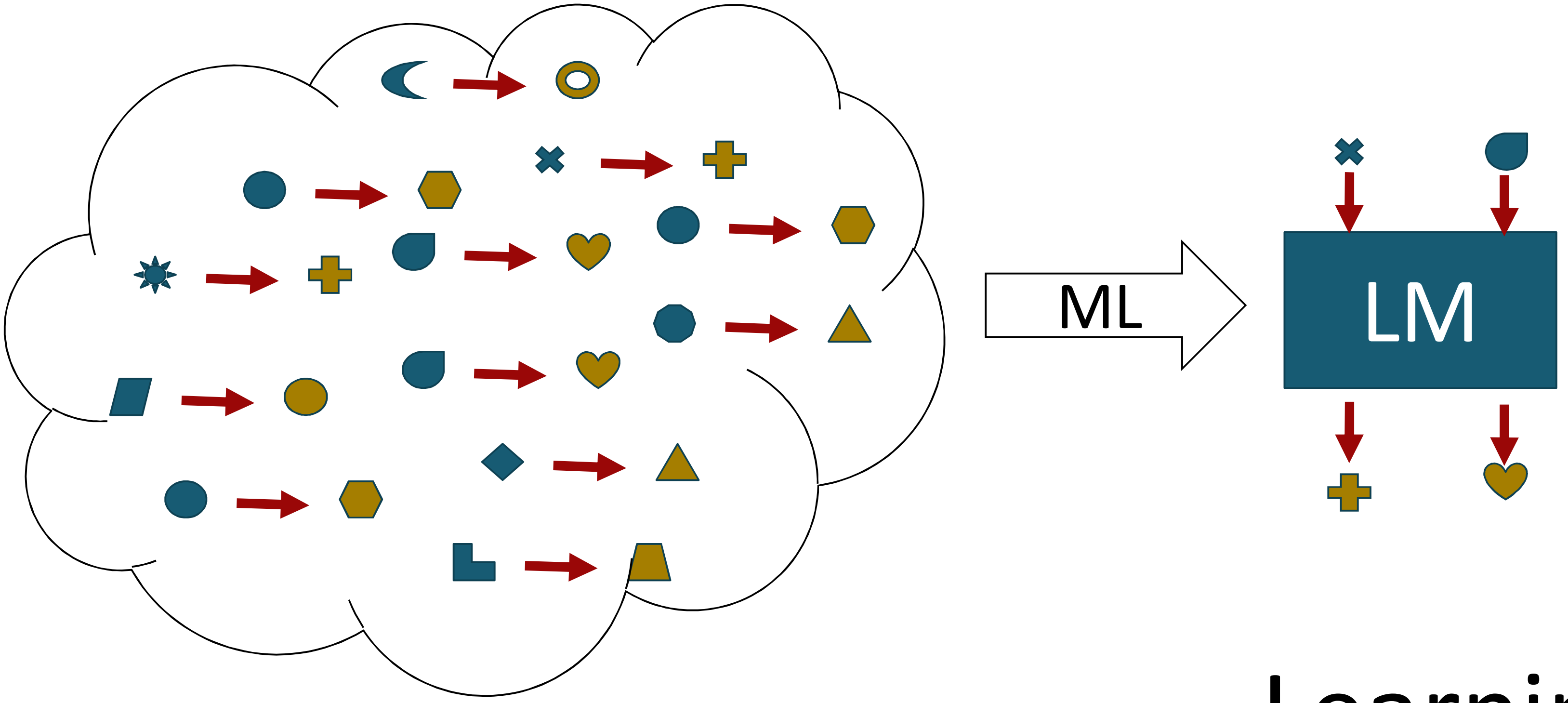
Learning + Synthesis

Armando Solar-Lezama

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

CSAIL

ML as Synthesis

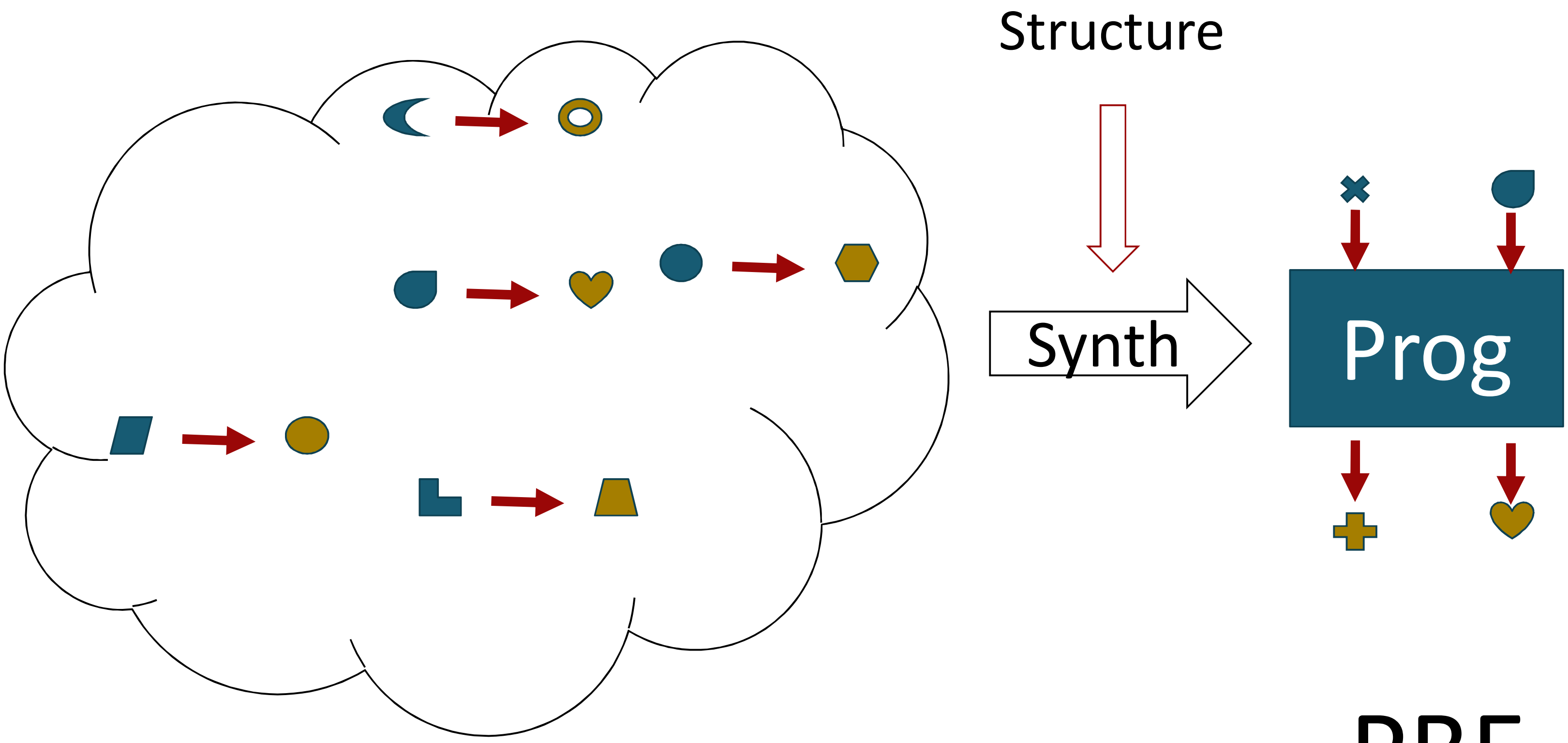


ML

LM

Learning

ML as Synthesis



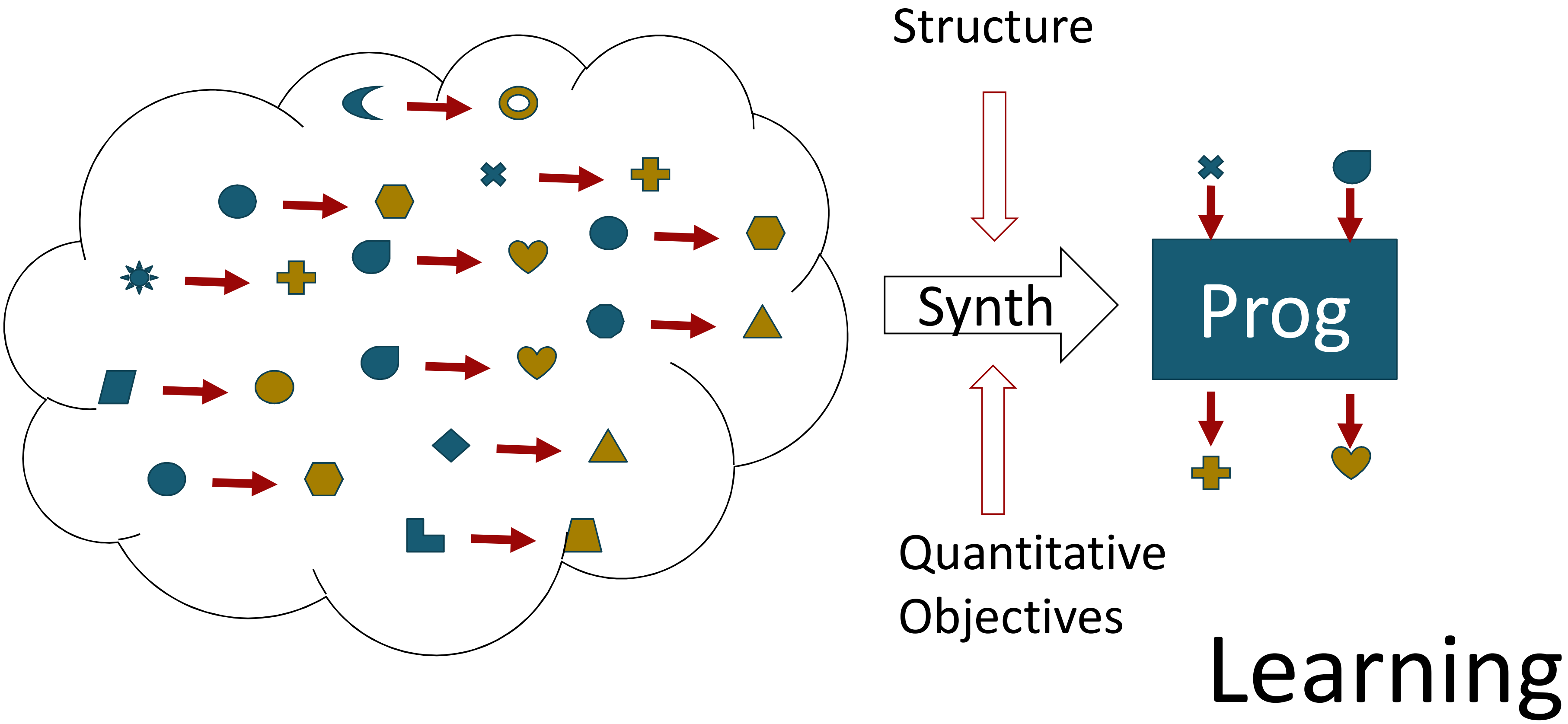
Structure

Synth

Prog

PBE

Synthesis as ML



Techniques

PL

- Formal reasoning
- Deduction
- Structure
- Modularity
- Abstraction
- Compositionality

$$\frac{\Gamma \vdash e_1 \qquad \Gamma \vdash e_2}{\Gamma \vdash e_1 + e_2} \qquad \frac{}{\Gamma \vdash x = e}$$

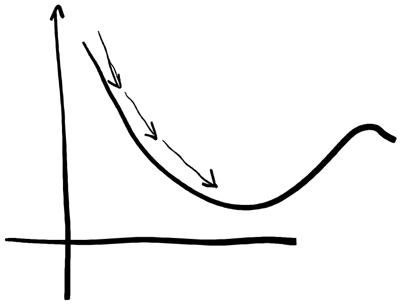
fold(_, _, _)

map(_, _, _)

cons(_,_)

