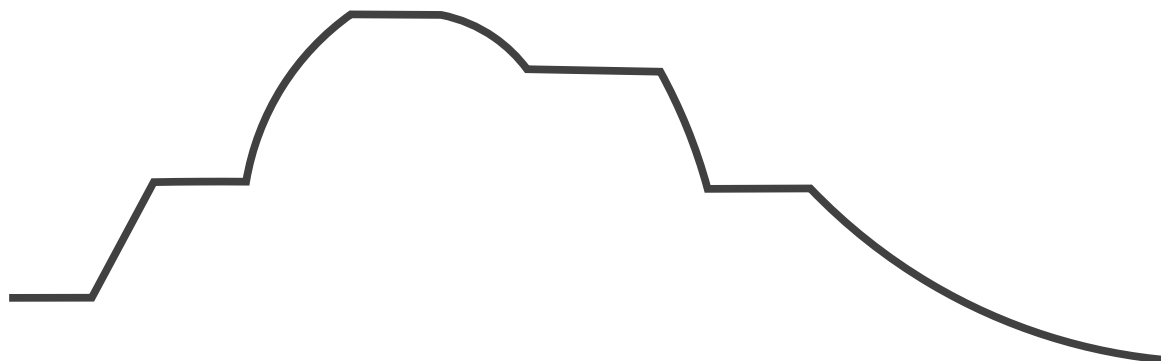


Semantically Identical Neighbors

Semantic Spaces

Key idea: Neighbor programs should be semantically different.

Syntax Space

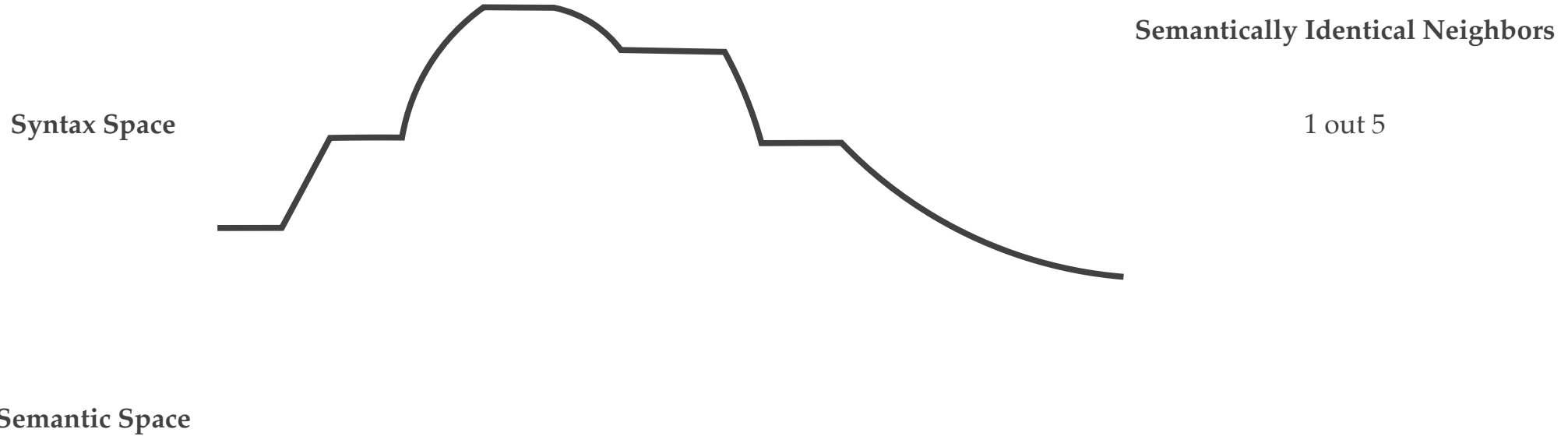


1 out 5

Semantically Identical Neighbors

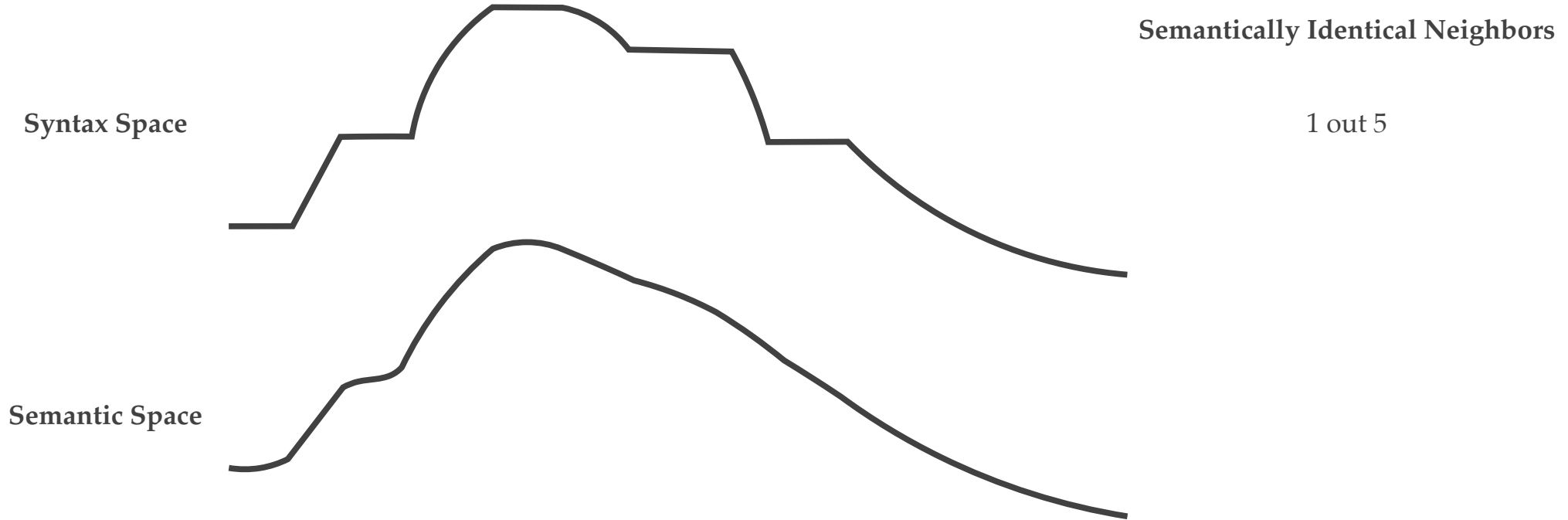
Semantic Spaces

Key idea: Neighbor programs should be semantically different.



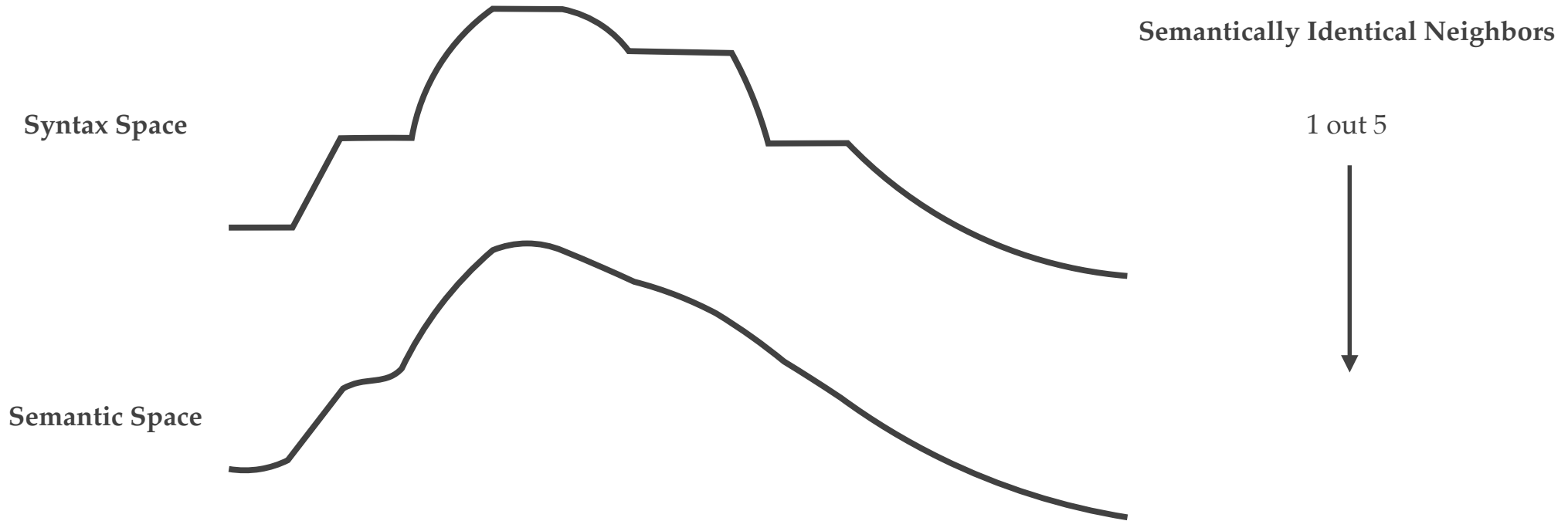
Semantic Spaces

Key idea: Neighbor programs should be semantically different.



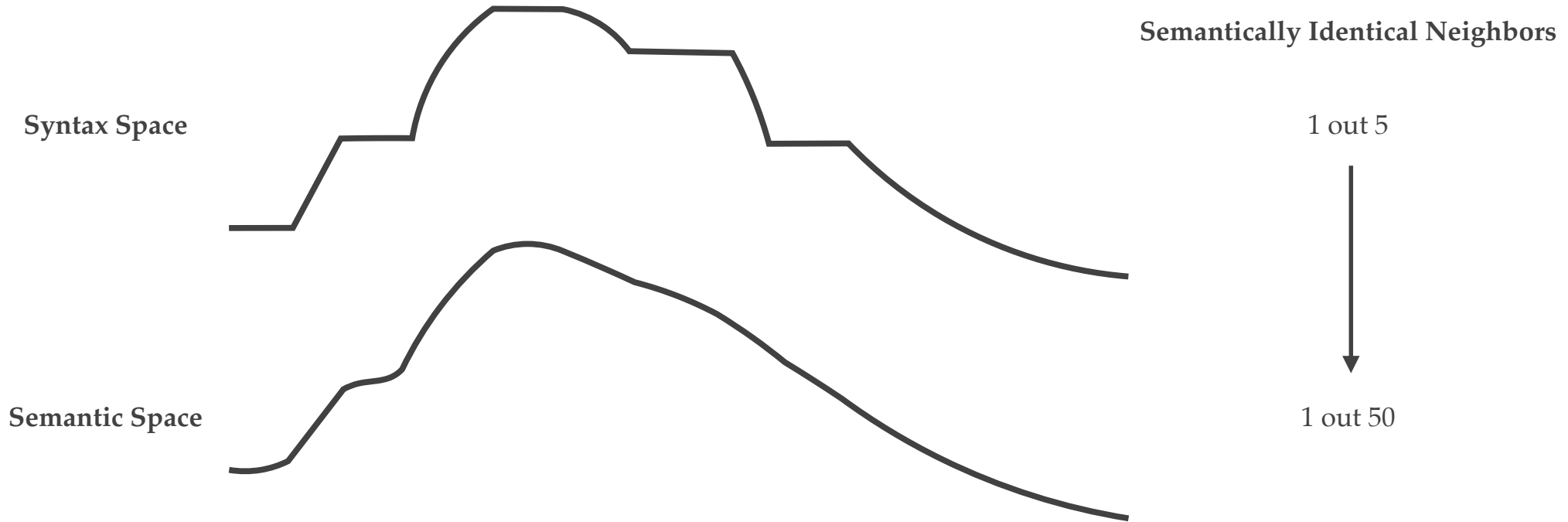
Semantic Spaces

Key idea: Neighbor programs should be semantically different.



Semantic Spaces

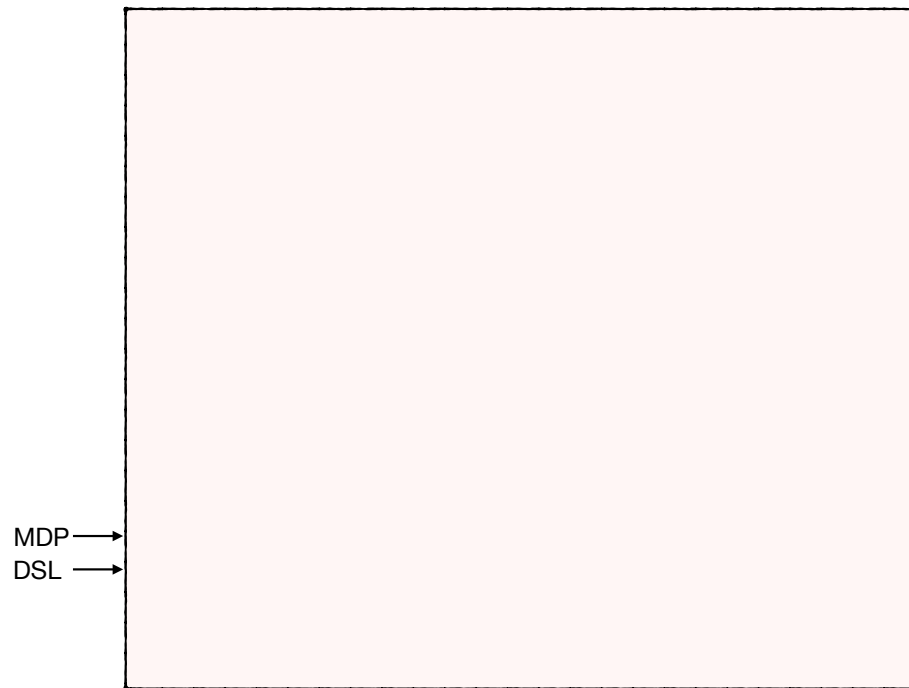
Key idea: Neighbor programs should be semantically different.



