

Open-ended generation of diverse solvable coding problems with autotelic generative models

Pierre-Yves Oudeyer

Inria

Flowers Lab

Inria, Univ. Bordeaux, Ensta ParisTech













































<http://www.pyoudeyer.com>

<https://flowers.inria.fr>

Twitter: @pyoudeyer

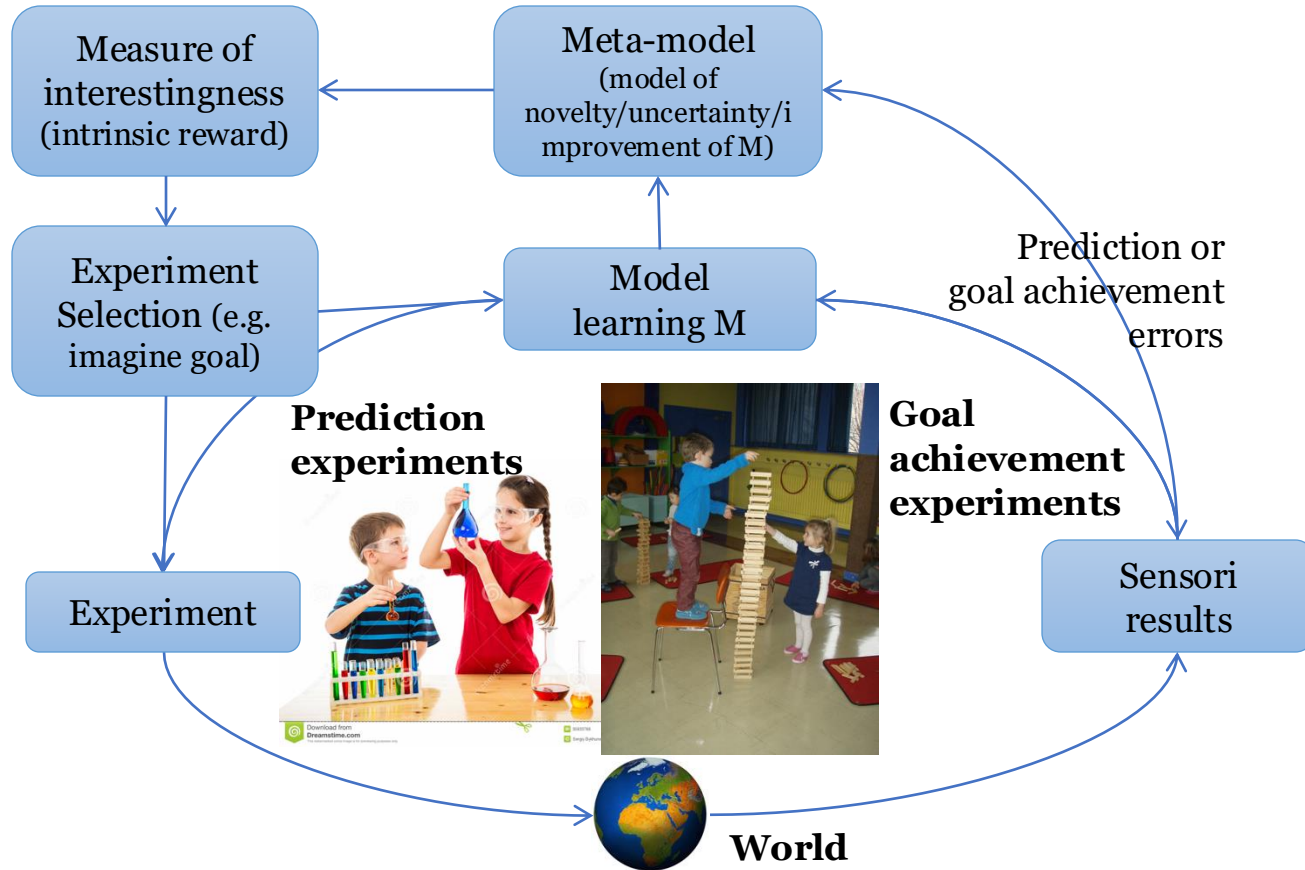


Open-ended and autonomous development in individuals: curiosity and language

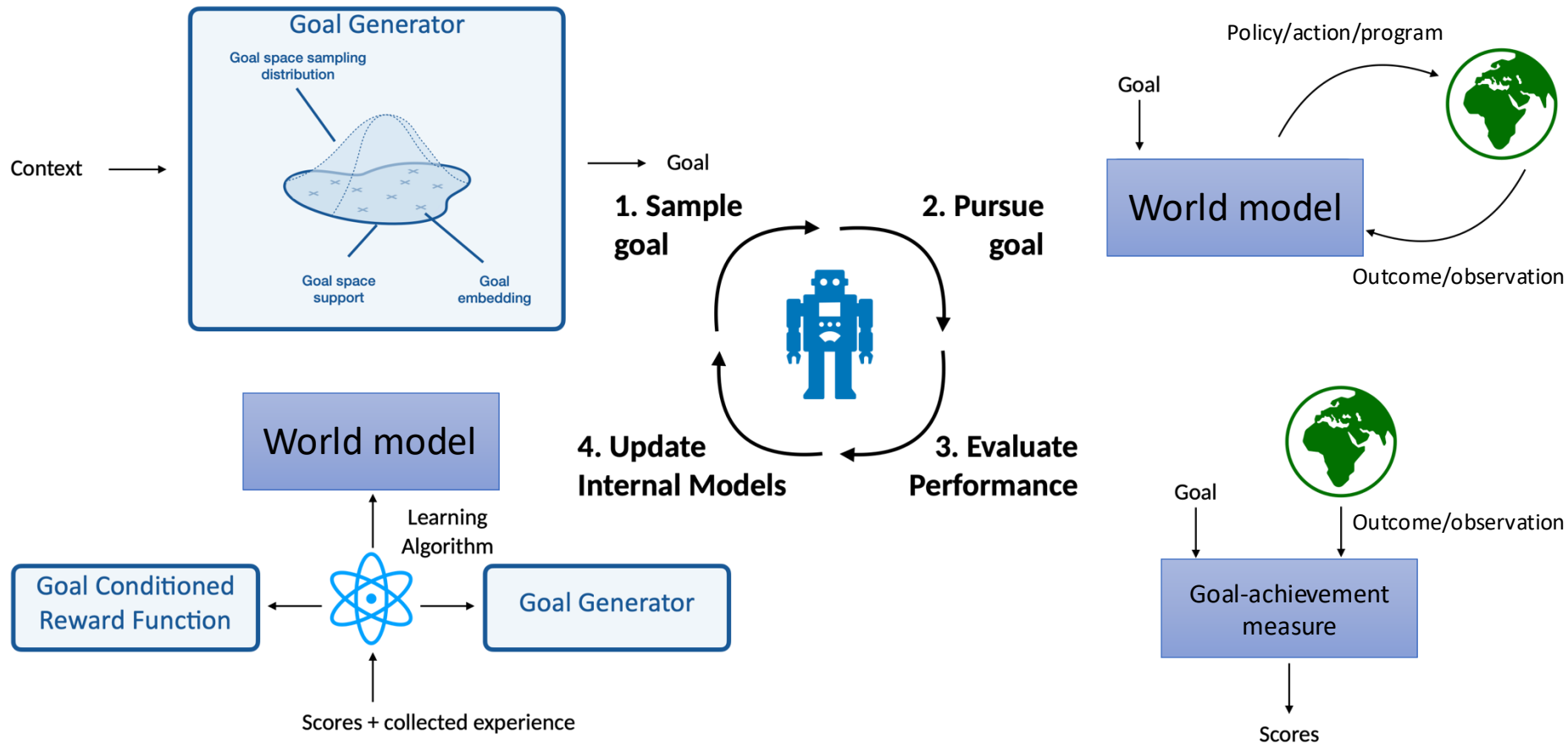
PHYSICAL DEVELOPMENT	Average age skills begin	3 months	6 months	9 months	1 year	2 years	3 years	5 years		
Head and trunk control	 lifts head part way up	 holds head up briefly	 holds head up high and well	 holds up head and shoulders	 turns head and shifts weight	 holds head up well when lifted	 moves and holds head easily in all directions			
Rolling		 rolls belly to back	 rolls back to belly	 rolls over and over easily in play						
Sitting		 sits only with full support	 sits with some support	 sits with hand support	 begins to sit without support	 sits well without support	 twists and moves easily while sitting			
Crawling and walking		 begins to creep	 scoots or crawls	 pulls to standing	 takes steps	 walks	 runs	 can walk on tiptoe and on heels	 walks easily backward	 hops on one foot
Arm and hand control	 grips finger put into hand	 begins to reach towards objects	 reaches and grasps with whole hand	 passes object from one hand to other	 grasps with thumb and forefinger	 easily moves fingers back and forth from nose to moving object	 throws and catches ball			
Seeing	 follows close object with eyes	 enjoys bright colors/shapes	 recognizes different faces	 eyes focus on far object	 looks at small things/pictures	 Sees small shapes clearly at 6 meters (see p. 453 for test).				
Hearing	 moves or cries at a loud noise	 turns head to sounds	 responds to mother's voice	 enjoys rhythmic music	 understands simple words	 hears clearly and understands most simple language				