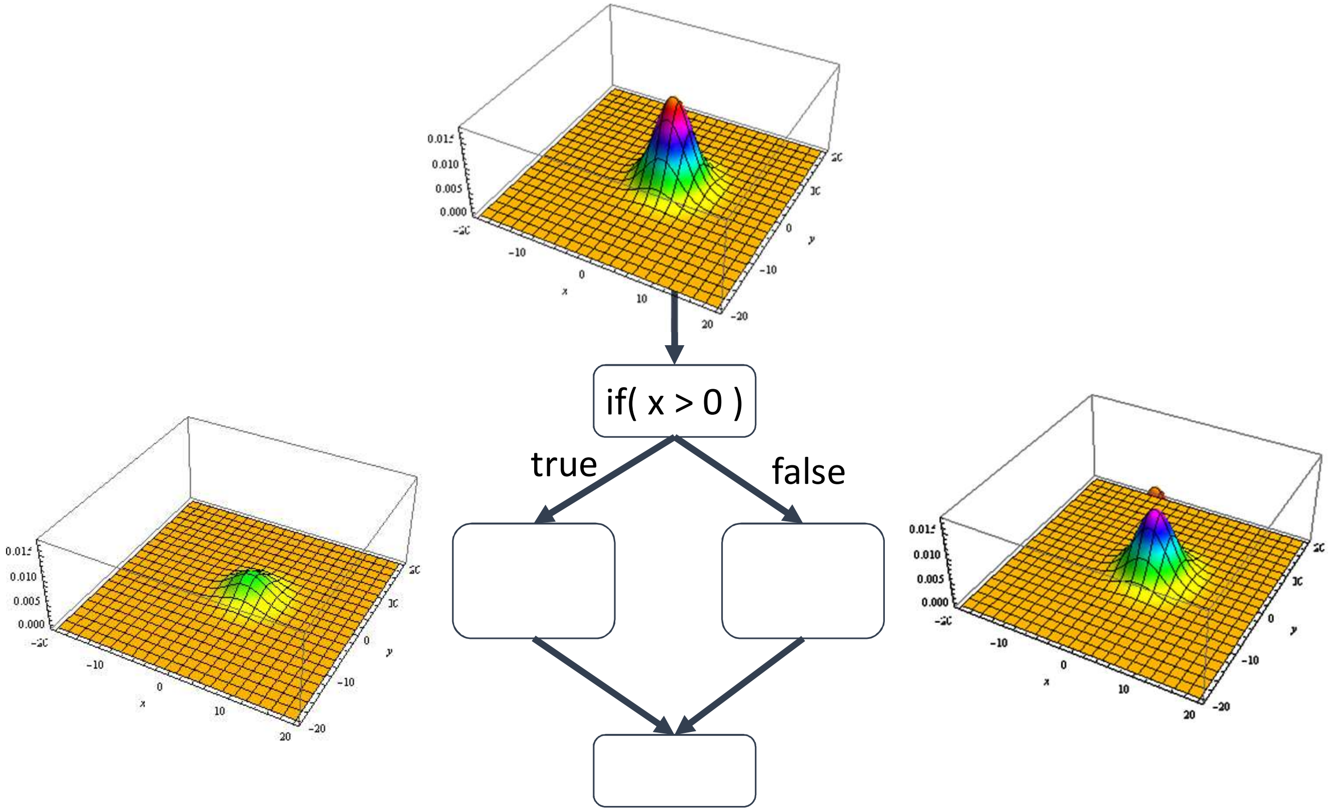
ML

- Optimization
- Probability



Smooth Interpretation

Swarat Chaudhuri and Armando Solar-Lezama, PLDI 2010



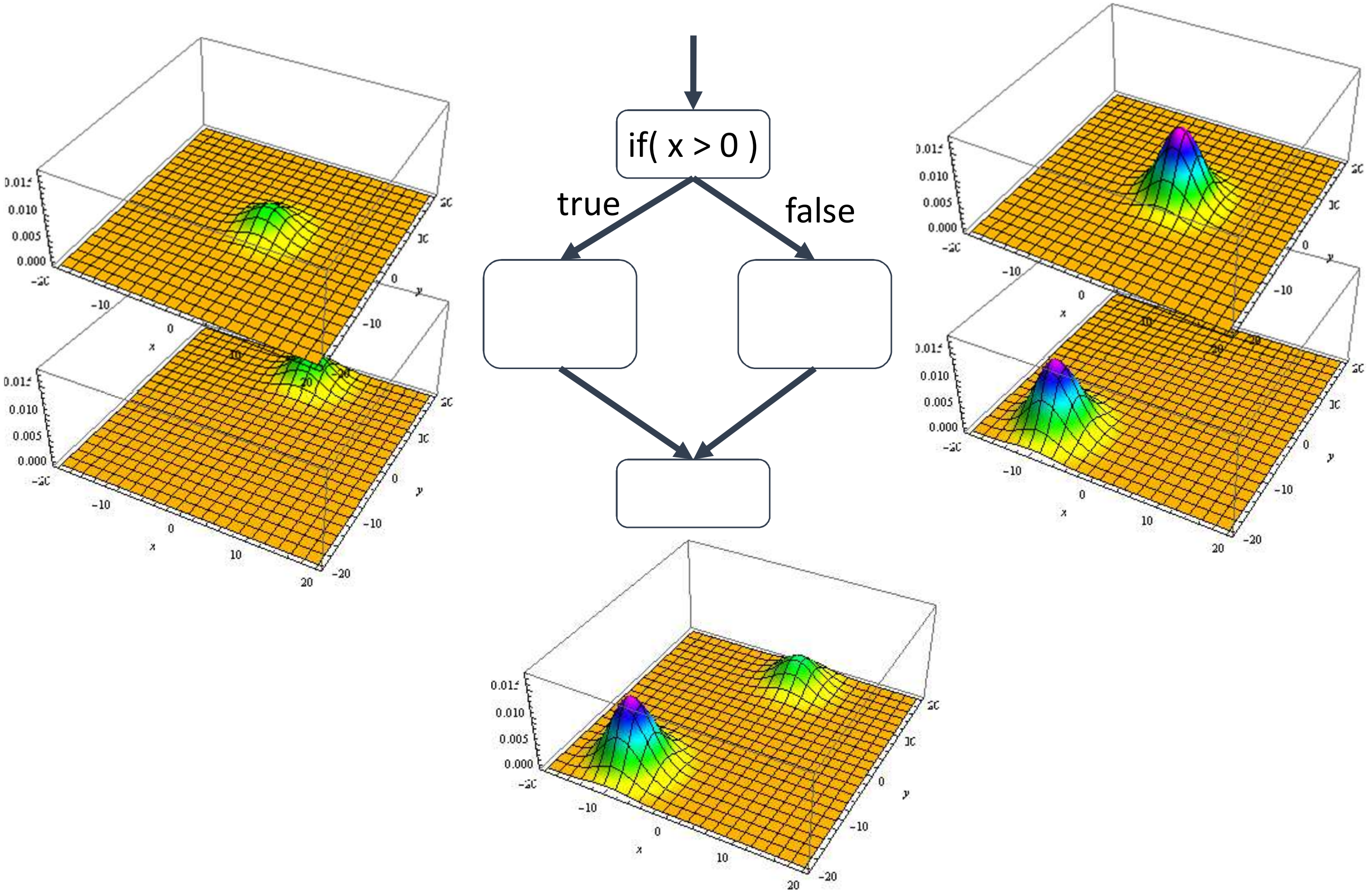
if(x > 0)

true

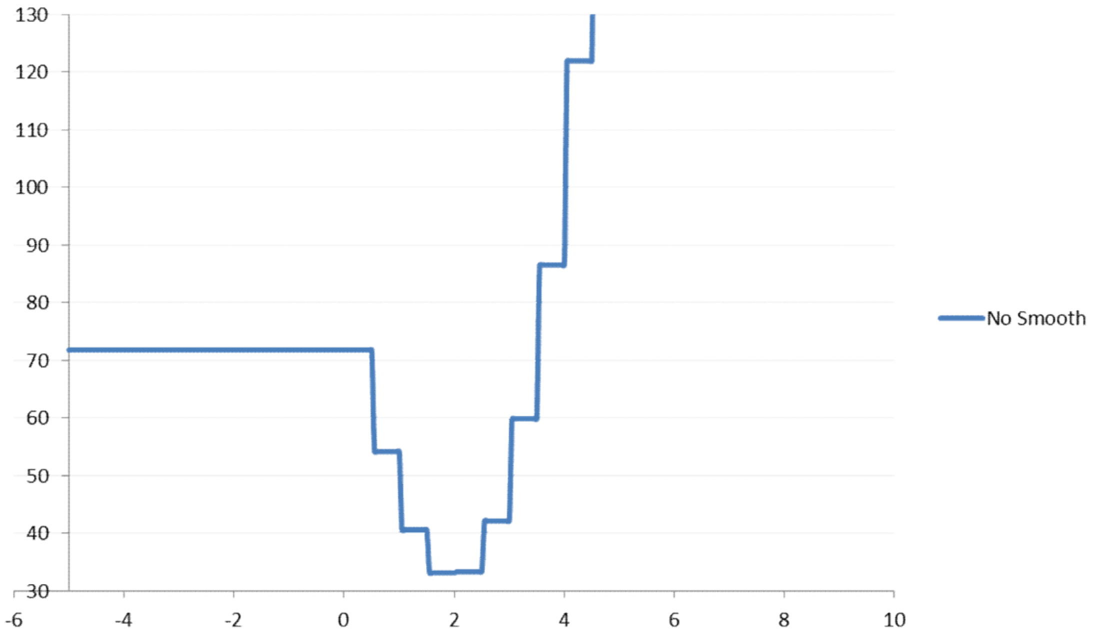
false

Smooth Interpretation

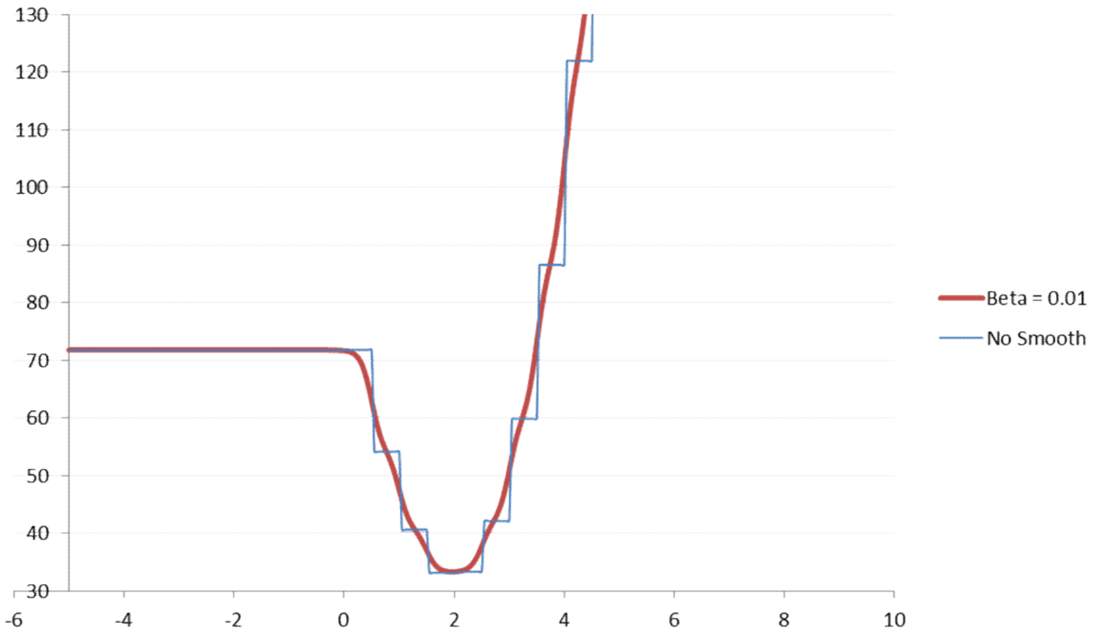
Swarat Chaudhuri and Armando Solar-Lezama, PLDI 2010