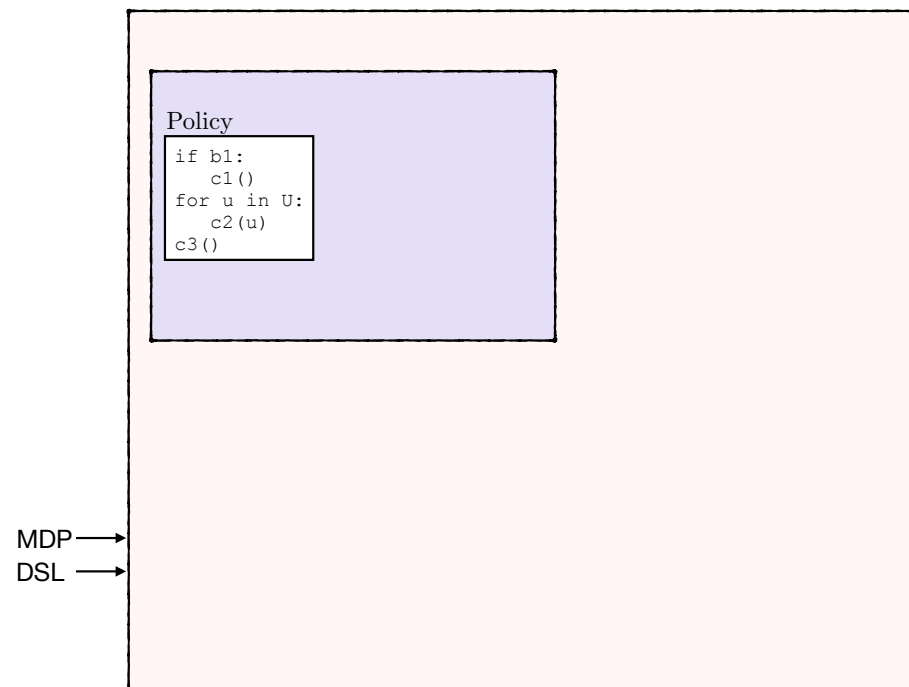
Library-Induced Semantic Spaces



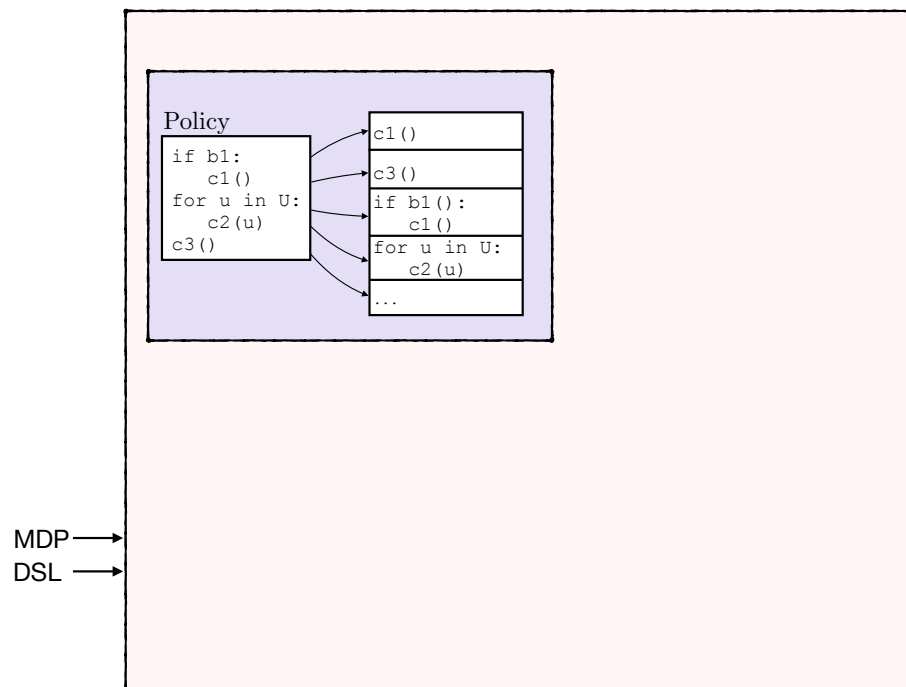
Library-Induced Semantic Spaces



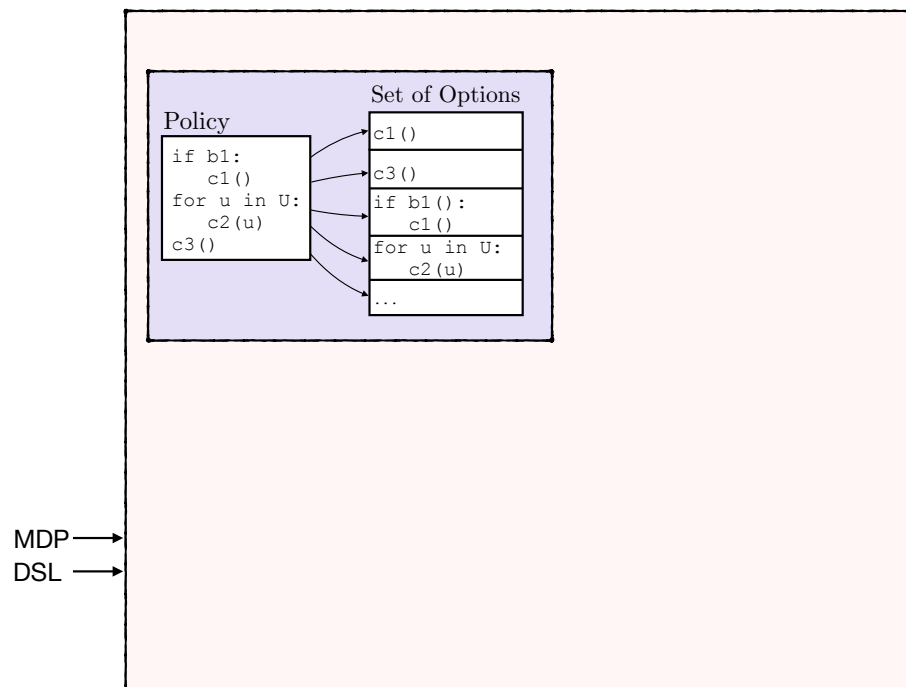
Library-Induced Semantic Spaces



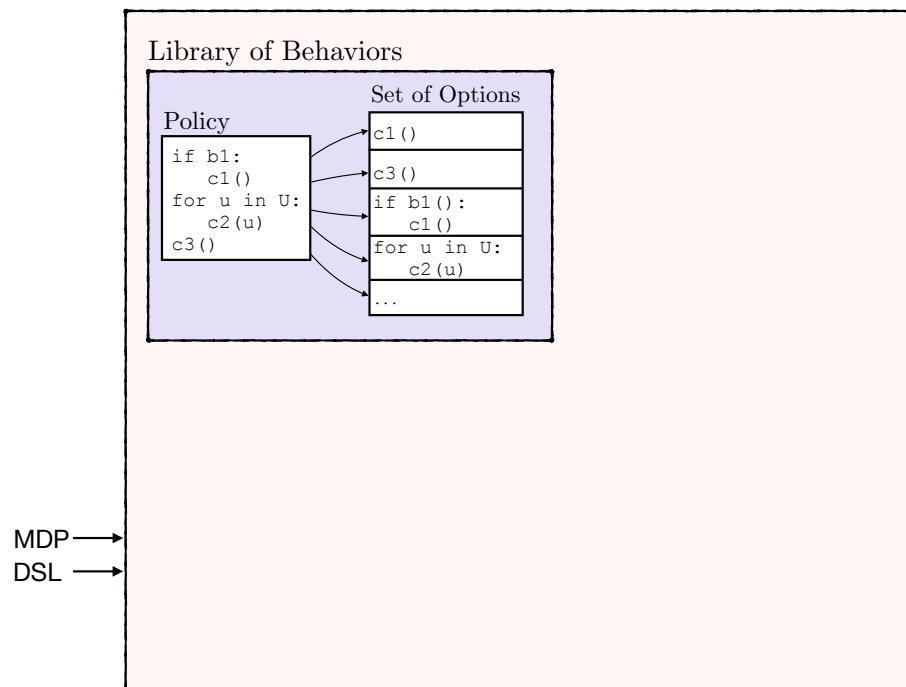
Library-Induced Semantic Spaces



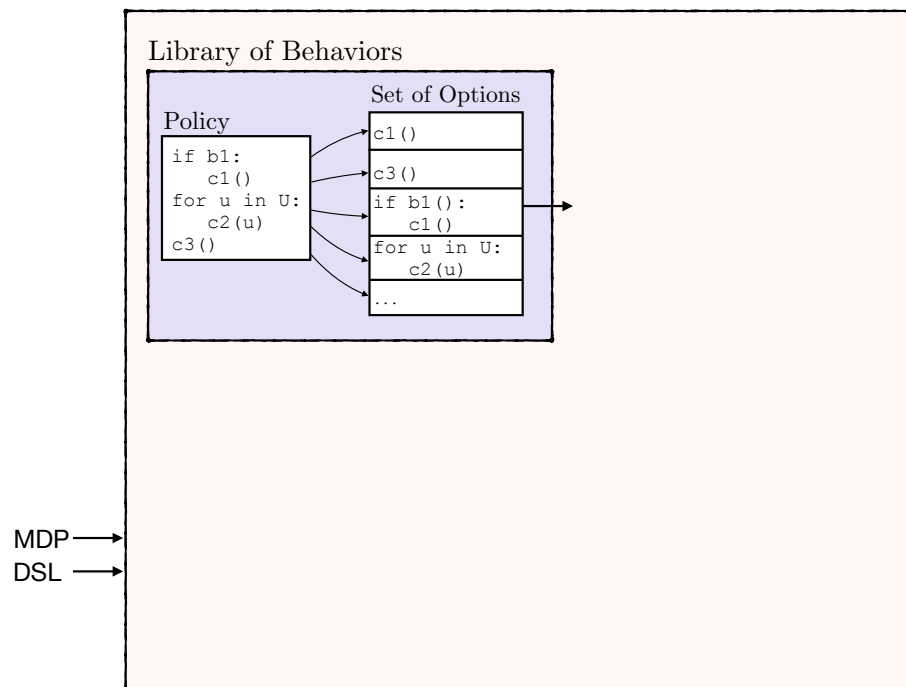
Library-Induced Semantic Spaces



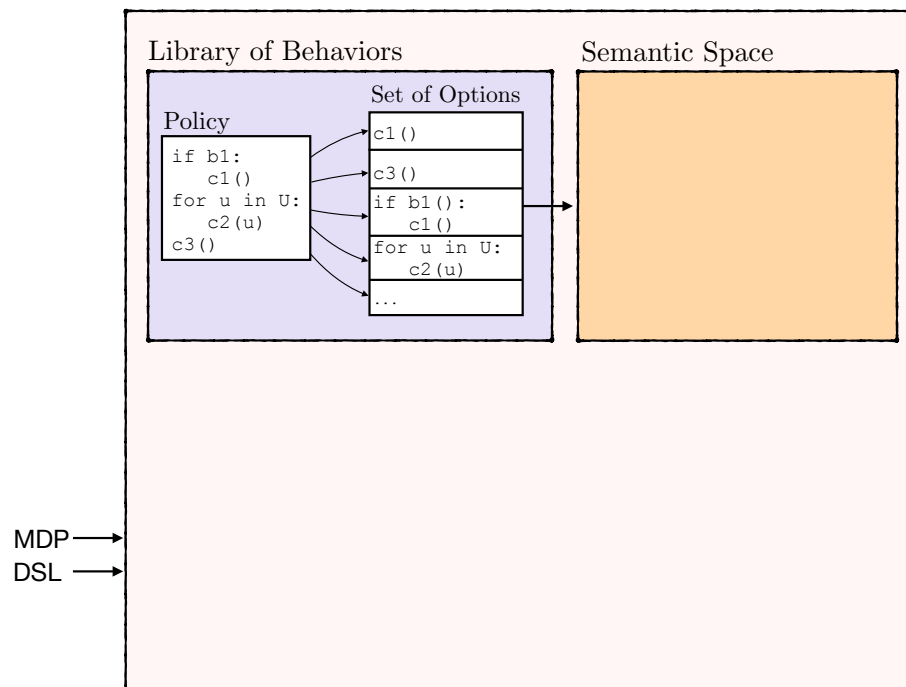
Library-Induced Semantic Spaces



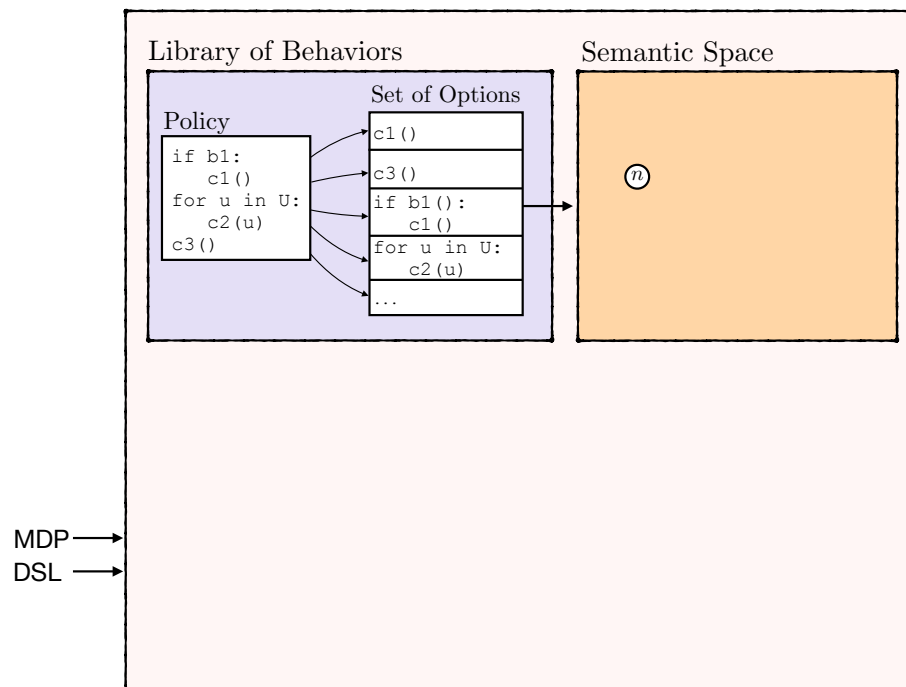
Library-Induced Semantic Spaces



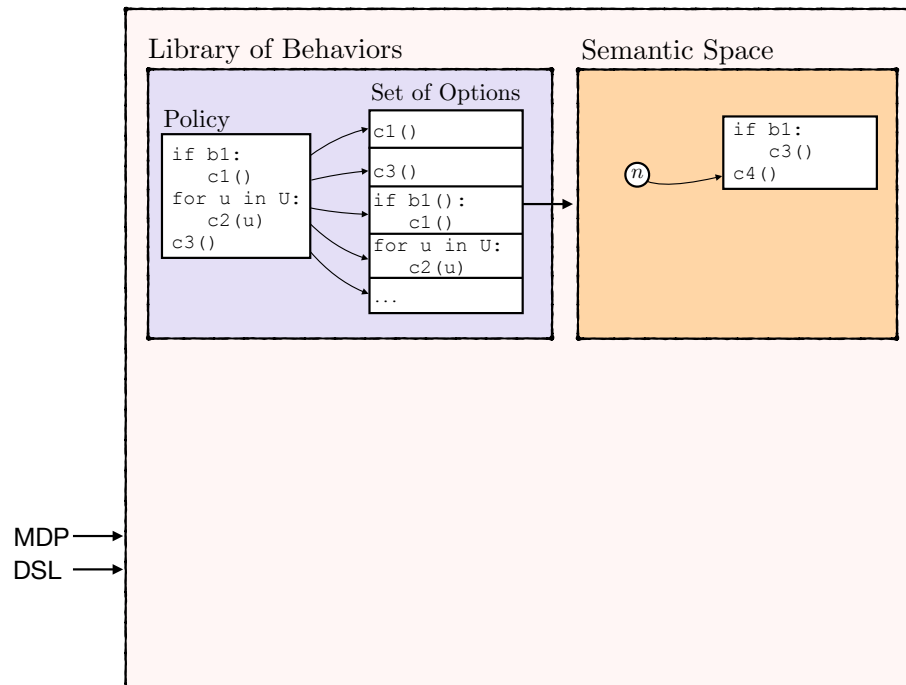
Library-Induced Semantic Spaces



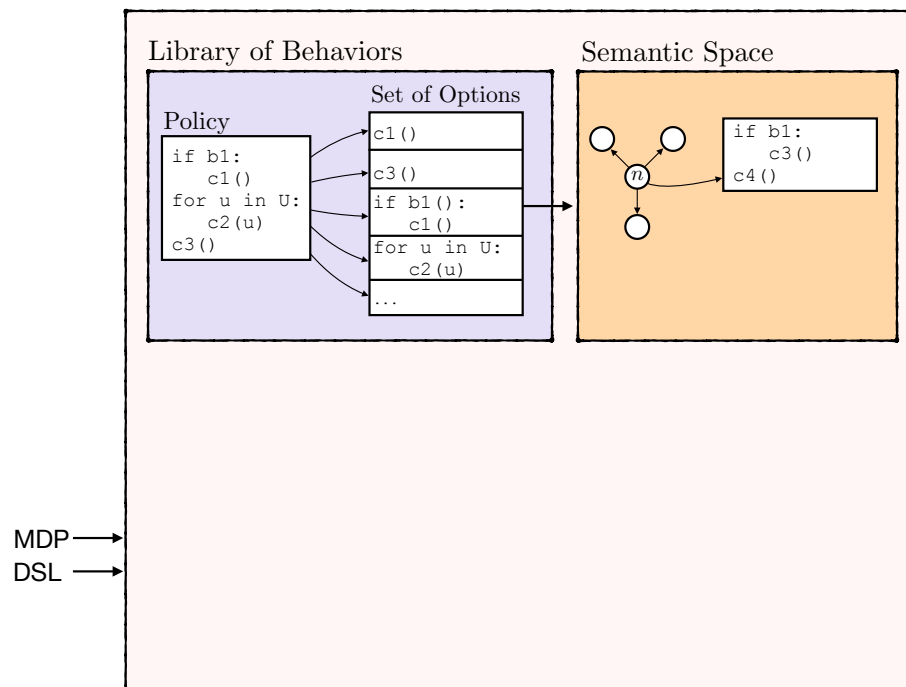
Library-Induced Semantic Spaces



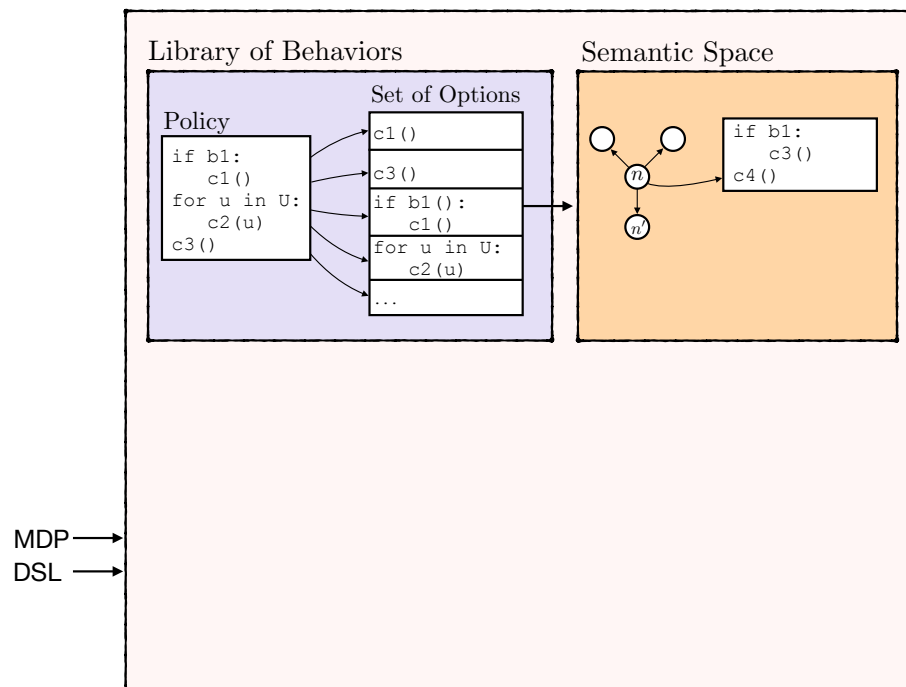
Library-Induced Semantic Spaces



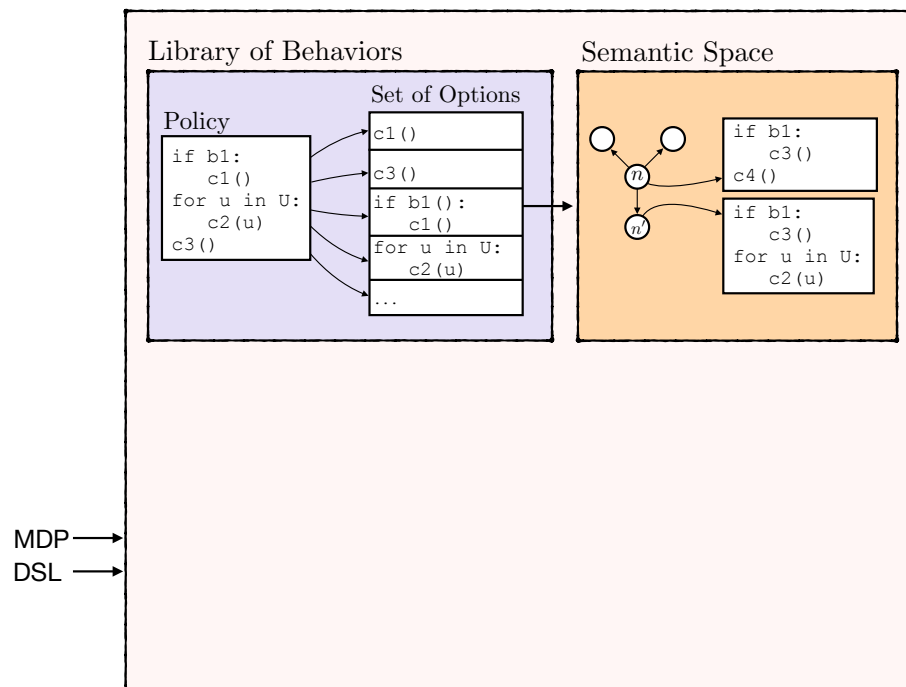
Library-Induced Semantic Spaces



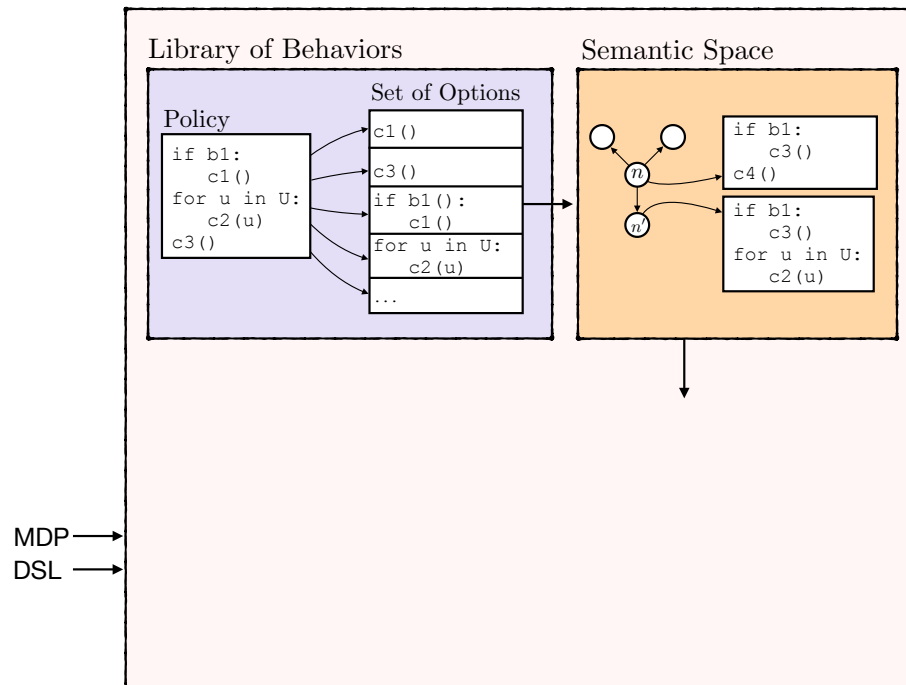
Library-Induced Semantic Spaces



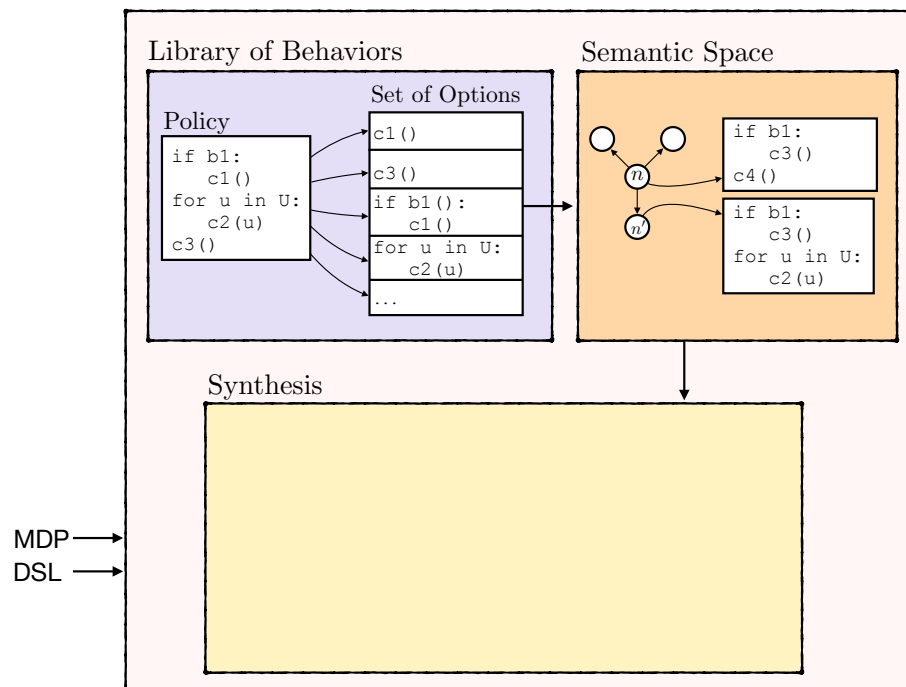
Library-Induced Semantic Spaces



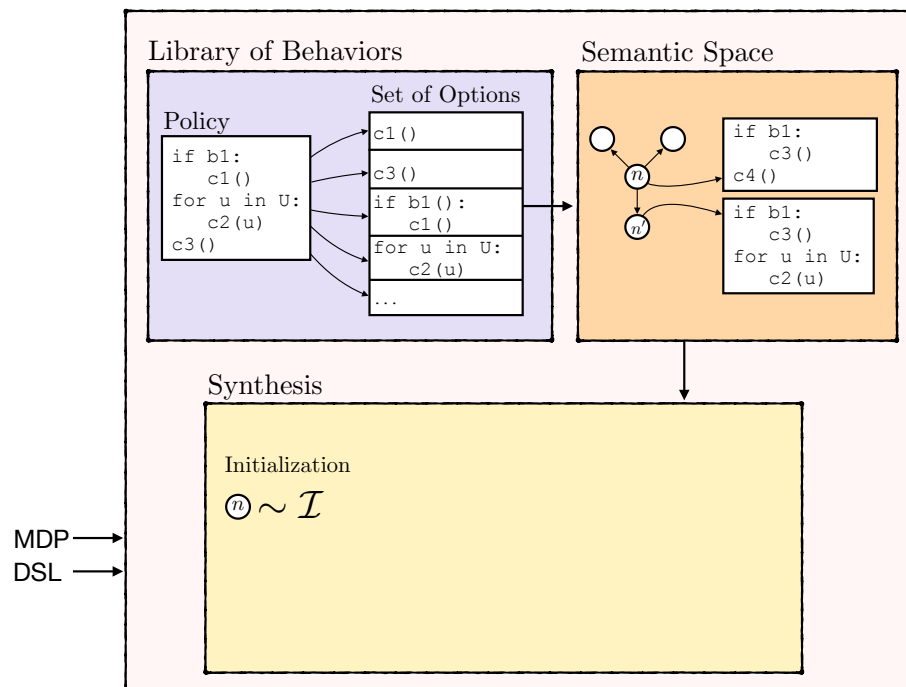
Library-Induced Semantic Spaces



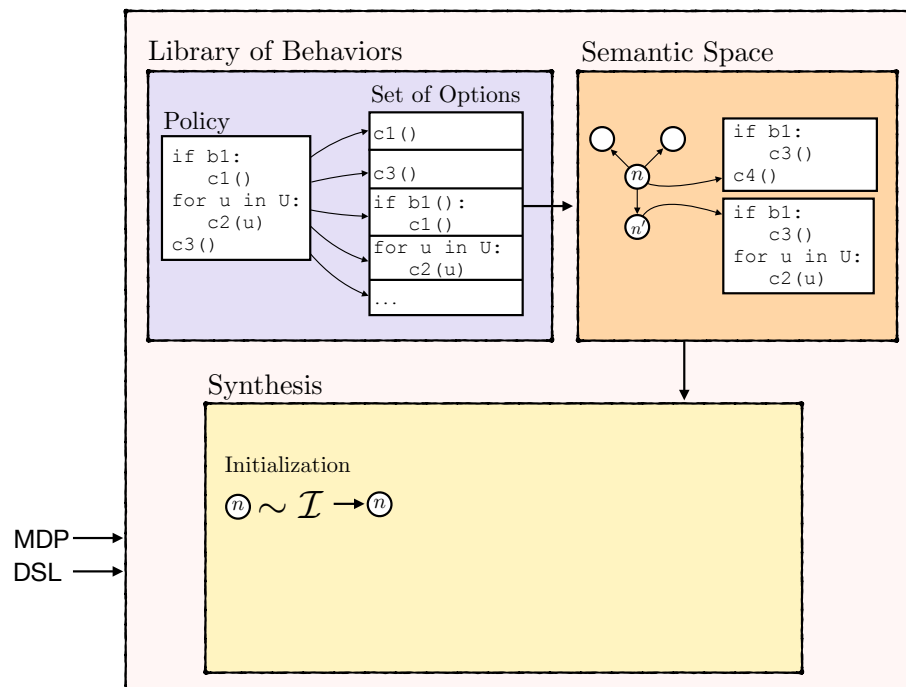
Library-Induced Semantic Spaces



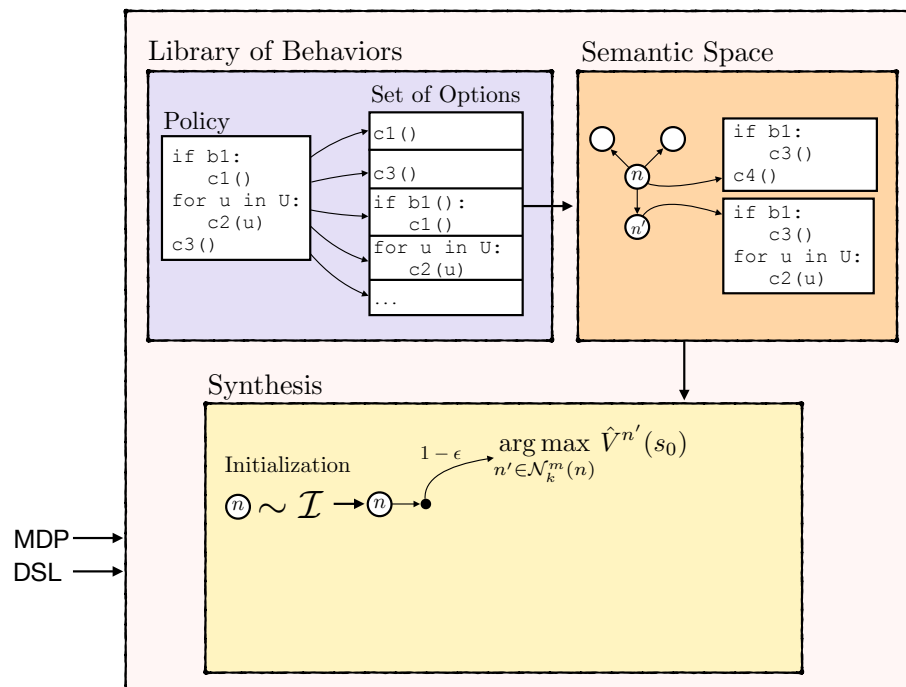
Library-Induced Semantic Spaces



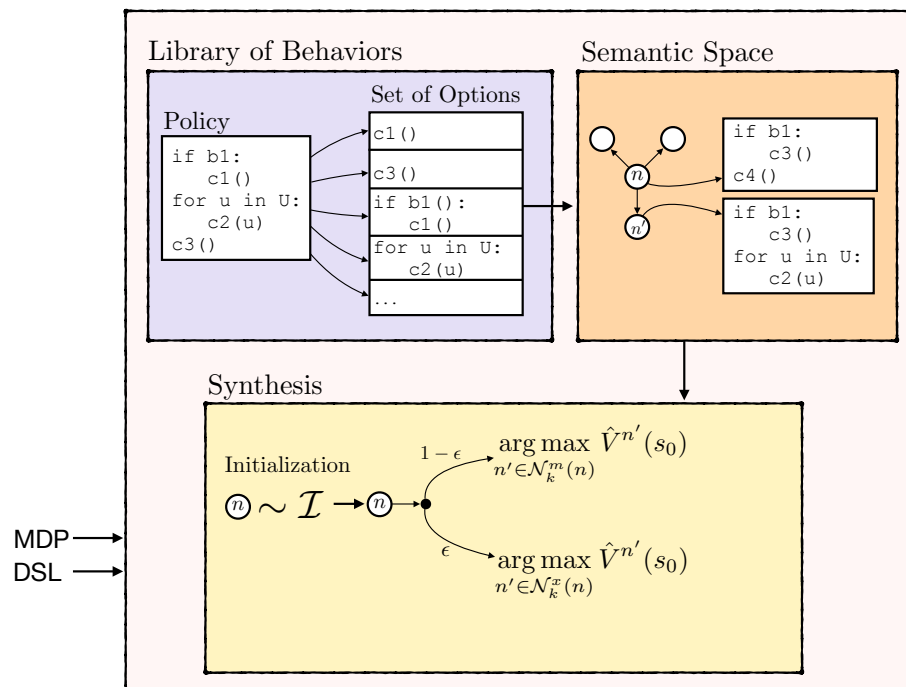
Library-Induced Semantic Spaces



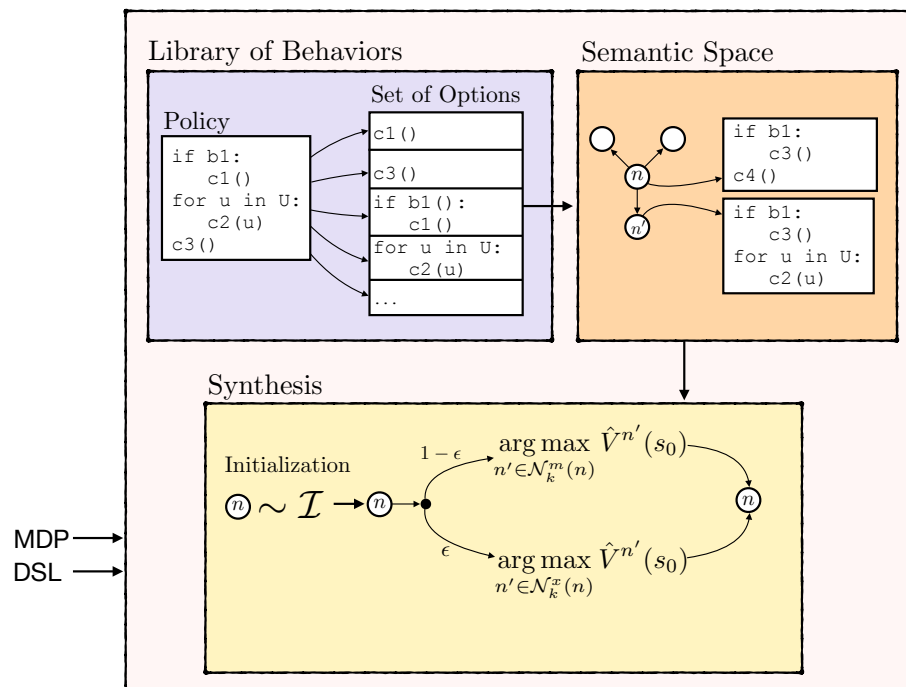
Library-Induced Semantic Spaces



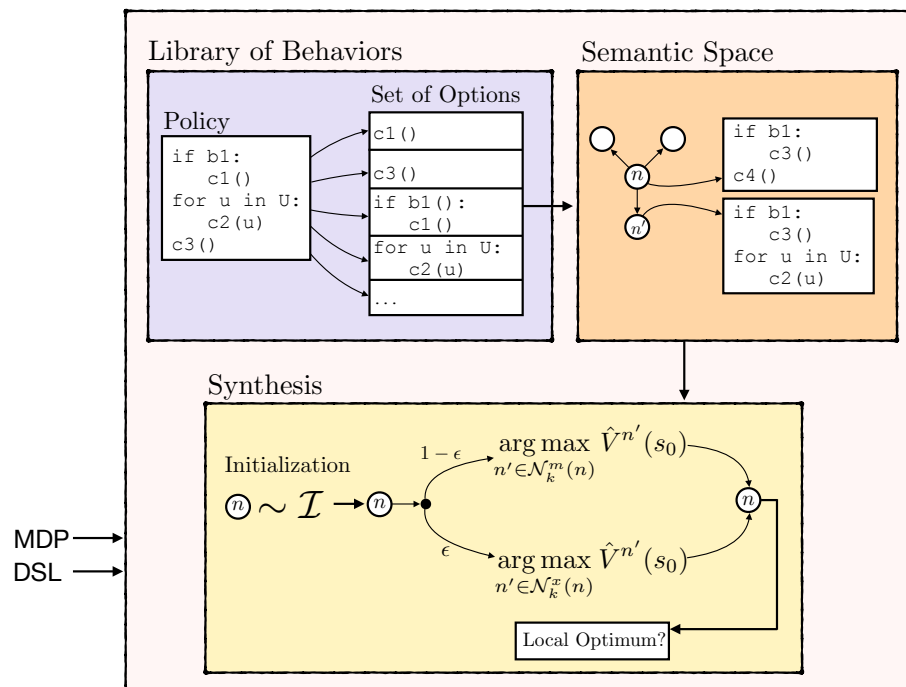
Library-Induced Semantic Spaces



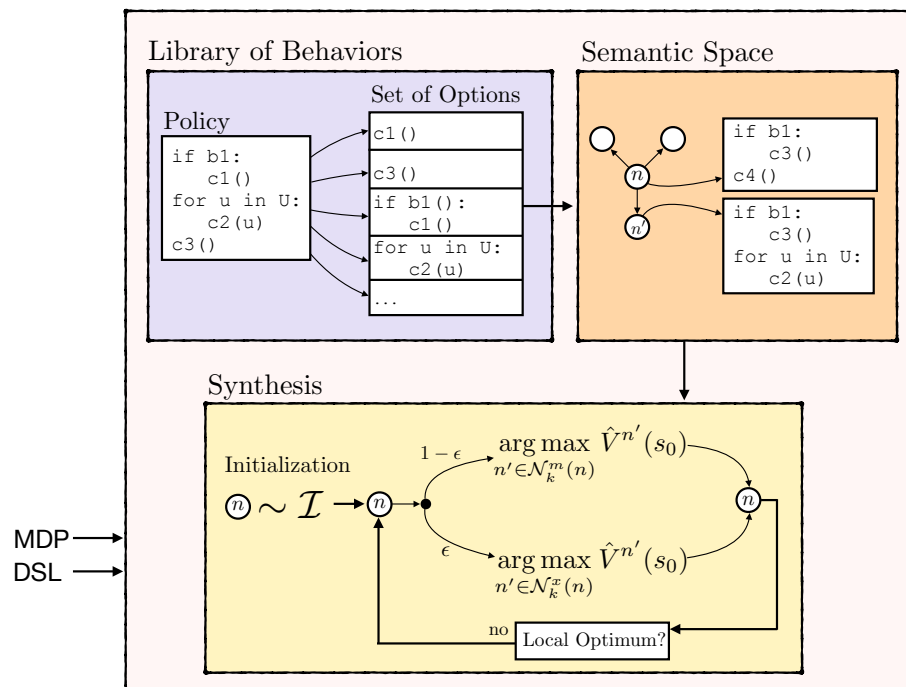
Library-Induced Semantic Spaces



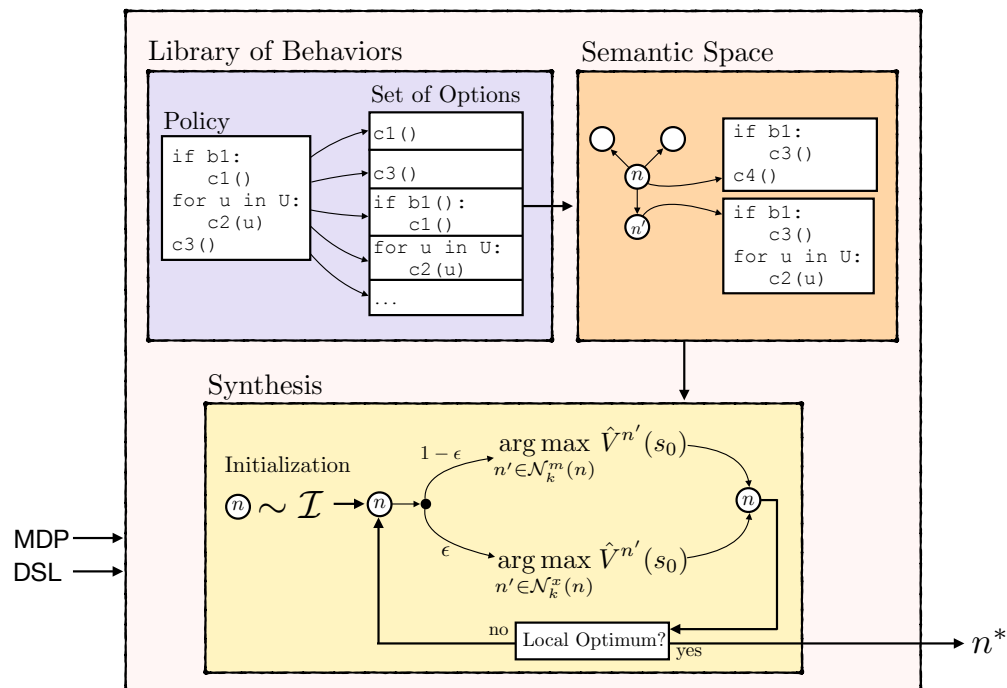
Library-Induced Semantic Spaces



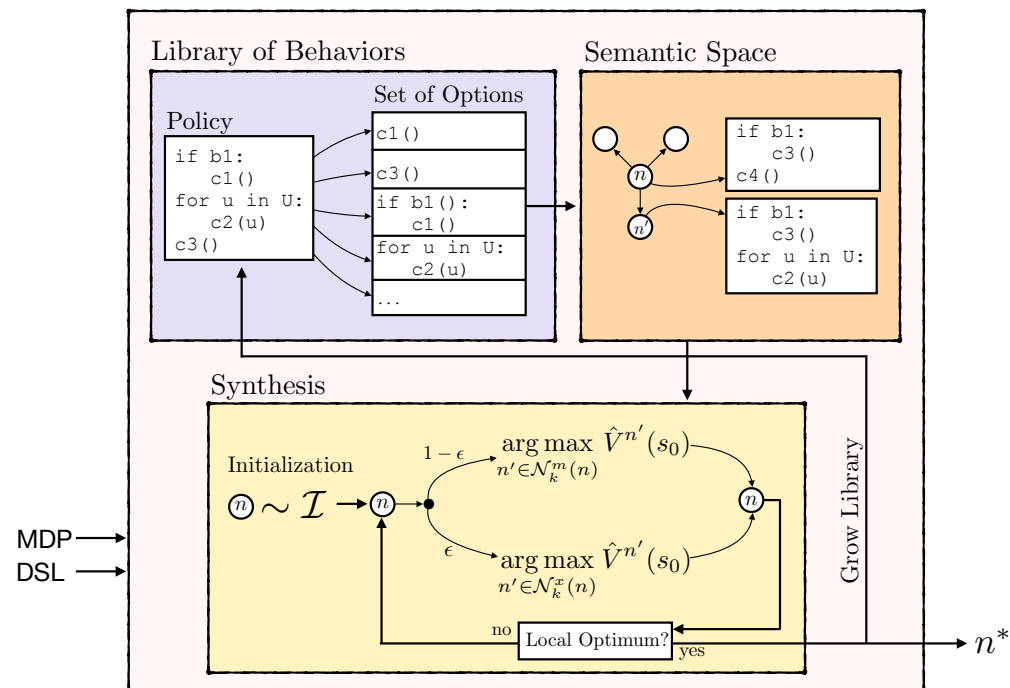