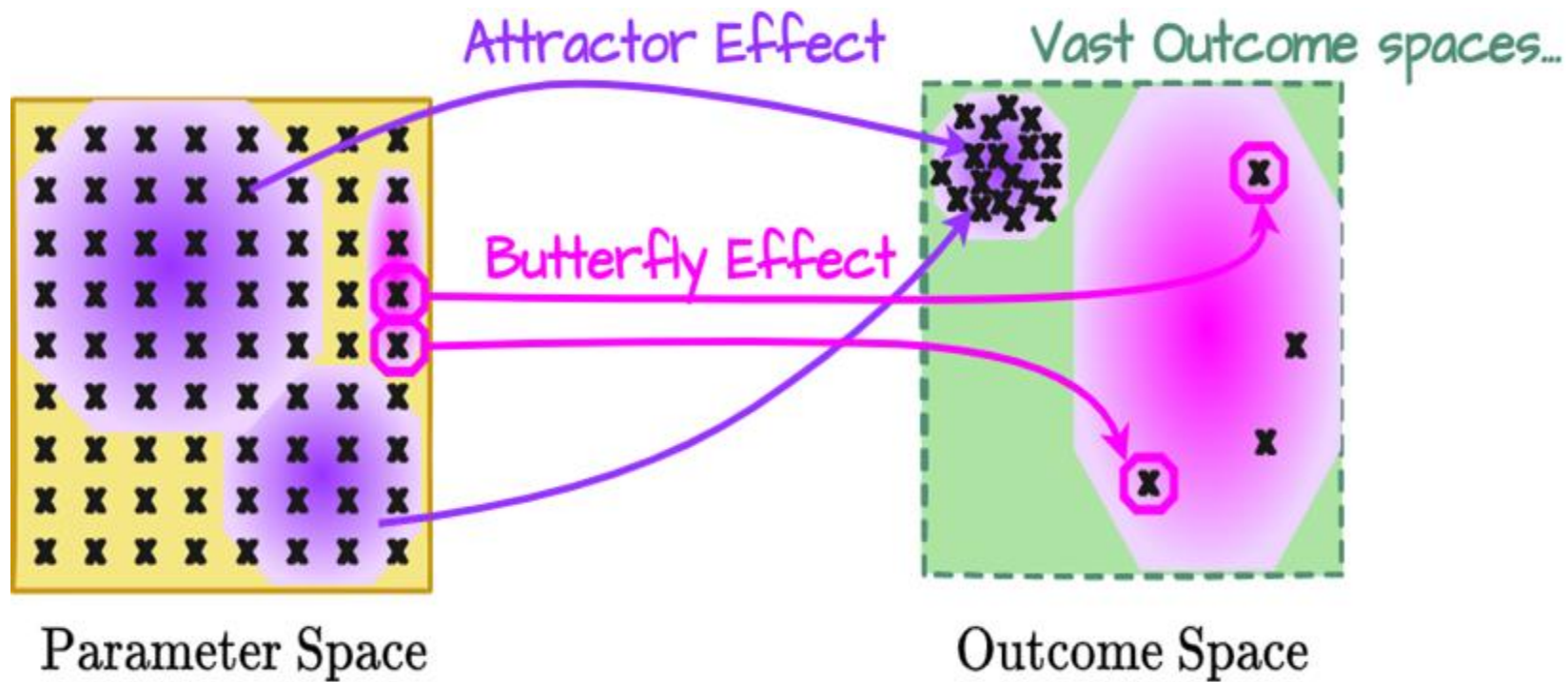
The child as an autotelic sense-making organism:

Exploring to make good predictive models of the world and control it!

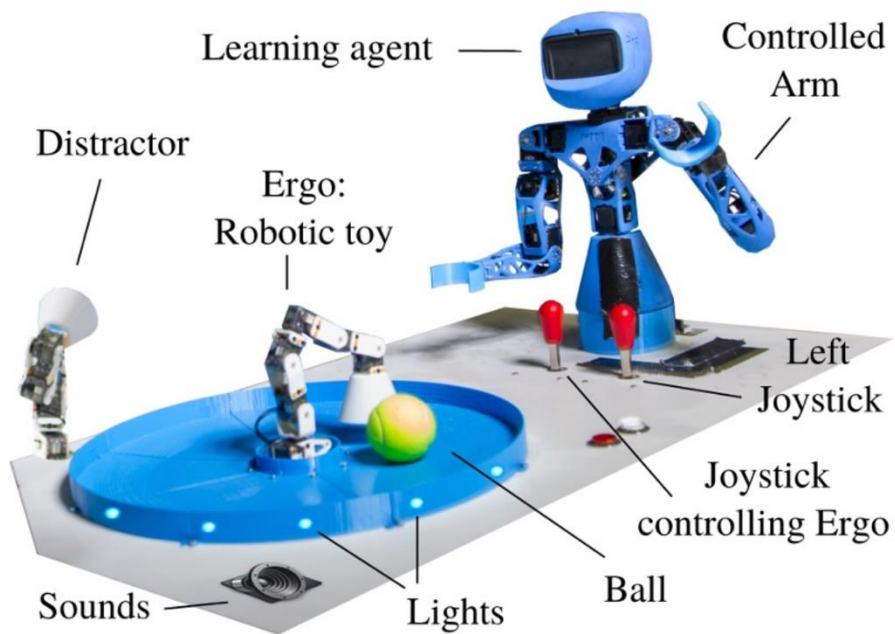


Autotelic agents (IMGEP algorithmic framework)