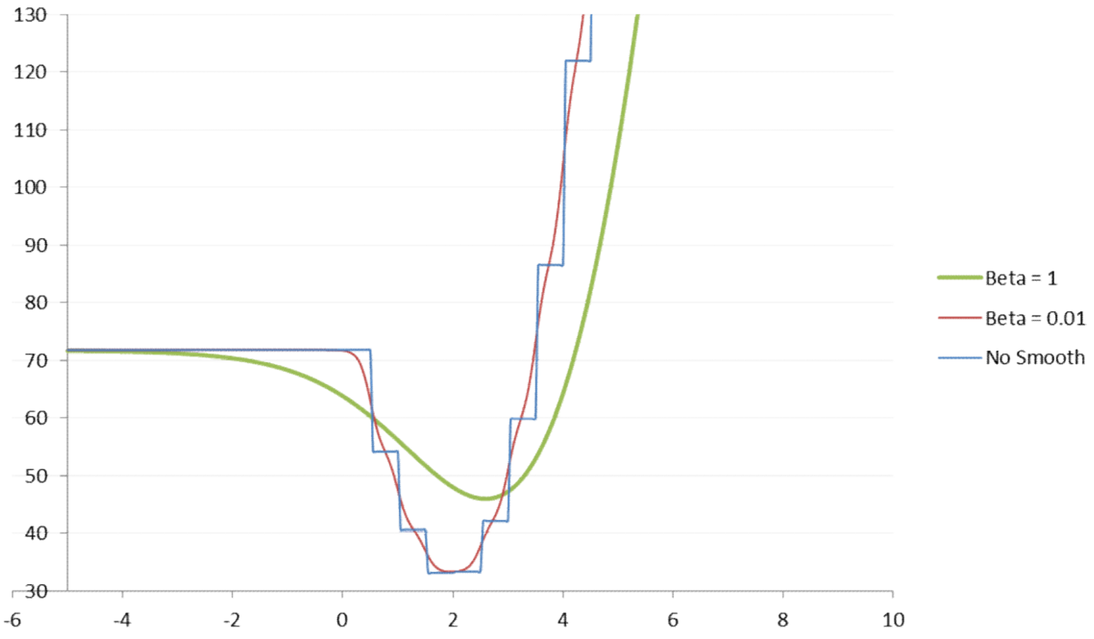
Error function



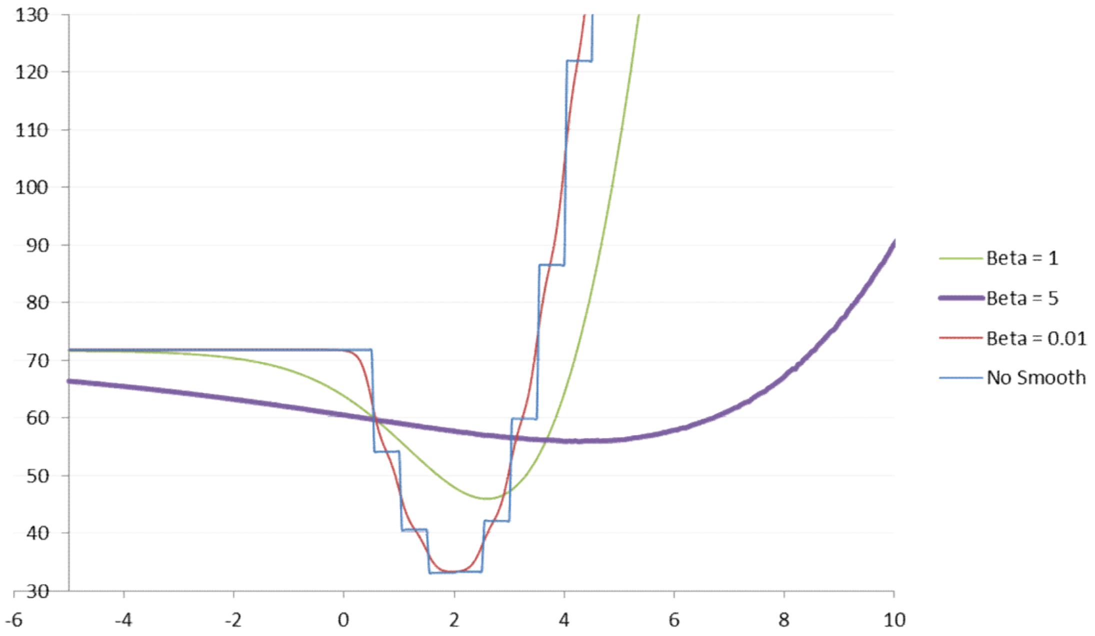
Error function



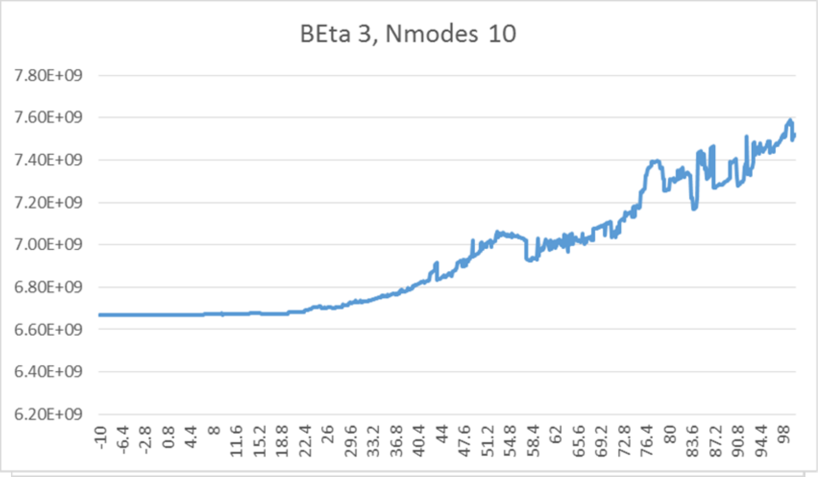
Error function



Error function

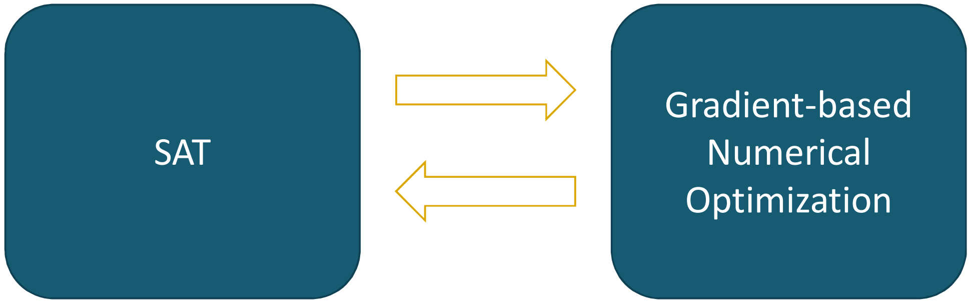


Excessive merging can cause problems



Synthesis with numerical optimization + SAT

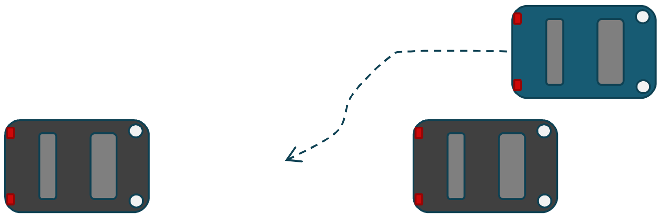
Jeevana Inala, Sean Gao, Soonho Kong, Armando Solar-Lezama arXiv 2017



SAT

Gradient-based Numerical Optimization

A Simple Example



A simple example

```
box obs1 = …  // stationary car 1
box obs2 = …  // stationary car 2
world wld = new world(n= 2, obstacles = {obs1, obs2});

// your car
car c = new car(…);

float dt = 0.1;

print(c);
for(int i=0; i<100; ++i){
    controller(c);
    simulate(c, dt);
    detectCollision(c, wld);
    checkCar(c);
    print(c);
}

assert reachGoal(c);
```

A simple example

```
void controller(car c) {
       float v = ??;
       float x1 = ??; float x2 = ??; float x3 = ??;
       checkSwitch(x1); checkSwitch(x2); checkSwitch(x3);
       if (c.b.xb > x1) {
              c.v = v;
              c.ang = 0.0;
       } else if (c.b.xb > x2) {
              c.v = v;
              c.ang = ??;
       } else if (c.b.xb > x3) {
              c.v = v;
              c.ang = ??;
       } else {
              c.v = 0.0;
       }
}
```

A simple example

```
void simulate(car c, float dt){
        float YL = c.b.yf - c.b.yb;
        float XL = c.b.xf - c.b.xb;
        float H = sqrt(YL*YL + XL*XL);
        float coa = (XL/H);
        float sia = (YL/H);
        float DY = c.v*dt*(sin(c.ang)*coa + cos(c.ang)*sia);
        float DX = c.v*dt*(cos(c.ang)*coa - sin(c.ang)*sia);

        c.b.xf += DX;
        c.b.yf += DY;
        float tt = (DX + XL)* coa + (DY + YL)*sia;
        float q = tt
             - 0.5*sqrt(4.0*(tt*tt) - 4.0*(DX*DX + 2.0*DX*XL + DY*(DY + 2.0*YL)) );
        c.b.xb += q*coa;
        c.b.yb += q*sia;
}
```

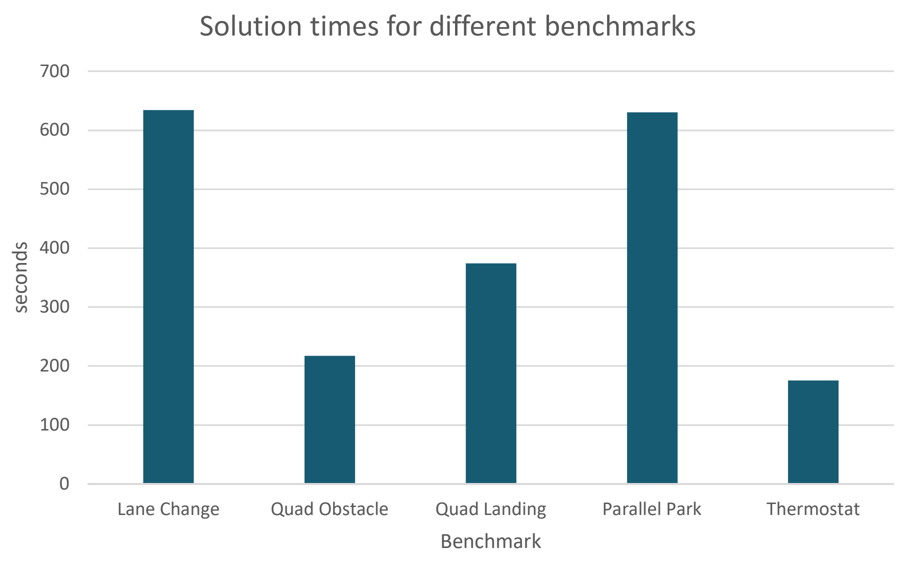
A simple example

```
void detectCollision(car c, world w){
    for (int i = 0; i < w.n; i++) {
       box o = w.obstacles[i];
       detectCollisionWithObject(c.b, o);
    }
}

void detectCollisionWithObject(box o1, box o2) {
    // make sure that vertices of o1 are not inside o2
    float[8] vertices = getVertices(o1);
    for (int i = 0; i < 4; i++) {
       assert(!isInside(vertices[2*i], vertices[2*i+1], o2));
    }
    // make sure that vertices of o2 are not inside o1
    vertices = getVertices(o2);
    for (int i = 0; i < 4; i++) {
       assert(!isInside(vertices[2*i], vertices[2*i+1], o1));
    }
}
```

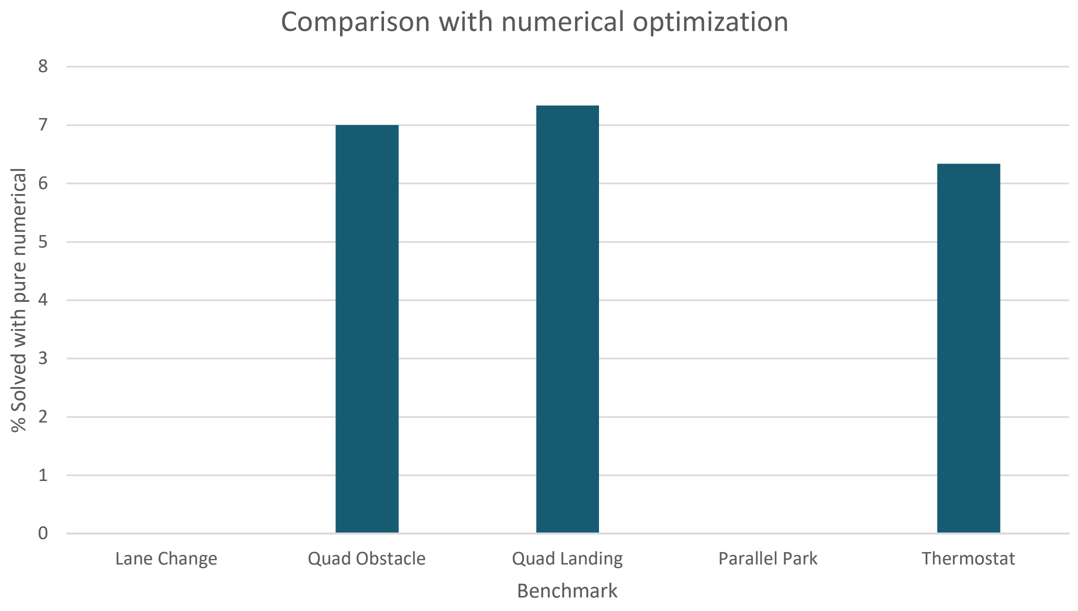
A simple example

Solver performance



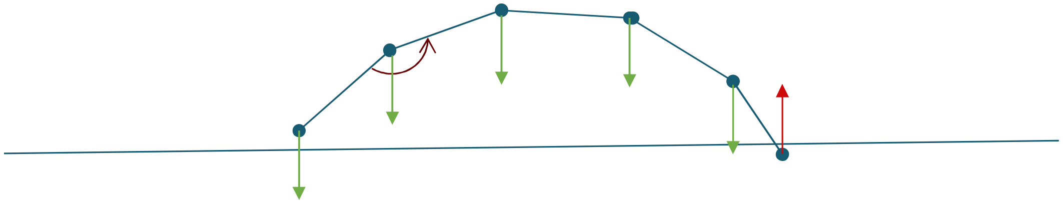
None of these benchmarks can be solved with previous SMT solvers!