Library-Induced Semantic Spaces



Library-Induced Semantic Spaces



Library-Induced Semantic Spaces



Library-Induced Semantic Spaces

Library-Induced Semantic Spaces

- ❖ Semantic spaces are more conducive to search.

Library-Induced Semantic Spaces

- ❖ Semantic spaces are more conducive to search.
 - ❖ *“give me the next behavior”*

Library-Induced Semantic Spaces

- ❖ Semantic spaces are more conducive to search.
 - ❖ *“give me the next behavior”*
- ❖ LLMs can give “innate” abilities to agents.

Library-Induced Semantic Spaces

- ❖ Semantic spaces are more conducive to search.
 - ❖ *“give me the next behavior”*
- ❖ LLMs can give “innate” abilities to agents.
- ❖ **Future:** learning how to learn (recognizer in DreamCoder).

Learning Programs with Gradient Descent

```
# Sales data for two stores
store_1_sales = [120.0, 250.5, 310.0, 95.0]
store_2_sales = [200.0, 150.0, 275.0, 100.0]

# Process data for Store 1
store_1_total = sum(store_1_sales)
if store_1_total > 500:
    store_1_total *= 0.9 # Apply a 10% discount
store_1_report = f"Store 1 Total Sales: ${store_1_total:.2f}"

# Process data for Store 2
store_2_total = sum(store_2_sales)
if store_2_total > 500:
    store_2_total *= 0.9 # Apply a 10% discount
store_2_report = f"Store 2 Total Sales: ${store_2_total:.2f}"

print(store_1_report)
print(store_2_report)
```

Learning Programs with Gradient Descent

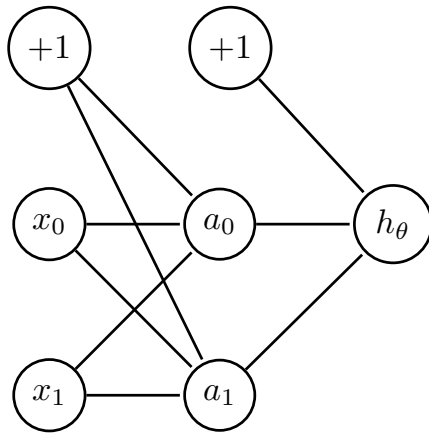
```
# Sales data for two stores
store_1_sales = [120.0, 250.5, 310.0, 95.0]
store_2_sales = [200.0, 150.0, 275.0, 100.0]

# Process data for Store 1
store_1_total = sum(store_1_sales)