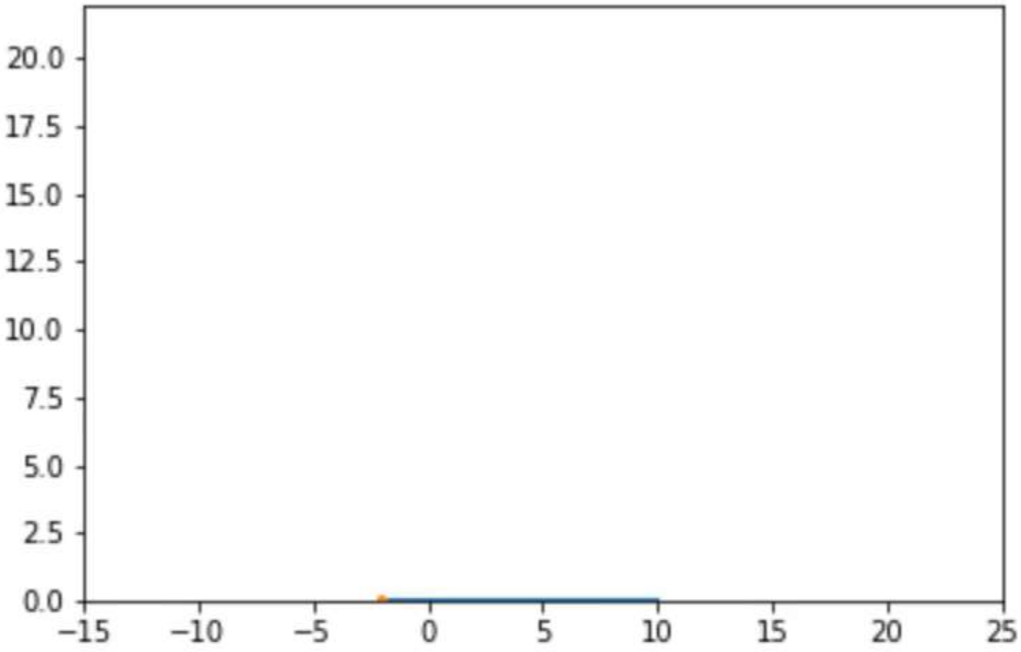
Solver performance



Another Example

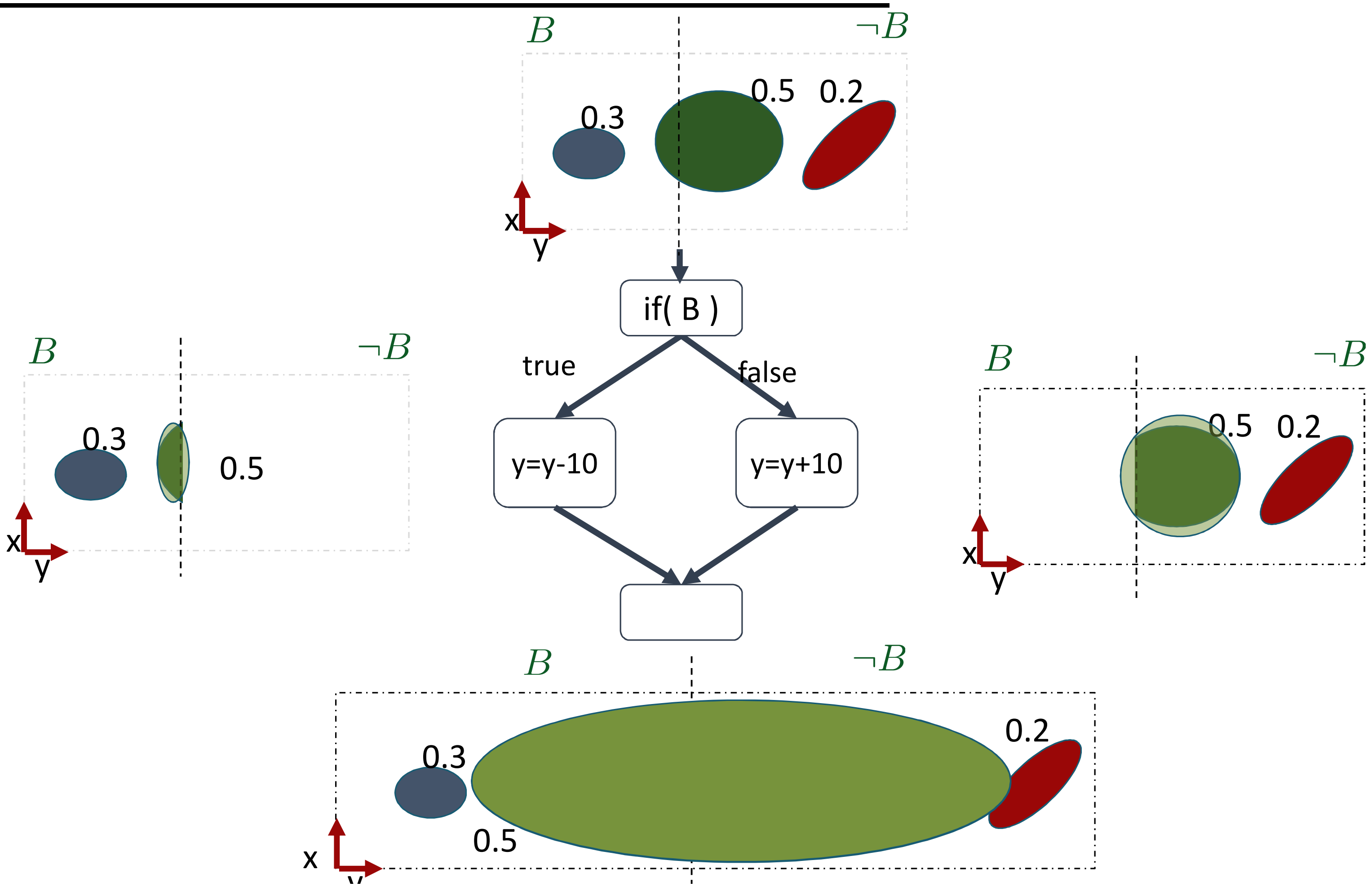
```
double[N] control([int N], Segment[N] segs, double time){
   double[N] deltas;
   double ii = 0.0;
   for(int i=0; i<N; ++i){
      ii += 1.0;
      repeat(5){
         if(time < ??*Dt()){
               deltas[i] = ?? + ii*?? + segs[i].ang*??;
   } }  }
   return deltas;
}
```




20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−15
−10
−5
0
5
10
15
20
25

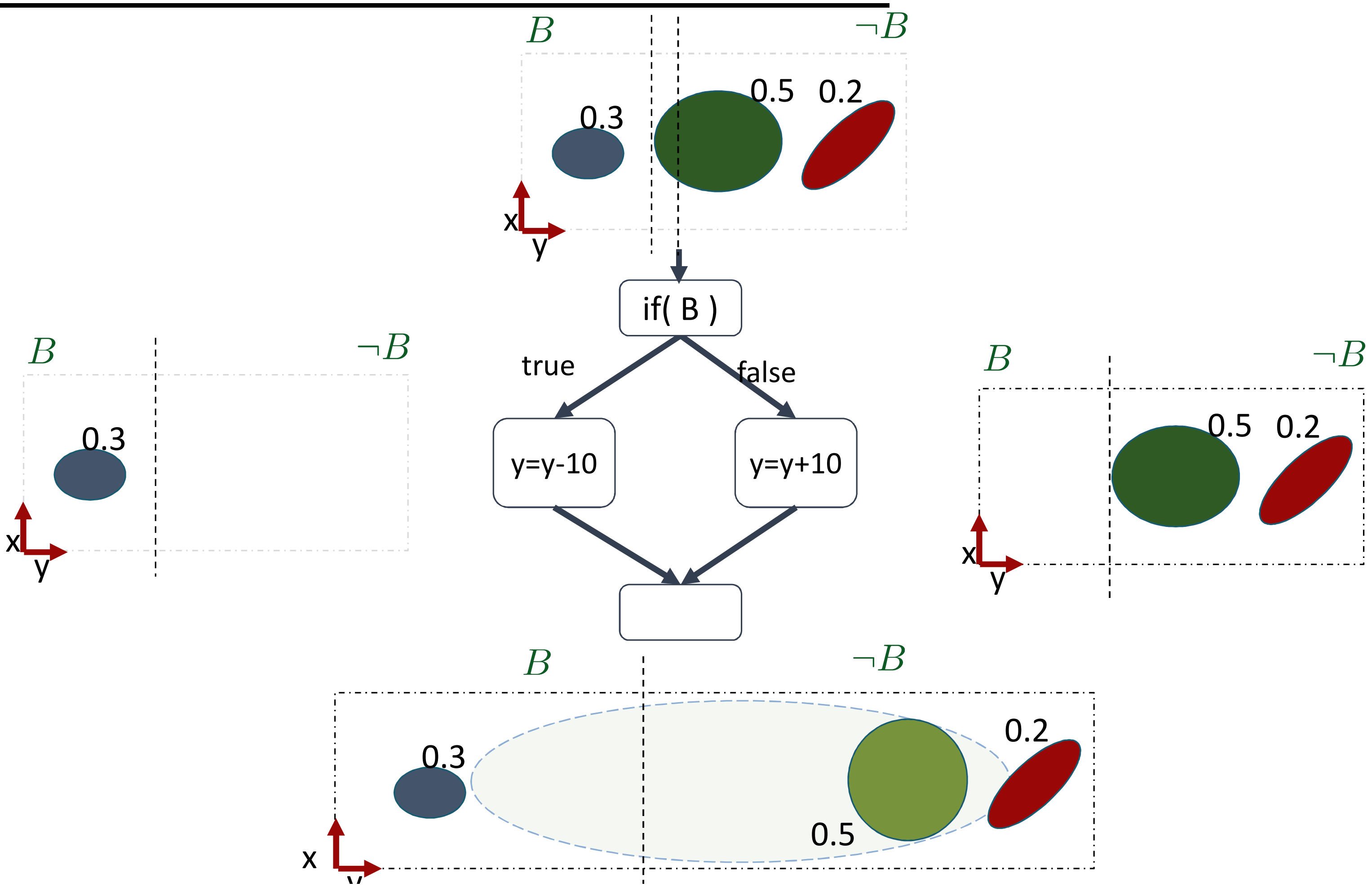
Smoothed Proof Search

Martin Clochard, Swarat Chaudhuri and Armando Solar-Lezama POPL 2014



Smoothed Proof Search

Martin Clochard, Swarat Chaudhuri and
Armando Solar-Lezama POPL 2014



if(B)

true

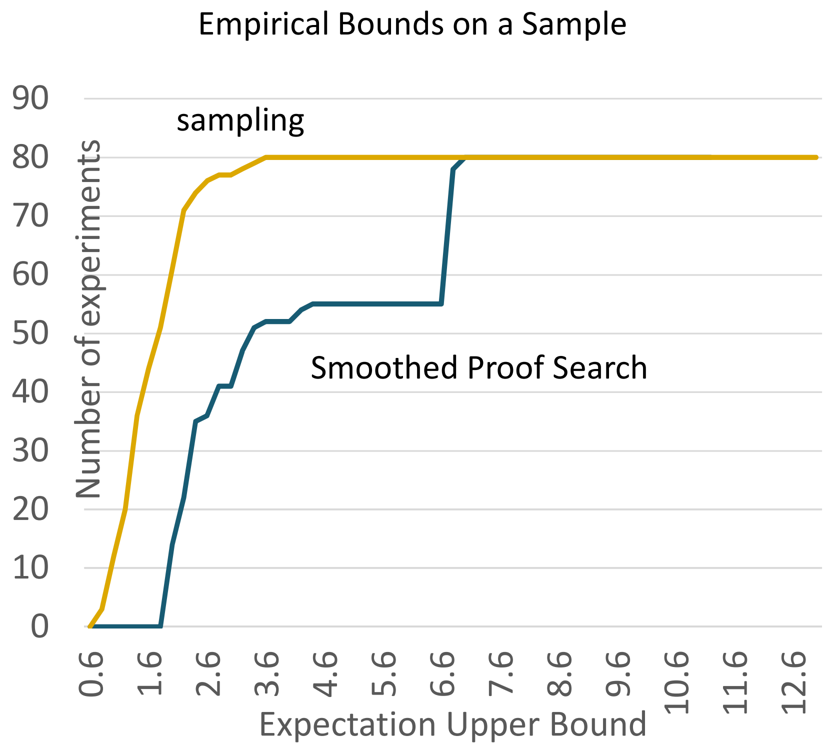
false

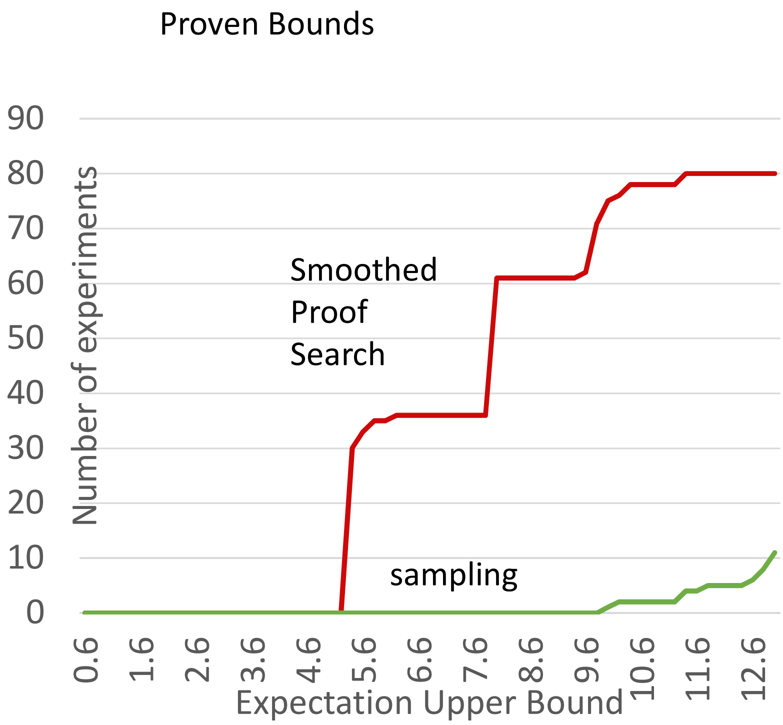
y=y-10

y=y+10

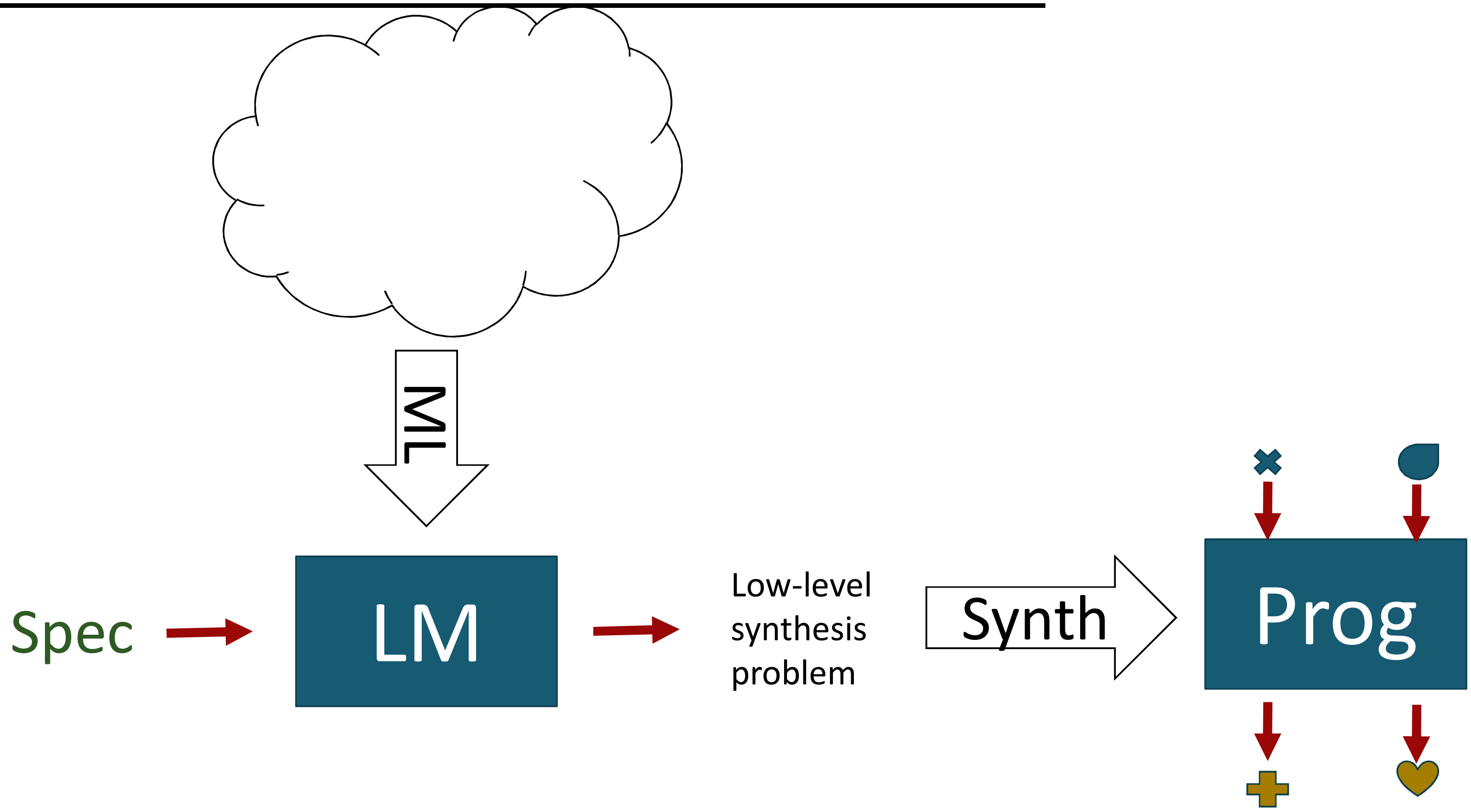
Smoothed Proof Search

Martin Clochard, Swarat Chaudhuri and Armando Solar-Lezama POPL 2014





Pipeline model



ML

Spec

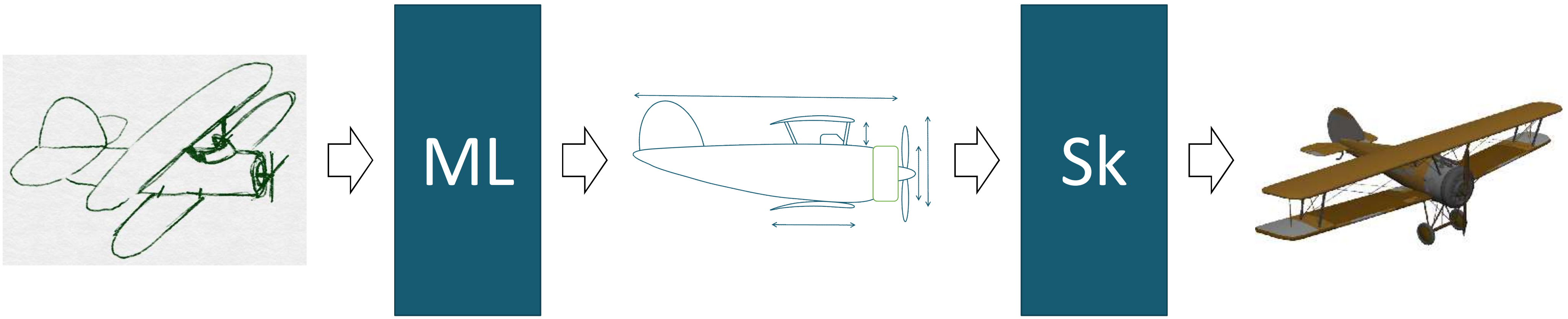
LM

Low-level synthesis problem

Synth

Prog

Pipeline model



High-level
unstructured spec

ML

Low-level
structured spec

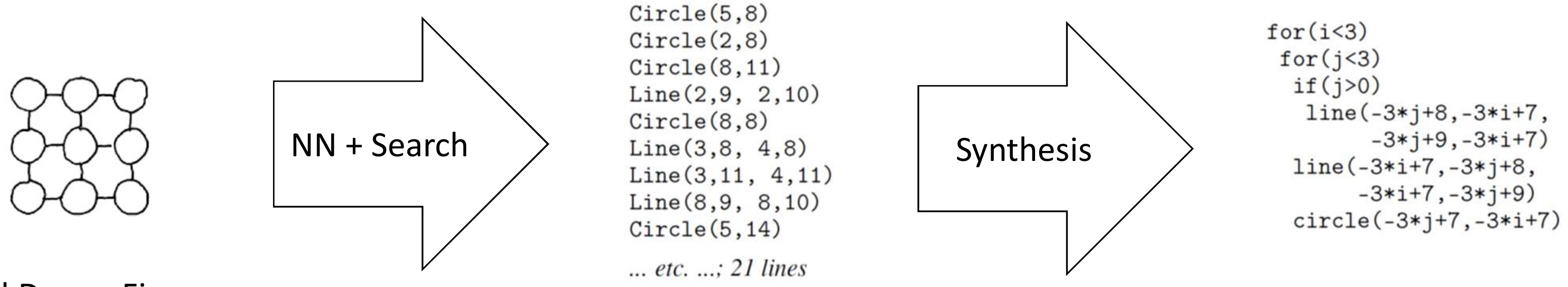
Sk

Complete
artifact

Learning to Infer Graphics Programs from Hand-Drawn Images

with Kevin Ellis, Daniel Ritchie, Josh Tenenbaum

From images to programs



NN + Search

Synthesis

```
Circle(5,8)
Circle(2,8)
Circle(8,11)
Line(2,9, 2,10)
Circle(8,8)
Line(3,8, 4,8)
Line(3,11, 4,11)
Line(8,9, 8,10)
Circle(5,14)
```

... etc. ...; 21 lines

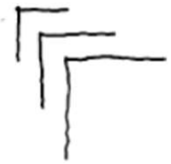
```
for(i<3)
 for(j<3)
  if(j>0)
   line(-3*j+8,-3*i+7,
        -3*j+9,-3*i+7)
  line(-3*i+7,-3*j+8,
       -3*i+7,-3*j+9)
  circle(-3*j+7,-3*i+7)
```