if store_1_total > 500:
    store_1_total *= 0.9 # Apply a 10% discount
store_1_report = f"Store 1 Total Sales: ${store_1_total:.2f}"

# Process data for Store 2
store_2_total = sum(store_2_sales)
if store_2_total > 500:
    store_2_total *= 0.9 # Apply a 10% discount
store_2_report = f"Store 2 Total Sales: ${store_2_total:.2f}"

print(store_1_report)
print(store_2_report)
```

Learning Programs with Gradient Descent

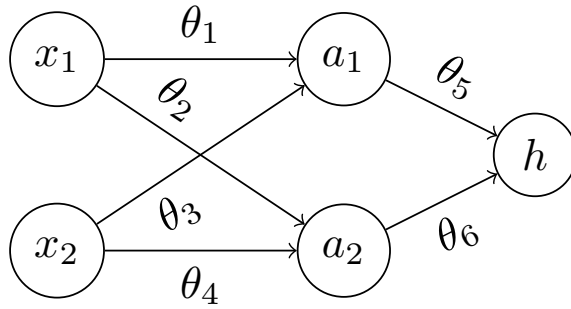
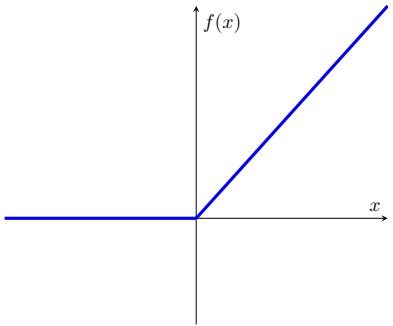


```
# Sales data for two stores
store_1_sales = [120.0, 250.5, 310.0, 95.0]
store_2_sales = [200.0, 150.0, 275.0, 100.0]

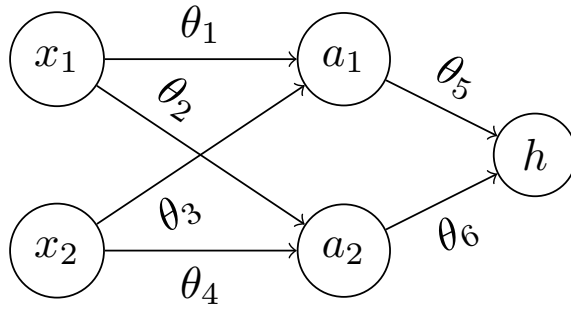
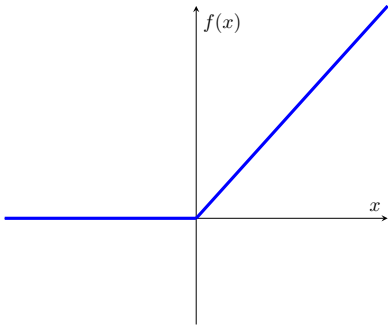
# Process data for Store 1
store_1_total = sum(store_1_sales)
if store_1_total > 500:
    store_1_total *= 0.9 # Apply a 10% discount
store_1_report = f"Store 1 Total Sales: ${store_1_total:.2f}"

# Process data for Store 2
store_2_total = sum(store_2_sales)
if store_2_total > 500:
    store_2_total *= 0.9 # Apply a 10% discount
store_2_report = f"Store 2 Total Sales: ${store_2_total:.2f}"

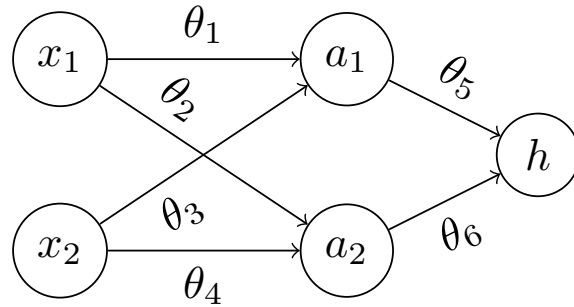
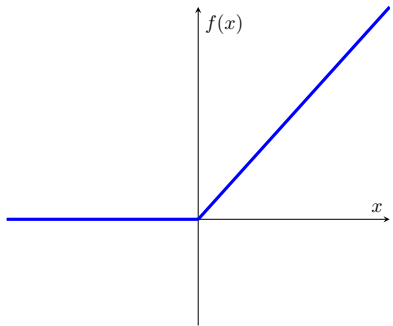
print(store_1_report)
print(store_2_report)
```



Orfanos & Lelis (2023)

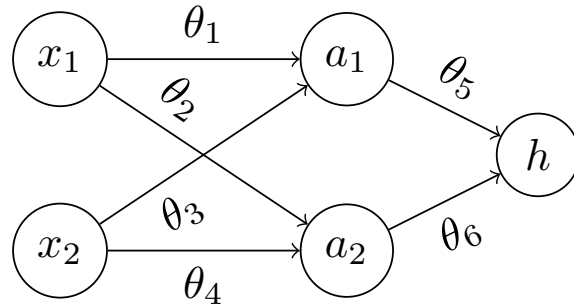
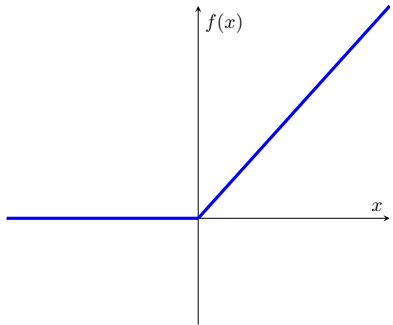


a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0



a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0

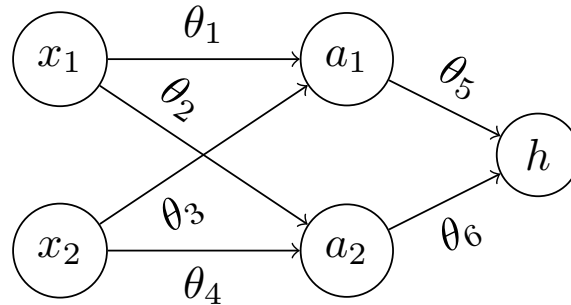
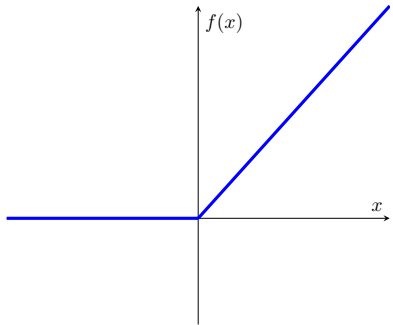
a_2 can be either $x_1\theta_2 + x_2\theta_4$ or 0