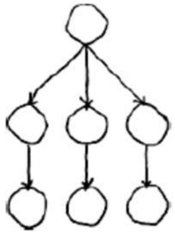
Hand Drawn Figure

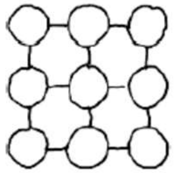
Description of
elements in the drawing

Program representation
of drawing

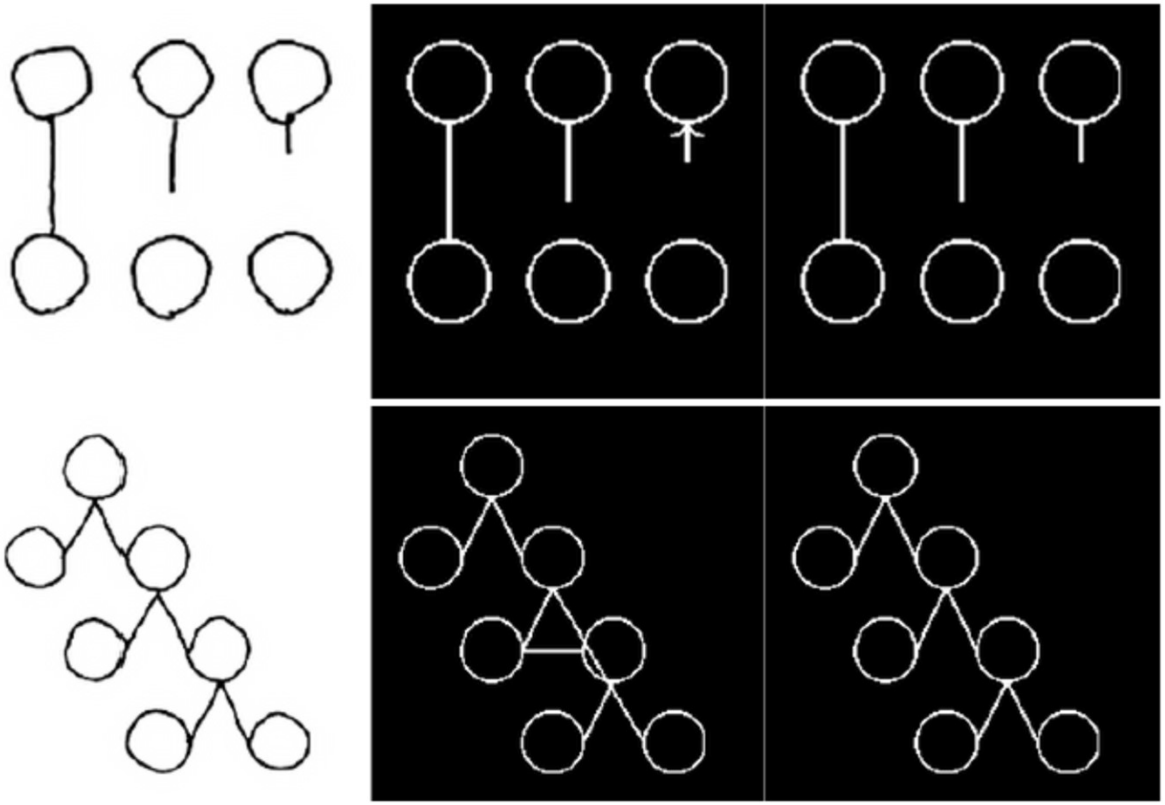
From images to programs



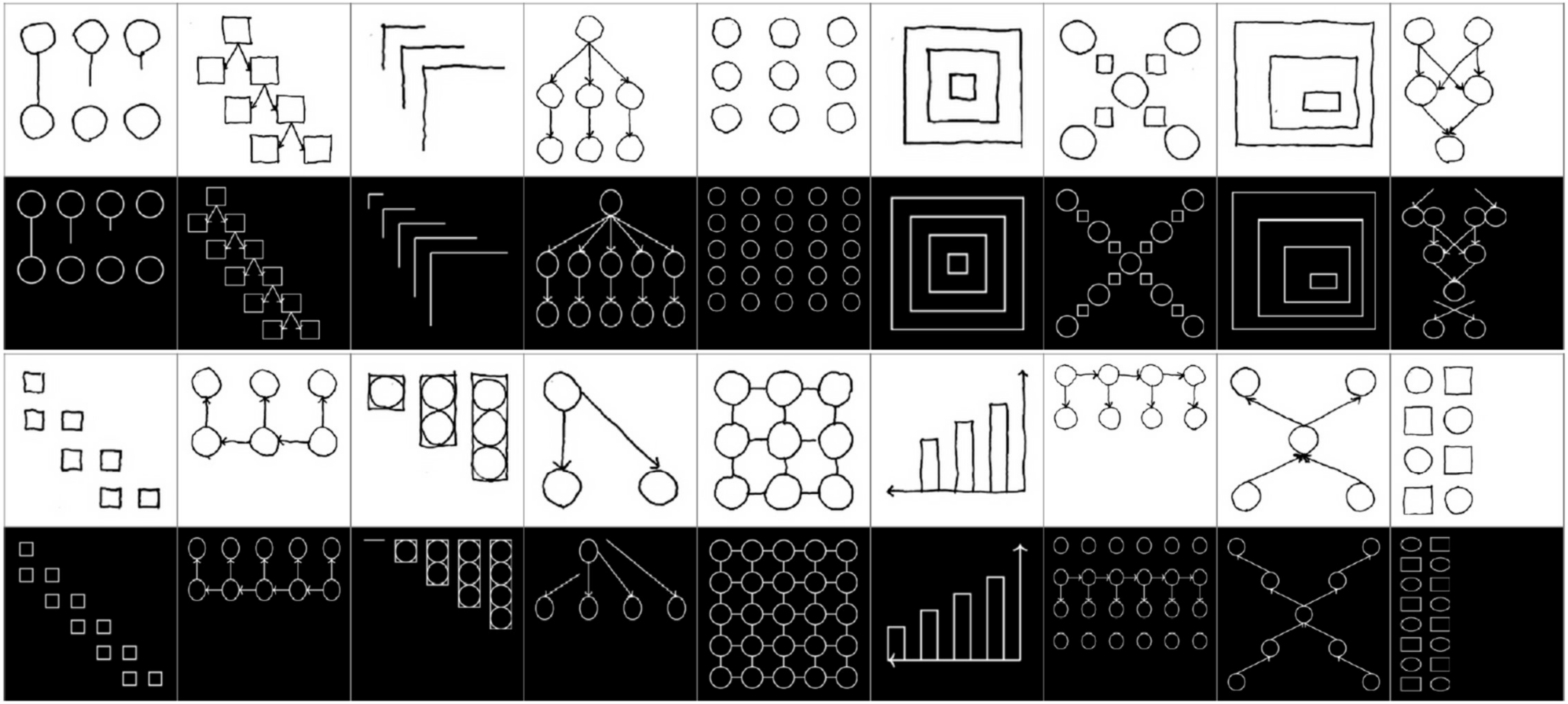




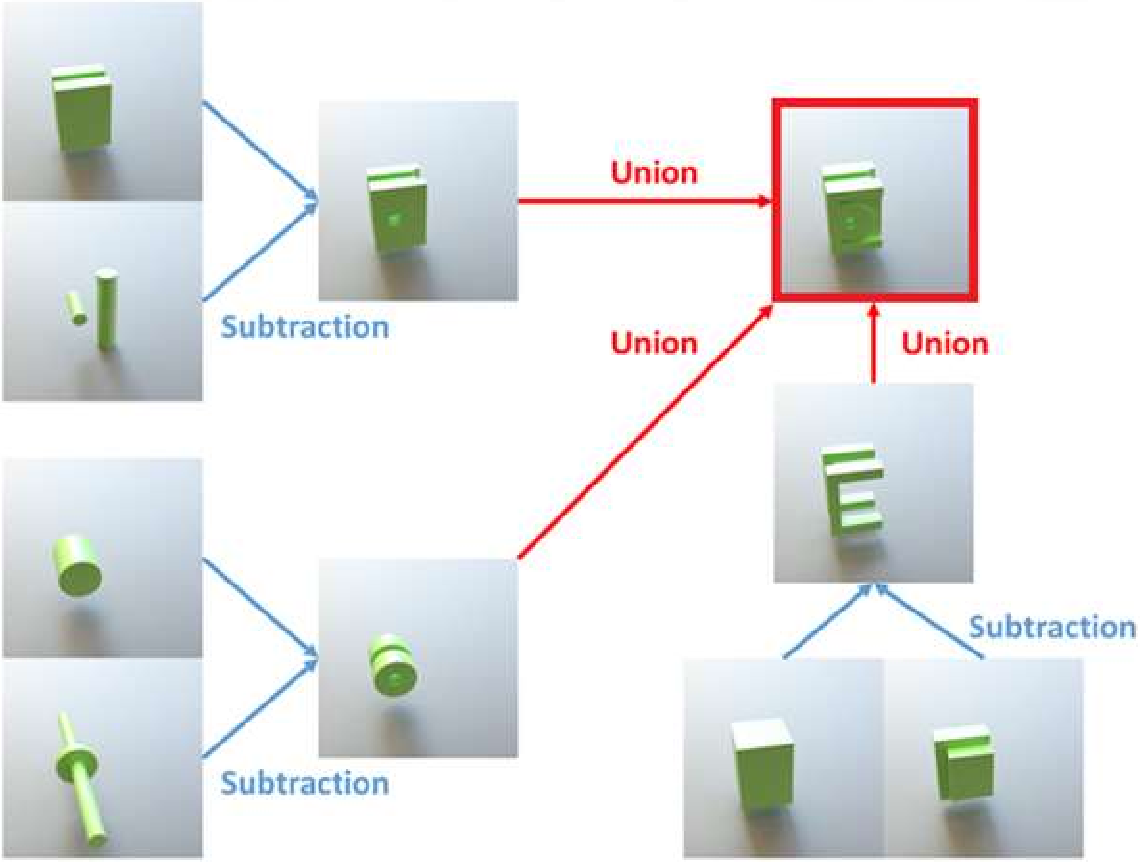
Why? Correcting errors in perception

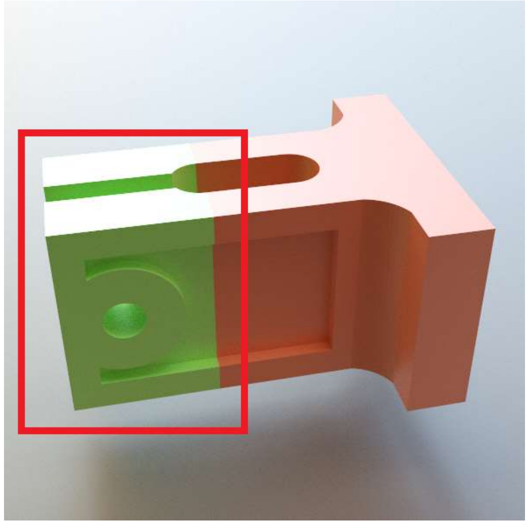


Why? Extrapolation



Synthesis for CAD

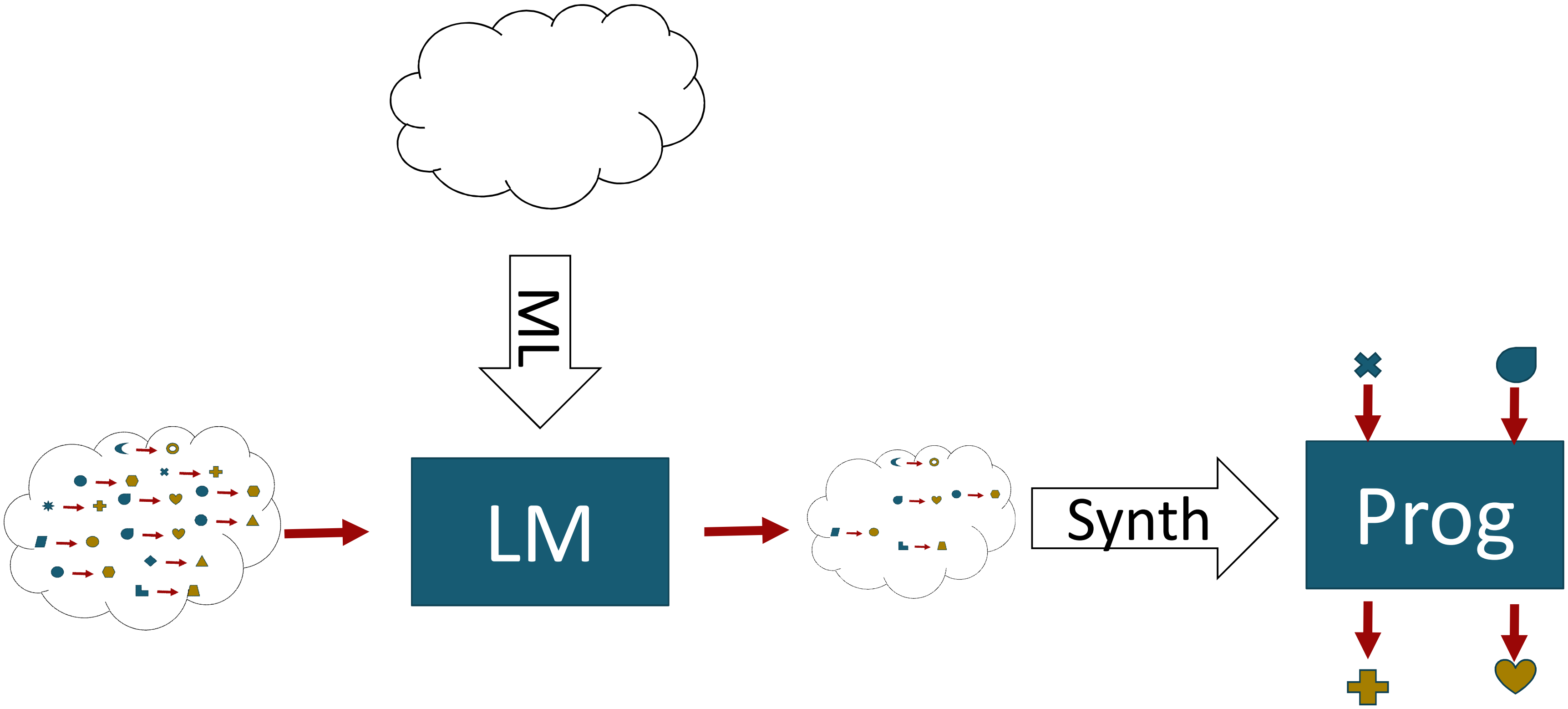




Tao Du, Adriana Schultz, Evan Pu, Jeevana Inala, Wojciech Matusik Armando Solar-Lezama (Submitted 2018)

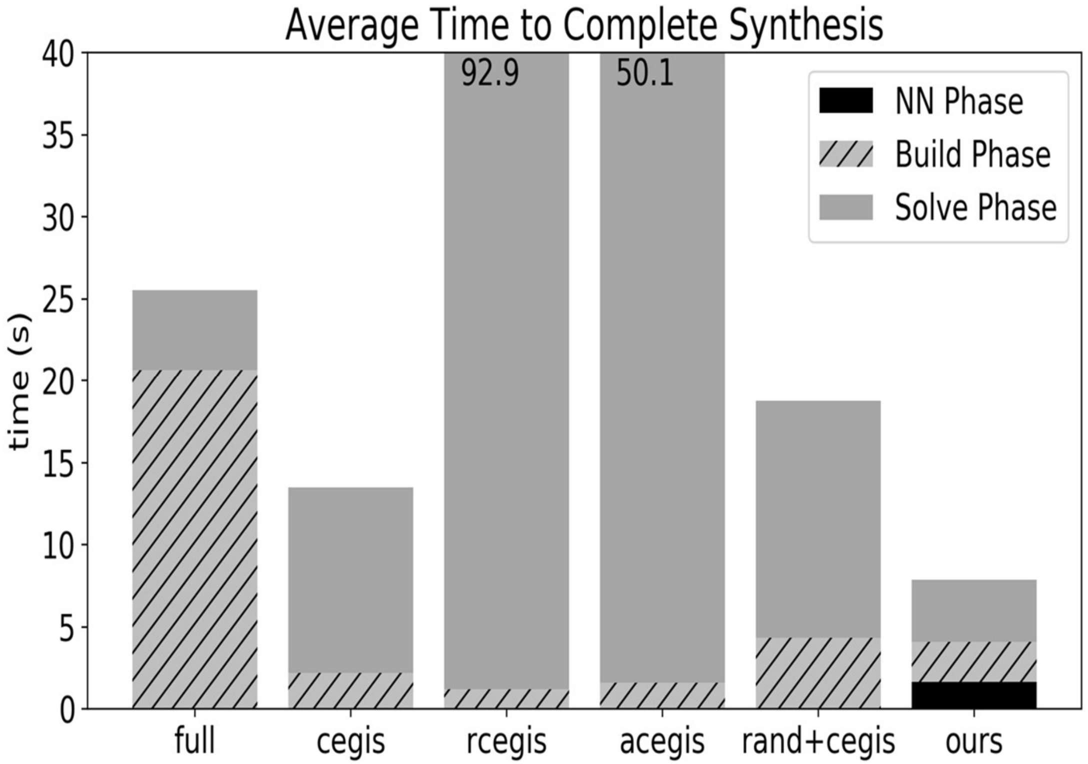
Selecting Representative Examples for Synthesis

Evan Pu, Zacherey Miranda, Leslie Kaelbling, Armando Solar-Lezama, 2016



Selecting Representative Examples for Synthesis

Evan Pu, Zacherey Miranda, Leslie Kaelbling, Armando Solar-Lezama, 2016



full: add all examples
cegis: add “first” example
rcegis: add random example
acegis: add arbitrary example
rand+cegis: instantiate rcegis with a random subset of examples
ours: instantiate rcegis with subset of examples chosen by neural network

Learning a DSL

Kevin Ellis, Josh Tenenbaum, Lucas E. Morales
in submission.

Domain Specific Language

Prior on the space of programs in the DSL

$\mathcal{D}$ θ ⇐ Can we learn these?

Requirement → $X = \{x_i\}$

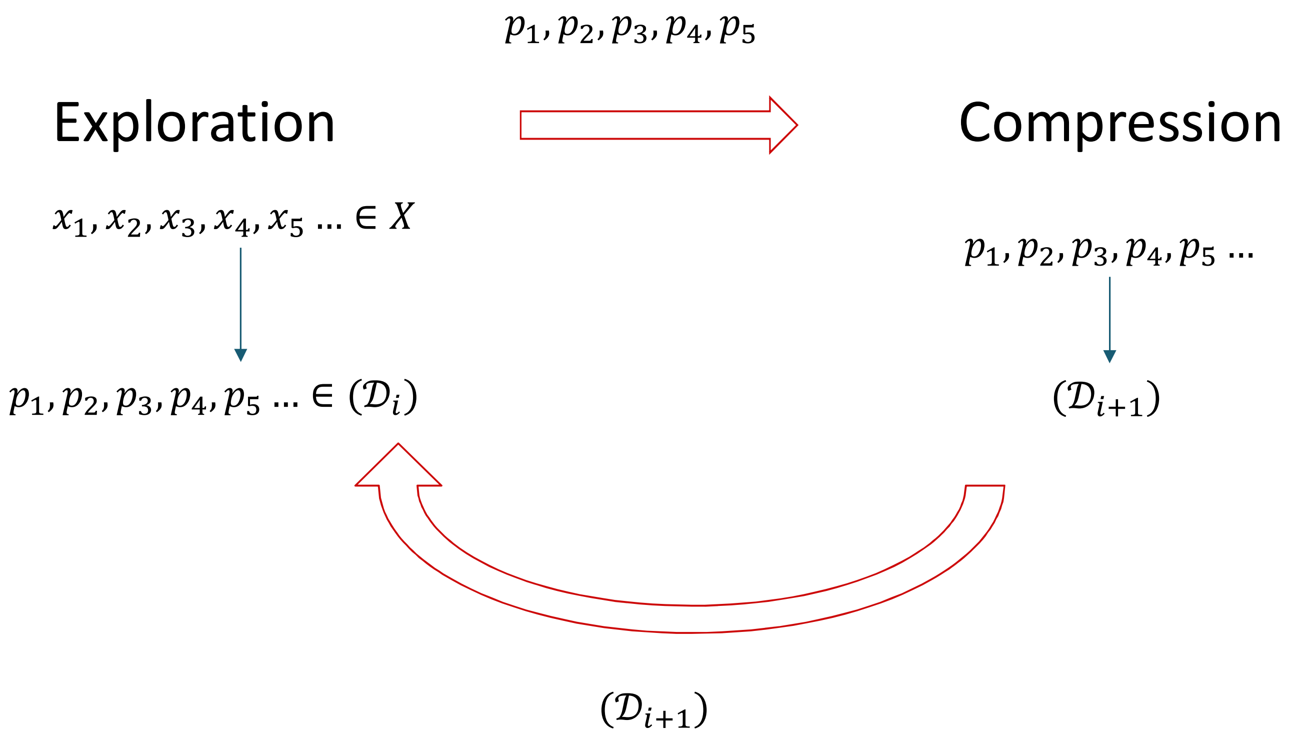
⇓

p ← Most likely program

$$\prod_{x \in X} \sum_{p} \mathbb{P}[x|p]\mathbb{P}[p|\mathcal{D},\theta]$$

E-C Algorithm

Dechter, Malmaud, Adams, Tenenbaum:
Bootstrap Learning via Modular Concept Discovery. IJCAI 2013



p_1, p_2, p_3, p_4, p_5

Exploration

$x_1, x_2, x_3, x_4, x_5 \dots \in X$

$p_1, p_2, p_3, p_4, p_5 \dots \in (\mathcal{D}_i)$

Compression

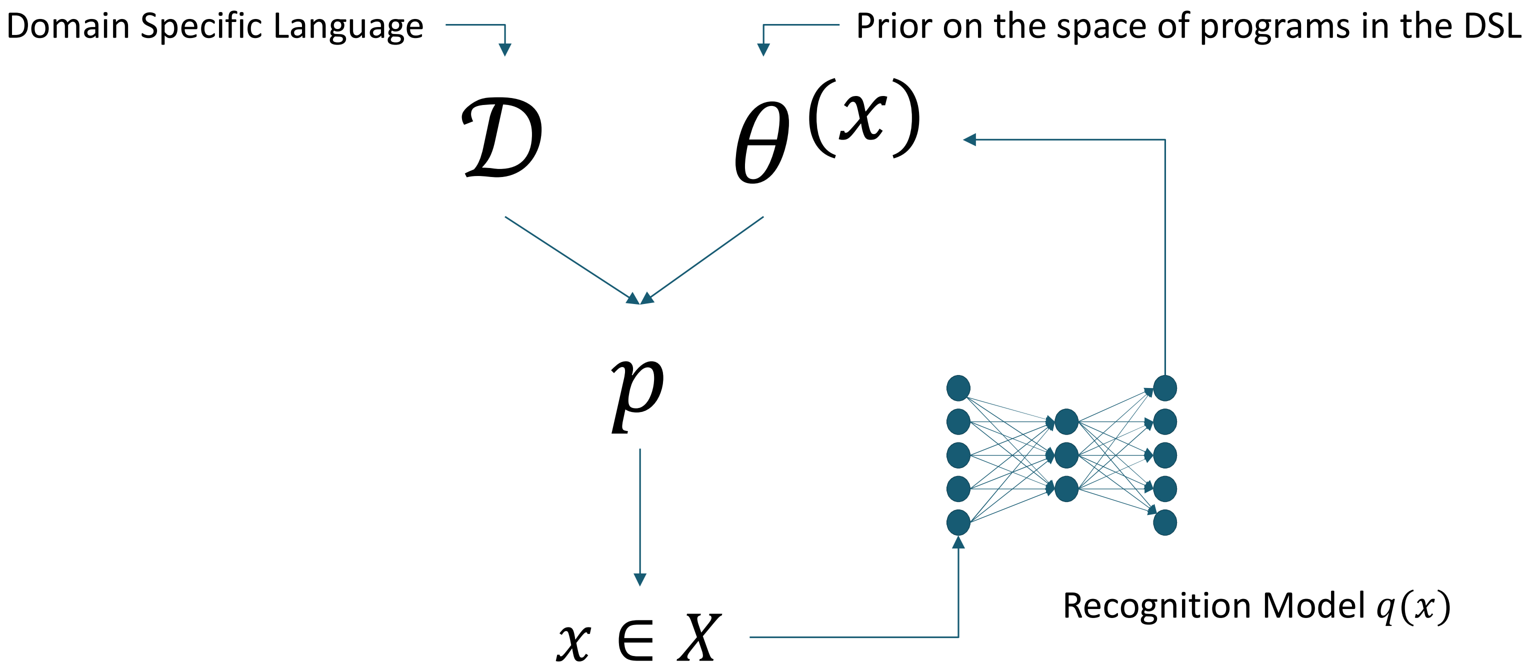
$p_1, p_2, p_3, p_4, p_5 \dots$

$(\mathcal{D}_{i+1})$

$(\mathcal{D}_{i+1})$

Learning a Recognition model

Deepcoder (Balog et.al. 2017)