a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0

a_2 can be either $x_1\theta_2 + x_2\theta_4$ or 0

if $a_1 = x_1\theta_1 + x_2\theta_3$ and $a_2 = x_1\theta_2 + x_2\theta_4$ then

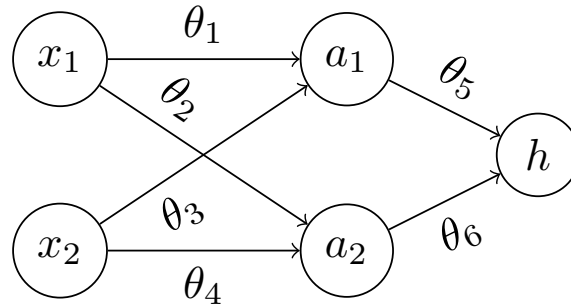
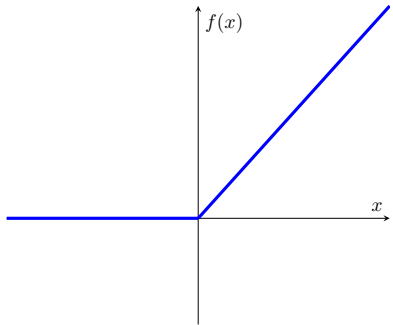


a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0

a_2 can be either $x_1\theta_2 + x_2\theta_4$ or 0

if $a_1 = x_1\theta_1 + x_2\theta_3$ and $a_2 = x_1\theta_2 + x_2\theta_4$ then

$$\begin{aligned}
 h &= \theta_5 a_1 + \theta_6 a_2 \\
 &= \theta_5 (x_1\theta_1 + x_2\theta_3) + \theta_6 (x_1\theta_2 + x_2\theta_4) \\
 &= x_1(\theta_5\theta_1 + \theta_6\theta_2) + x_2(\theta_5\theta_3 + \theta_6\theta_4) \\
 &= x_1\theta' + x_2\theta''
 \end{aligned}$$



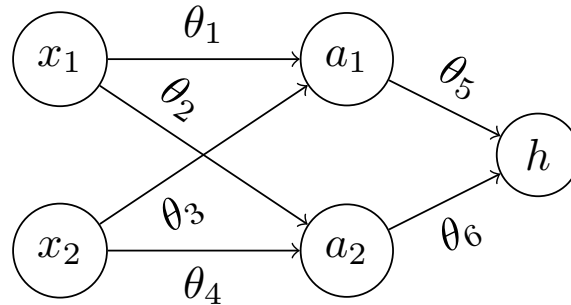
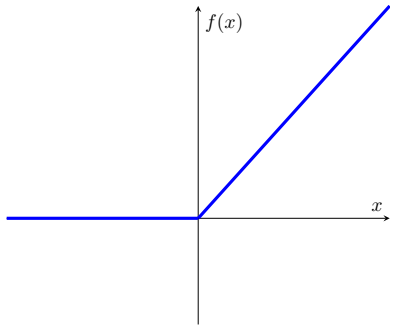
a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0

a_2 can be either $x_1\theta_2 + x_2\theta_4$ or 0

The network above is equivalent to:

if $a_1 = x_1\theta_1 + x_2\theta_3$ and $a_2 = x_1\theta_2 + x_2\theta_4$ then

$$\begin{aligned}
 h &= \theta_5 a_1 + \theta_6 a_2 \\
 &= \theta_5 (x_1 \theta_1 + x_2 \theta_3) + \theta_6 (x_1 \theta_2 + x_2 \theta_4) \\
 &= x_1 (\theta_5 \theta_1 + \theta_6 \theta_2) + x_2 (\theta_5 \theta_3 + \theta_6 \theta_4) \\
 &= x_1 \theta' + x_2 \theta''
 \end{aligned}$$



a_1 can be either $x_1\theta_1 + x_2\theta_3$ or 0

a_2 can be either $x_1\theta_2 + x_2\theta_4$ or 0

The network above is equivalent to:

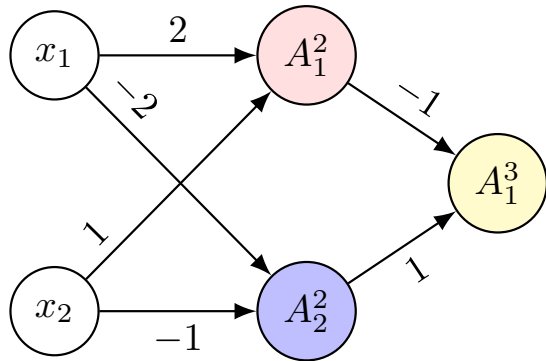
if $x_1\theta_1 + x_2\theta_3 > 0$
if $x_1\theta_2 + x_2\theta_4 > 0$
then $x_1\theta' + x_2\theta''$
else ...