Domain Specific Language

Prior on the space of programs in the DSL

$\mathcal{D}$

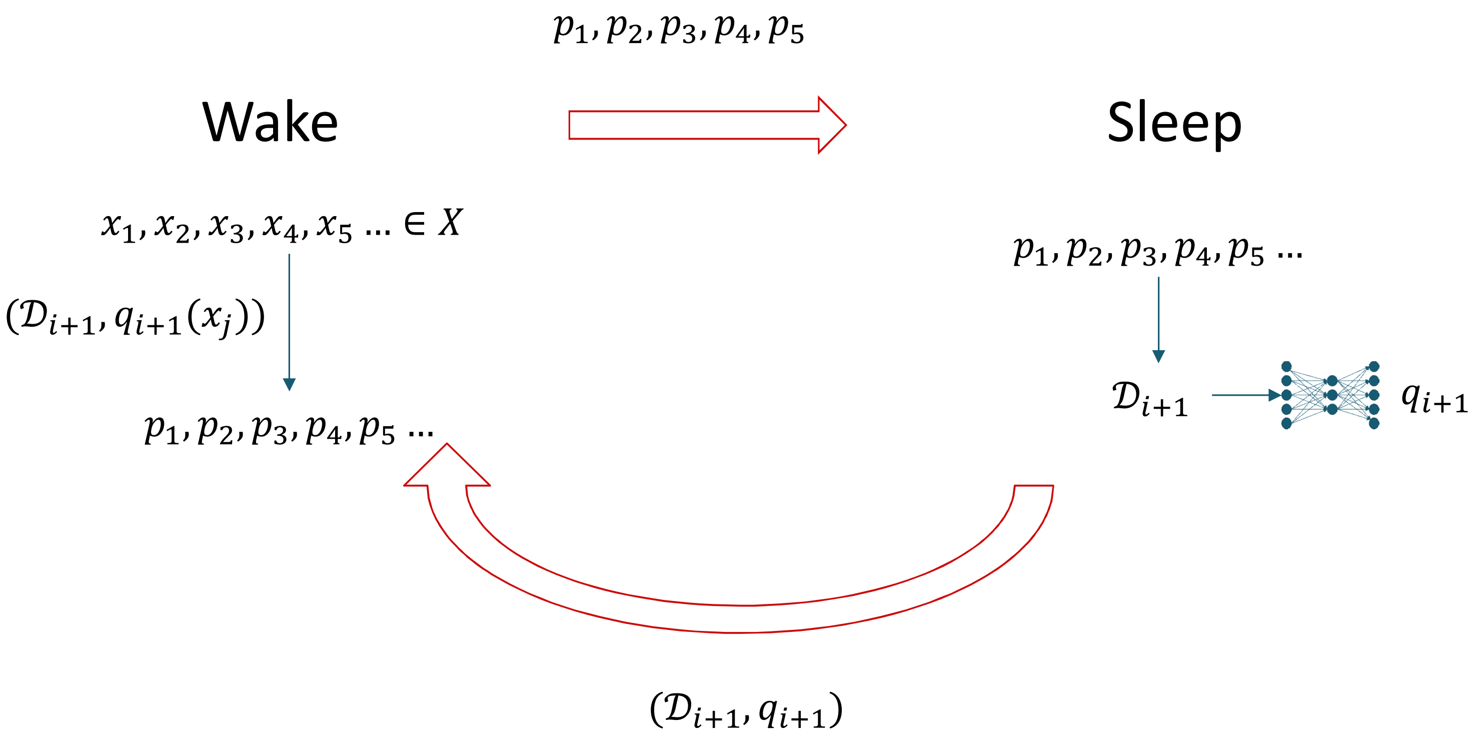
$\theta^{(x)}$

p

$x \in X$

Recognition Model $q(x)$

New DreamCoder Algorithm



p_1, p_2, p_3, p_4, p_5

Wake

Sleep

$x_1, x_2, x_3, x_4, x_5 \ldots \in X$

$(\mathcal{D}_{i+1}, q_{i+1}(x_j))$

$p_1, p_2, p_3, p_4, p_5 \ldots$

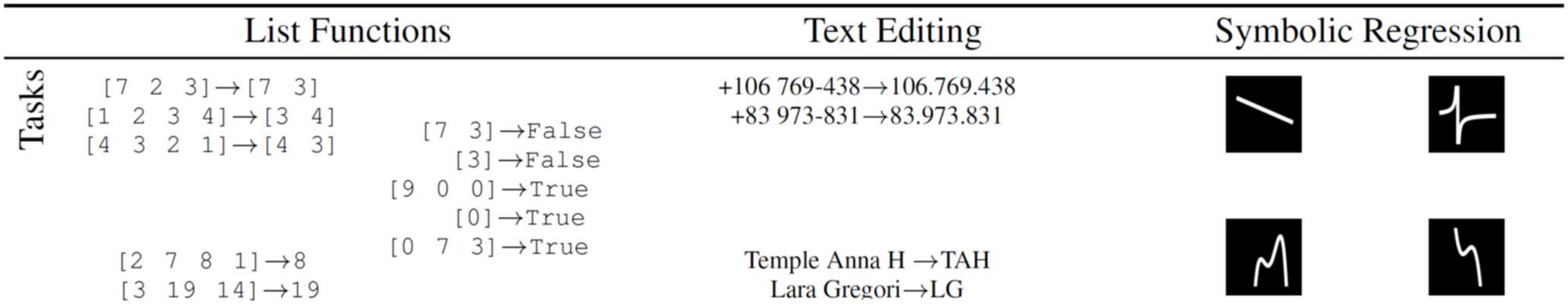
$p_1, p_2, p_3, p_4, p_5 \ldots$

$\mathcal{D}_{i+1}$

q_{i+1}

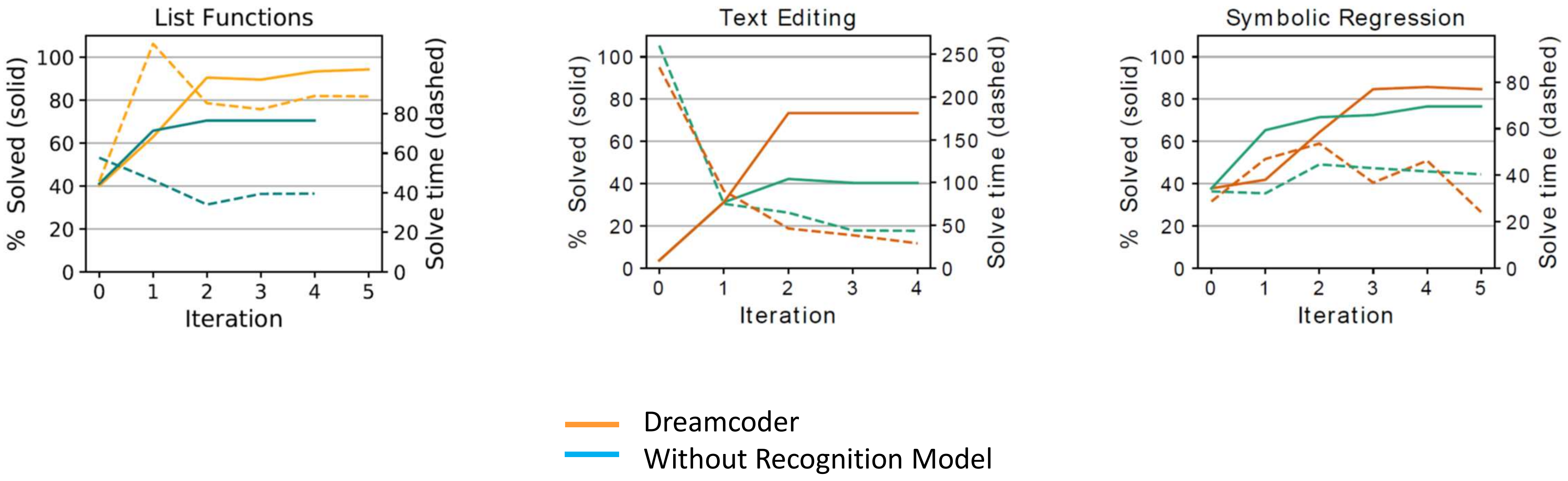
$(\mathcal{D}_{i+1}, q_{i+1})$

Results

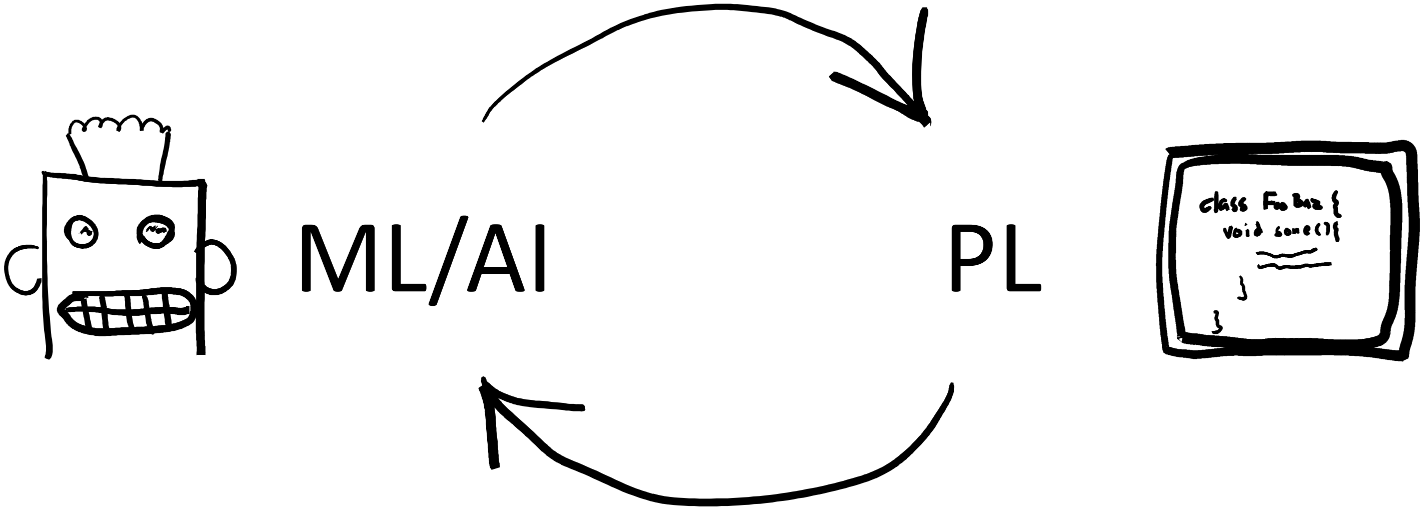


	List Functions		Text Editing	Symbolic Regression
Tasks	[7 2 3]→[7 3] [1 2 3 4]→[3 4] [4 3 2 1]→[4 3]	[7 3]→False [3]→False [9 0 0]→True [0]→True [0 7 3]→True	+106 769-438→106.769.438 +83 973-831→83.973.831	
	[2 7 8 1]→8 [3 19 14]→19		Temple Anna H →TAH Lara Gregori→LG	

Results


List Functions
% Solved (solid)
100
80
60
40
20
0
Solve time (dashed)
80
60
40
20
0
0
1
2
3
4
5
Iteration
Text Editing
% Solved (solid)
100
80
60
40
20
0
Solve time (dashed)
250
200
150
100
50
0
0
1
2
3
4
Iteration
Symbolic Regression
% Solved (solid)
100
80
60
40
20
0
Solve time (dashed)
80
60
40
20
0
0
1
2
3
4
5
Iteration
Dreamcoder
Without Recognition Model

Conclusion



ML/AI

PL