...

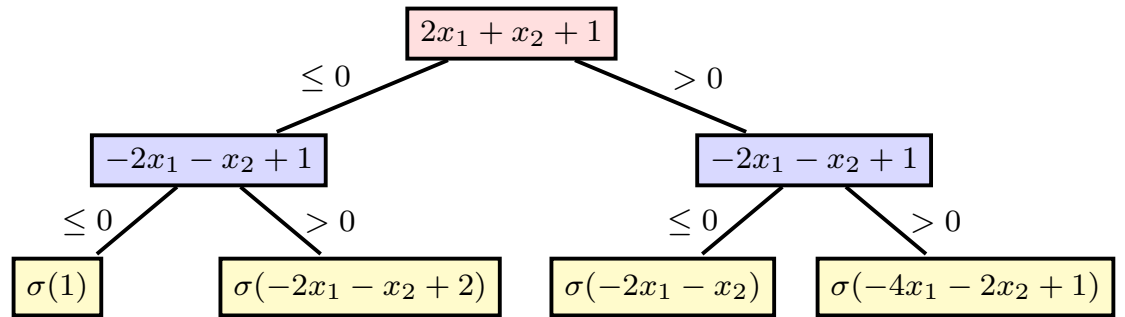
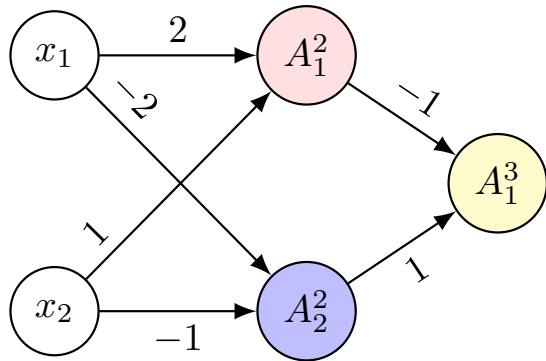
if $a_1 = x_1\theta_1 + x_2\theta_3$ and $a_2 = x_1\theta_2 + x_2\theta_4$ then

$$\begin{aligned} h &= \theta_5 a_1 + \theta_6 a_2 \\ &= \theta_5 (x_1\theta_1 + x_2\theta_3) + \theta_6 (x_1\theta_2 + x_2\theta_4) \\ &= x_1(\theta_5\theta_1 + \theta_6\theta_2) + x_2(\theta_5\theta_3 + \theta_6\theta_4) \\ &= x_1\theta' + x_2\theta'' \end{aligned}$$

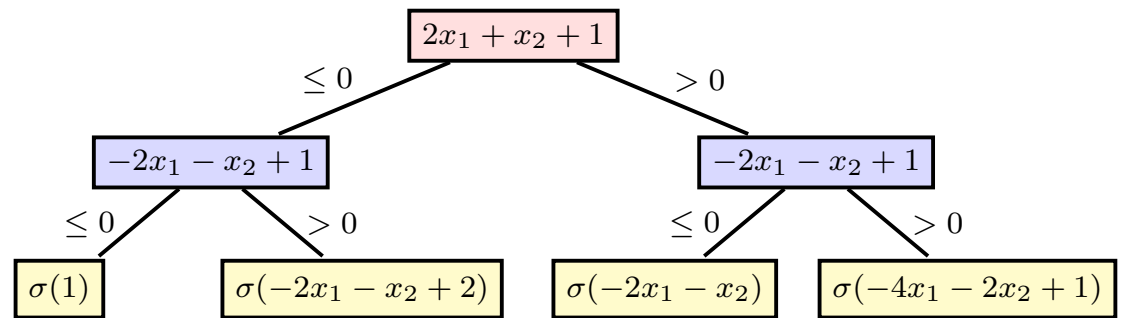
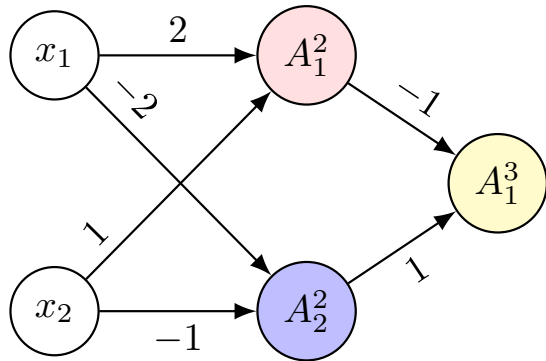
Neural Trees



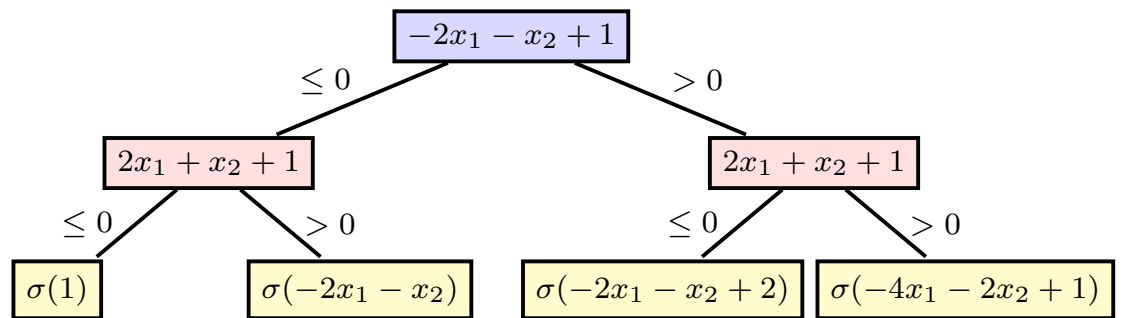
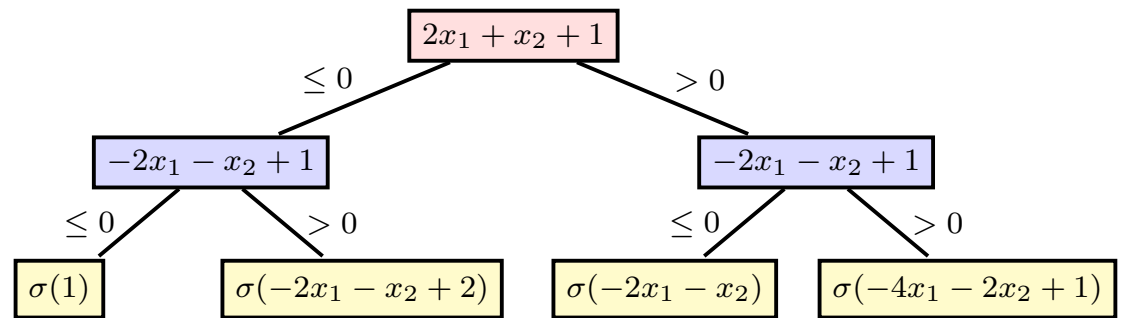
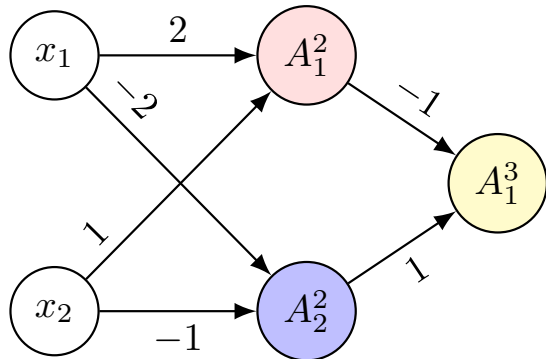
Neural Trees



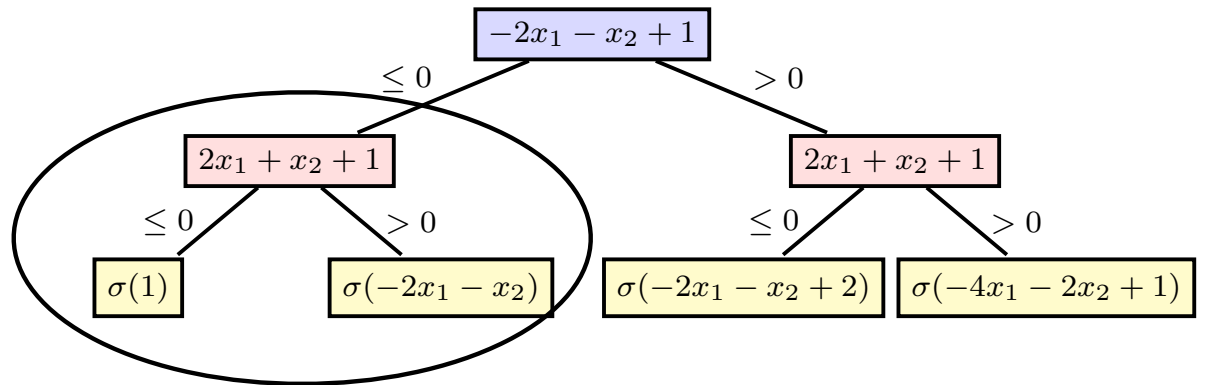
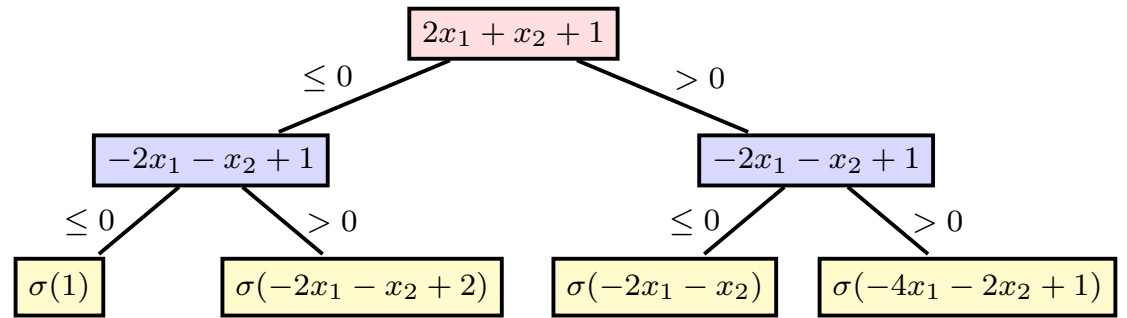
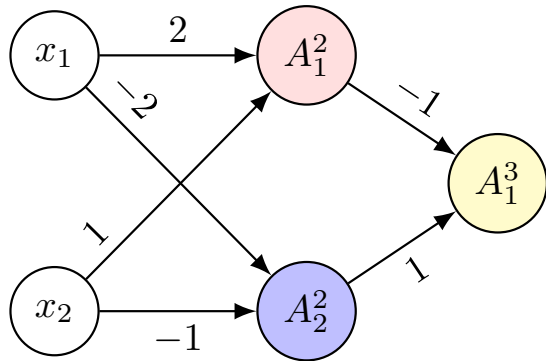
Neural Trees



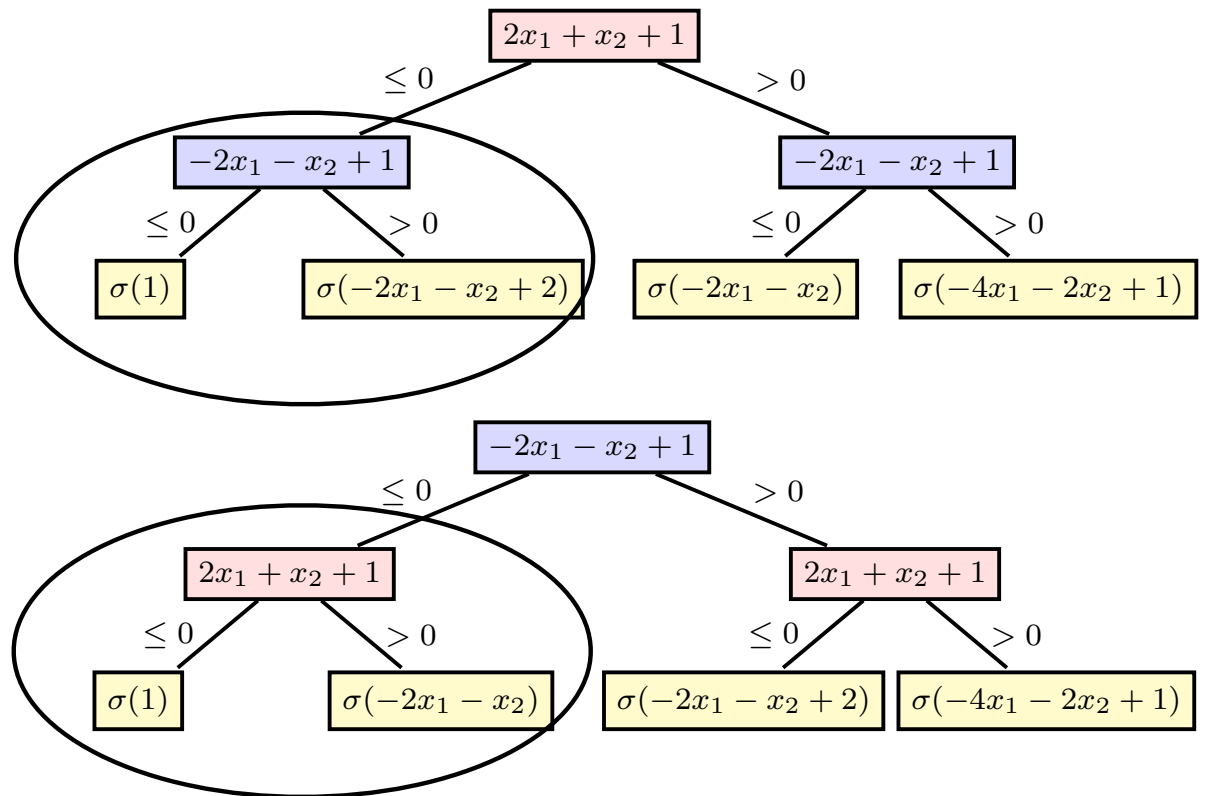
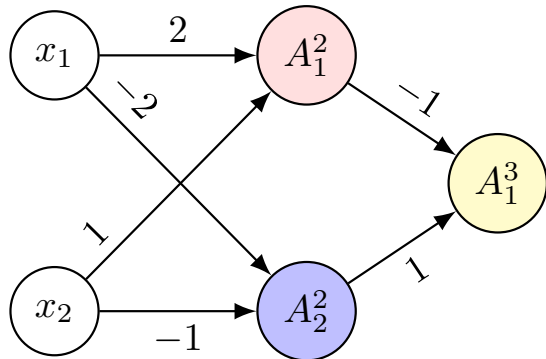
Neural Trees



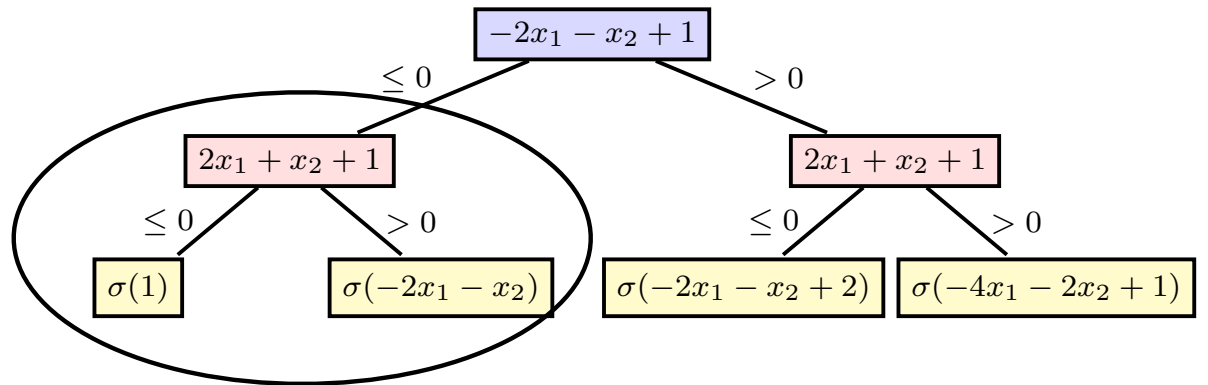
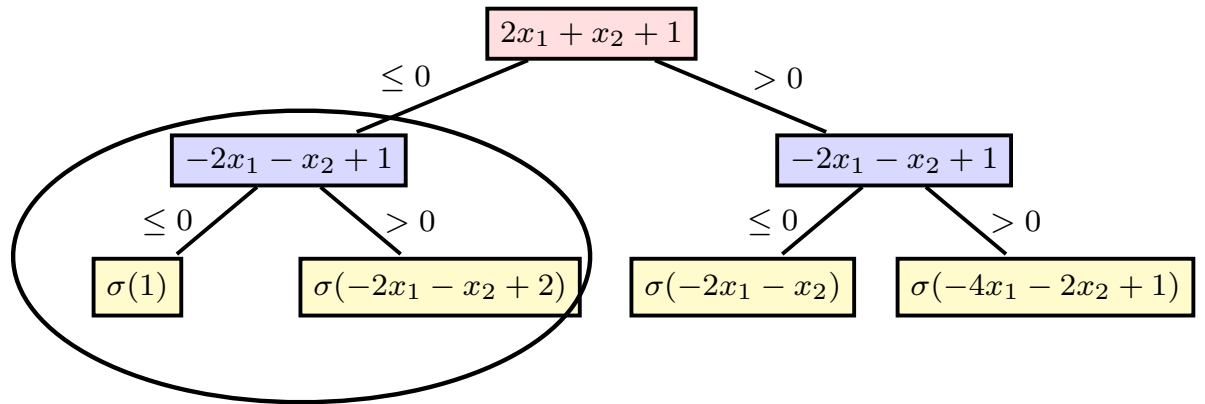
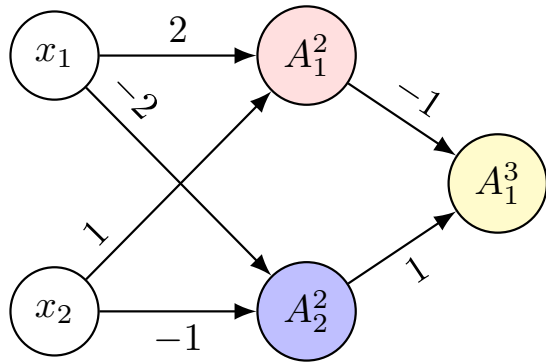
Neural Trees



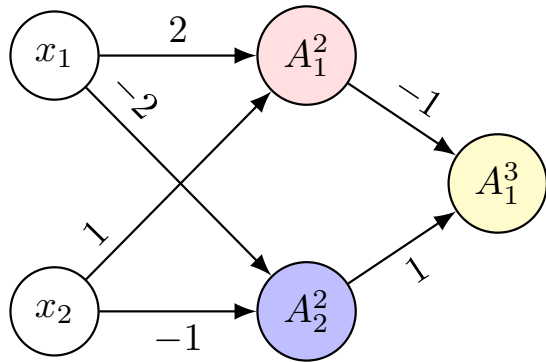
Neural Trees



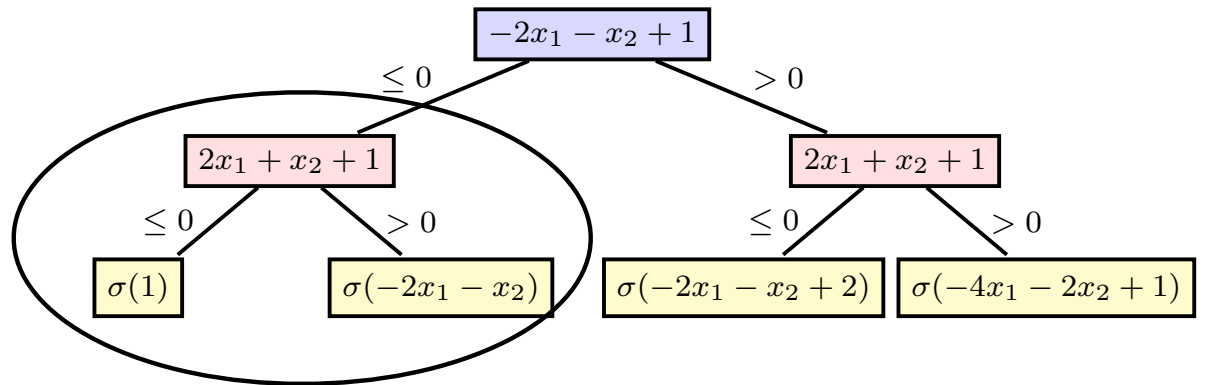
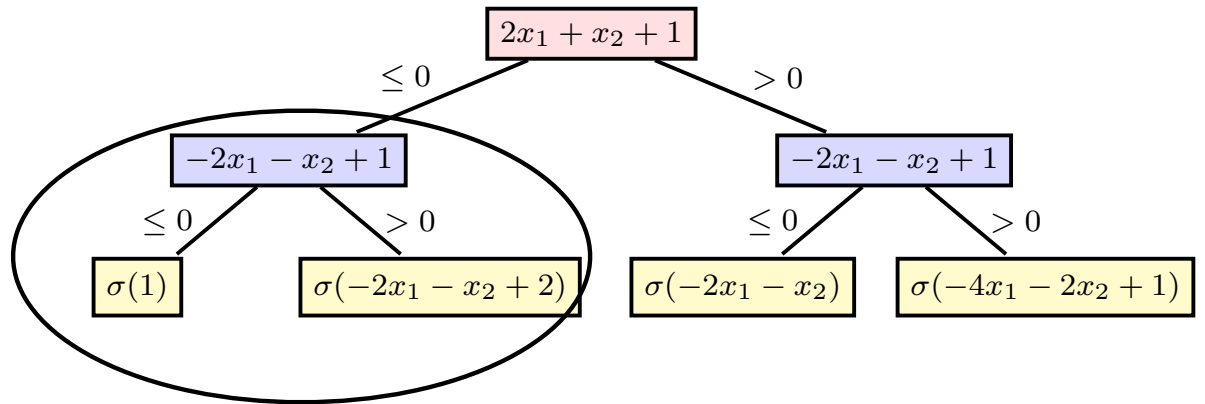
Neural Trees



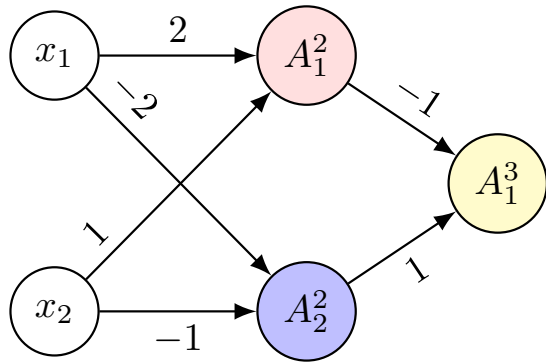
Neural Trees



Three possibilities for each neuron:

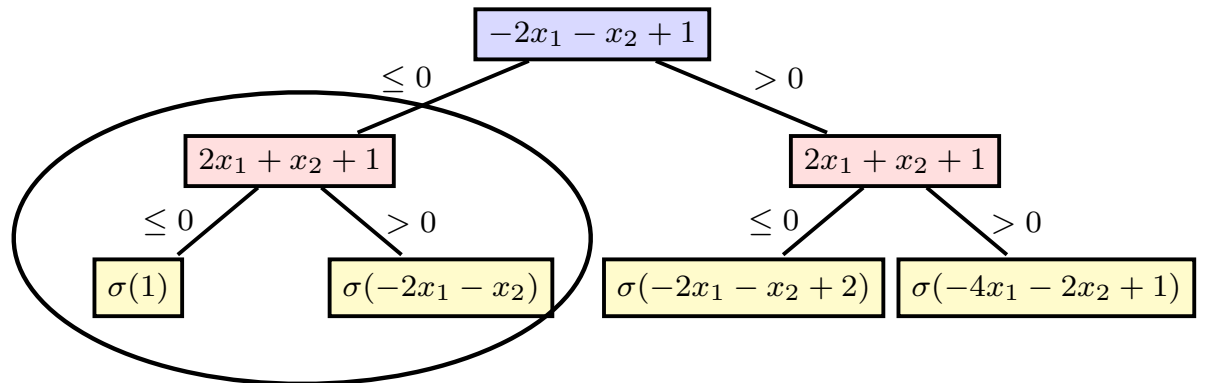
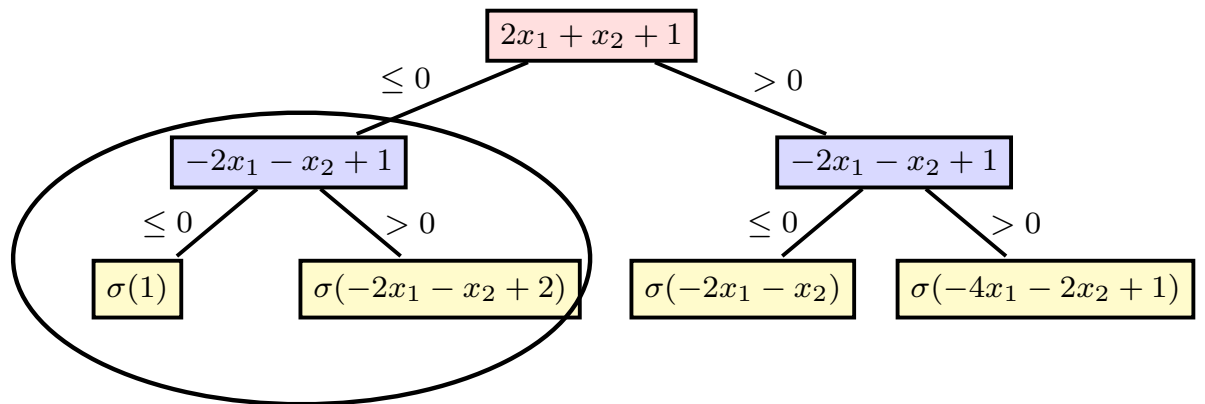


Neural Trees

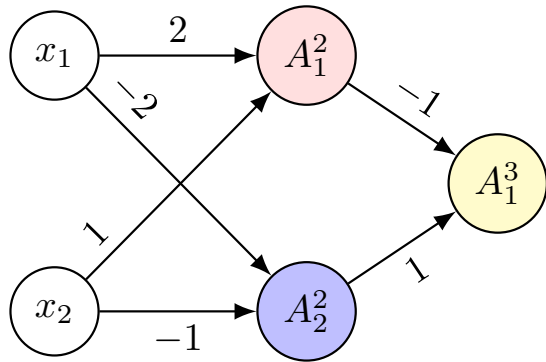


Three possibilities for each neuron:

1. Inactive

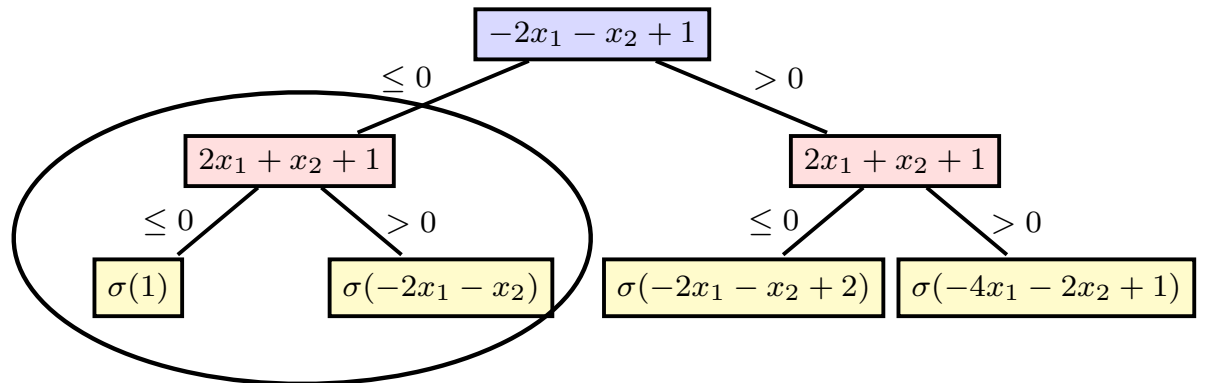
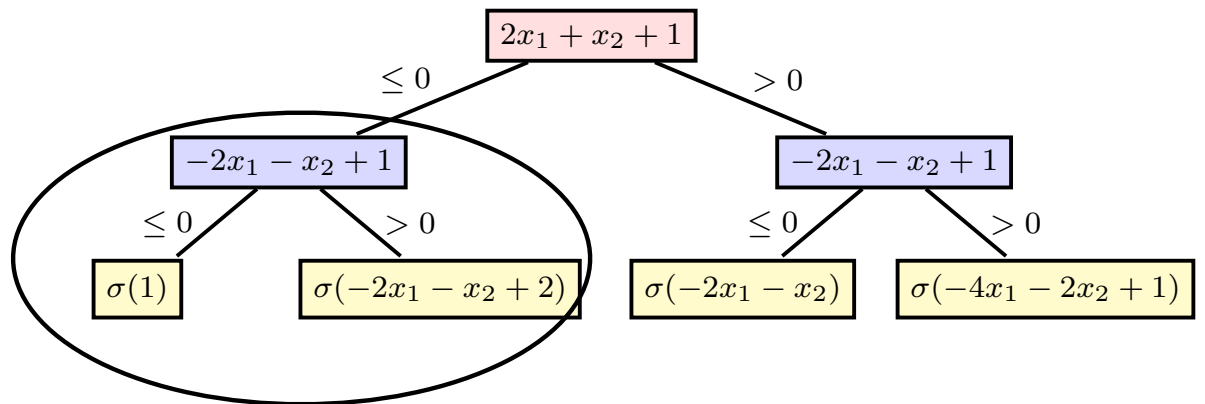


Neural Trees

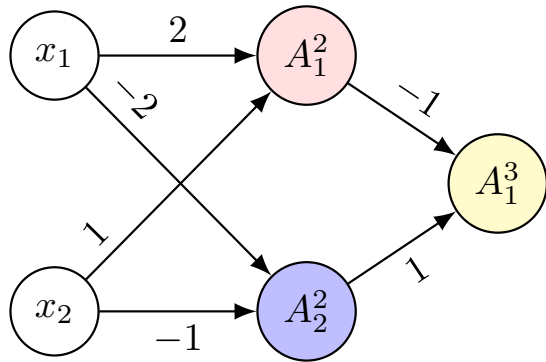


Three possibilities for each neuron:

1. Inactive
2. Active

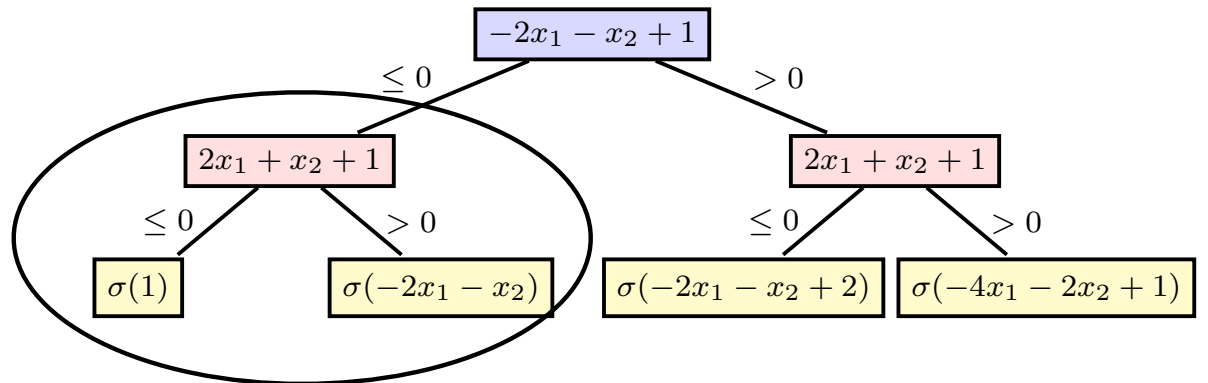
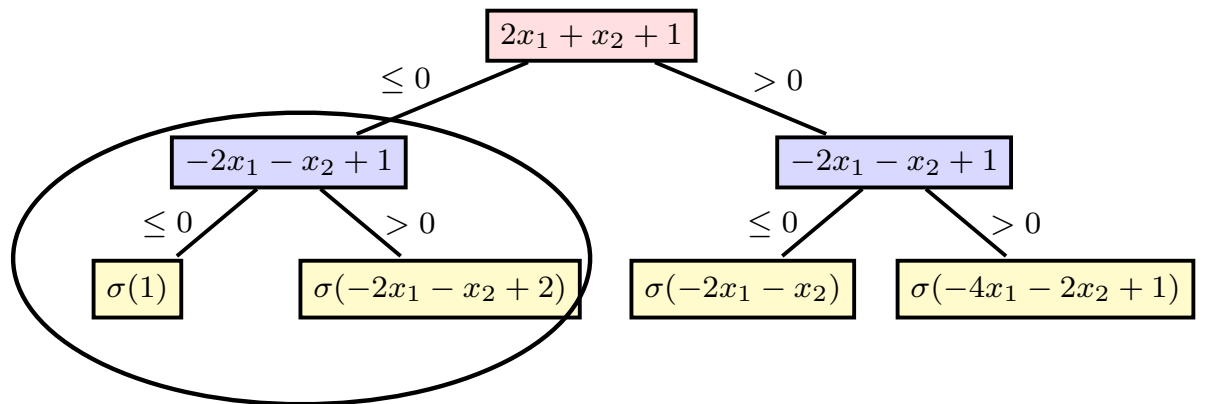


Neural Trees

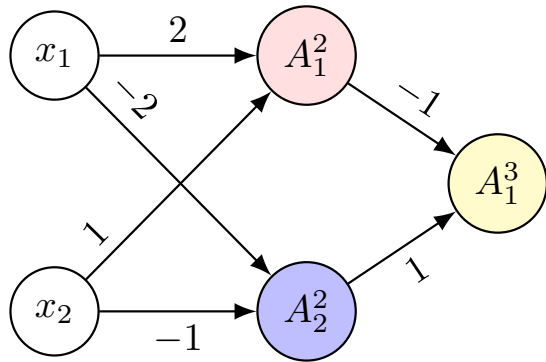


Three possibilities for each neuron:

1. Inactive
2. Active
3. Part of the program



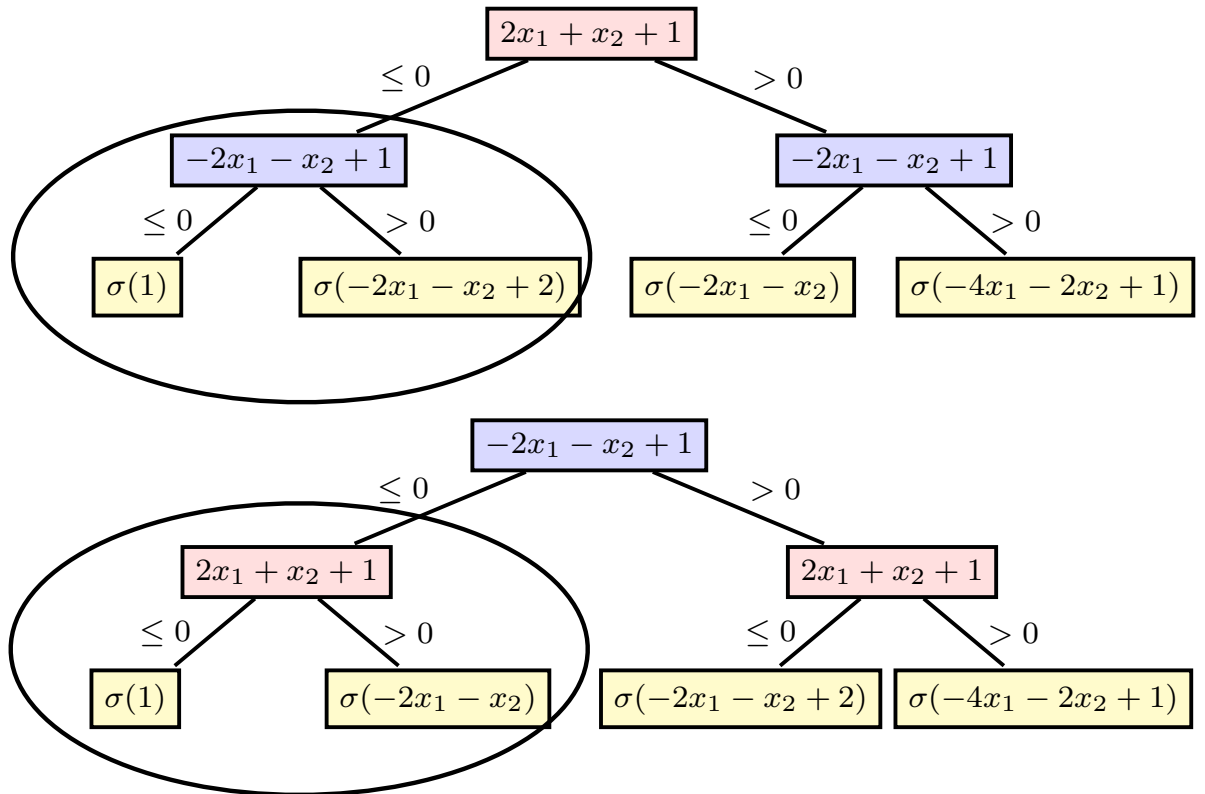
Neural Trees



Three possibilities for each neuron:

1. Inactive
2. Active
3. Part of the program

For N hidden neurons, 3^N subprograms.



Two Effects of a Growing Library

Two Effects of a Growing Library

- ❖ **Problem becomes harder:** more functions to learn how to use.

Two Effects of a Growing Library

- ❖ **Problem becomes harder:** more functions to learn how to use.
- ❖ **Problem becomes easier:** helpful functions make it easier.

Two Effects of a Growing Library

- ❖ **Problem becomes harder:** more functions to learn how to use.
- ❖ **Problem becomes easier:** helpful functions make it easier.
- ❖ Select a subset of programs while minimizing the Levin loss
(Alikhasi & Lelis, ICLR 2024) (Orseau, Lelis, Lattimore & Weber, Neurips 2018).

Two Effects of a Growing Library

- ❖ **Problem becomes harder:** more functions to learn how to use.
- ❖ **Problem becomes easier:** helpful functions make it easier.
- ❖ Select a subset of programs while minimizing the Levin loss
(Alikhasi & Lelis, ICLR 2024) (Orseau, Lelis, Lattimore & Weber, Neurips 2018).
- ❖ Evaluates all programs. :-)

Levin Loss

$$\frac{d(n)}{\pi(n)}$$

Orseau et al. (2018)

Levin Loss

$$\frac{d(n)}{\pi(n)}$$

$\frac{1}{\pi(n)}$ is the expected of number of samples to observe n .

Levin Loss

$$\frac{d(n)}{\pi(n)}$$

$\frac{1}{\pi(n)}$ is the expected of number of samples to observe n .

$d(n)$ is the cost of each sample.

Selecting a Subset of Options

Selecting a Subset of Options

- ❖ Iteratively select the option that minimizes the Levin loss.

Selecting a Subset of Options

- ❖ Iteratively select the option that minimizes the Levin loss.
- ❖ Stop when Levin loss goes up for the selected subset.

Dec-Options

Alikhasi & Lelis (2024)

Dec-Options

- ❖ Learn policy to solve some problems.

Dec-Options

- ❖ Learn policy to solve some problems.
- ❖ Decompose them into sub-programs.

Dec-Options

- ❖ Learn policy to solve some problems.
- ❖ Decompose them into sub-programs.
- ❖ Select a sub-set of them to use as options.

Dec-Options

- ❖ Learn policy to solve some problems.
- ❖ Decompose them into sub-programs.
- ❖ Select a sub-set of them to use as options.
- ❖ Use options in downstream tasks.

Alikhasi & Lelis (2024)

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2

	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M		M
	A M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M		M
	A M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M		M
	A M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M A	
M		M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	A
M		M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M		A M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
M	A	M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2



	M	
A M		M
	M	

Extracting Modules from Networks

0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2

	M	
A M		M
	M	

Extracting Modules from Networks

down **up** **up** **right** **down** **left** **left**
0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2

	M	
A M		M
	M	

Extracting Modules from Networks

down up up right down left left
0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2
option 3 option 2 option 2 option 2 option 4 option 2 option 4

	M	
A M		M
	M	

Extracting Modules from Networks

down **up** **up** **right** **down** **left** **left**
0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2
option 3 option 2 option 2 option 2 option 4 option 2 option 4

	M	
A M		M
	M	

Extracting Modules from Networks

down up up right down left left
0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2
option 3 option 2 option 2 option 2 option 4 option 2 option 4

	M	
A M		M
	M	

Learning Programs

Learning Programs

- ❖ Libraries of programs can be used to approximate the semantic space of a language.

Learning Programs

- ❖ Libraries of programs can be used to approximate the semantic space of a language.
- ❖ Libraries of programs can be learned by decomposing neural networks.

We Are Hiring! University of Alberta, Canada.

- ❖ Several (10+) AI+X faculty positions across campus.
- ❖ 3 CS Positions (Theory+ML, SE+ML, and Theory)
- ❖ Please spread the word!