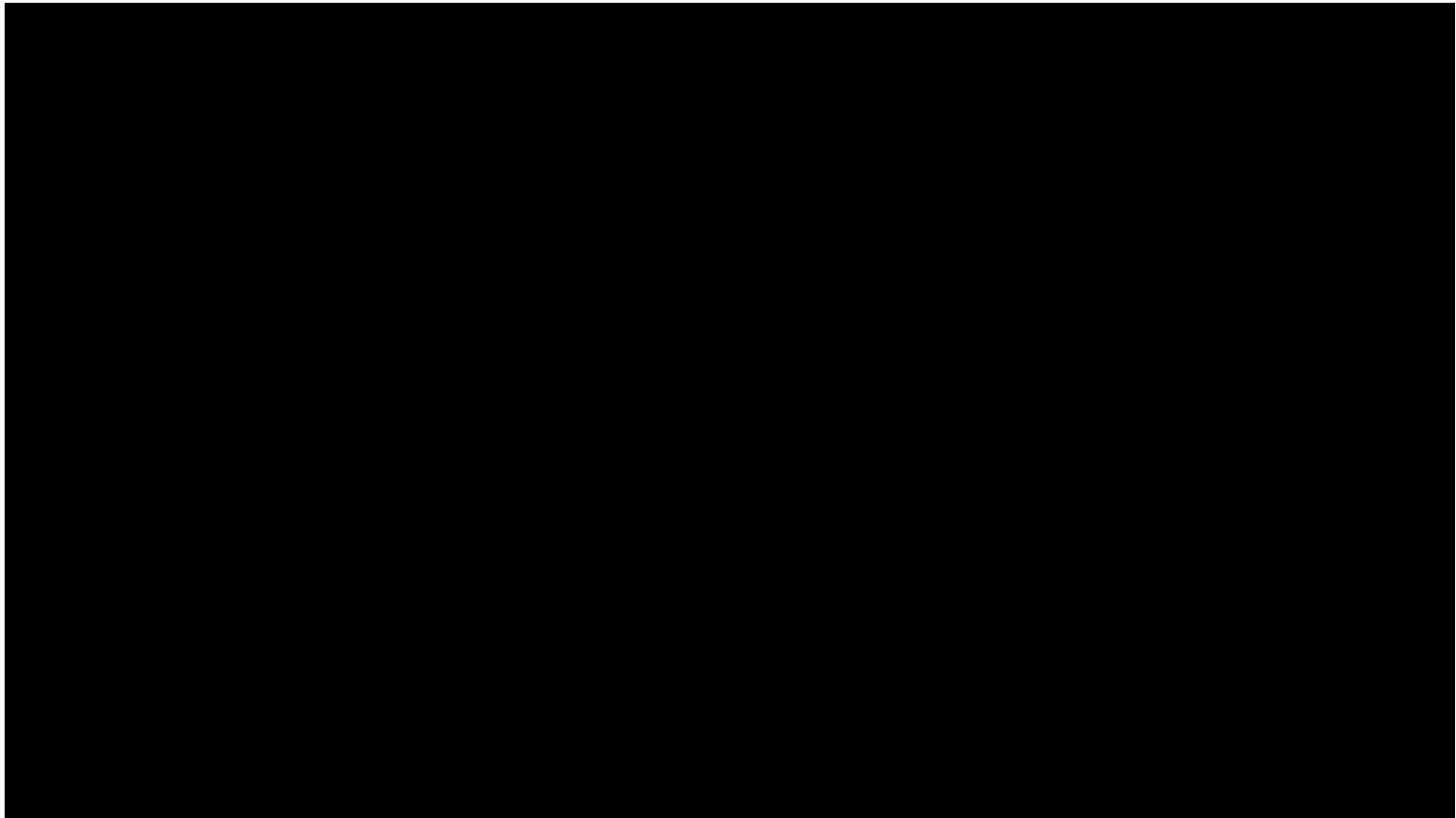




Curiosity-driven autotelic robots



Curiosity-driven autotelic robots



CHEMISTRY

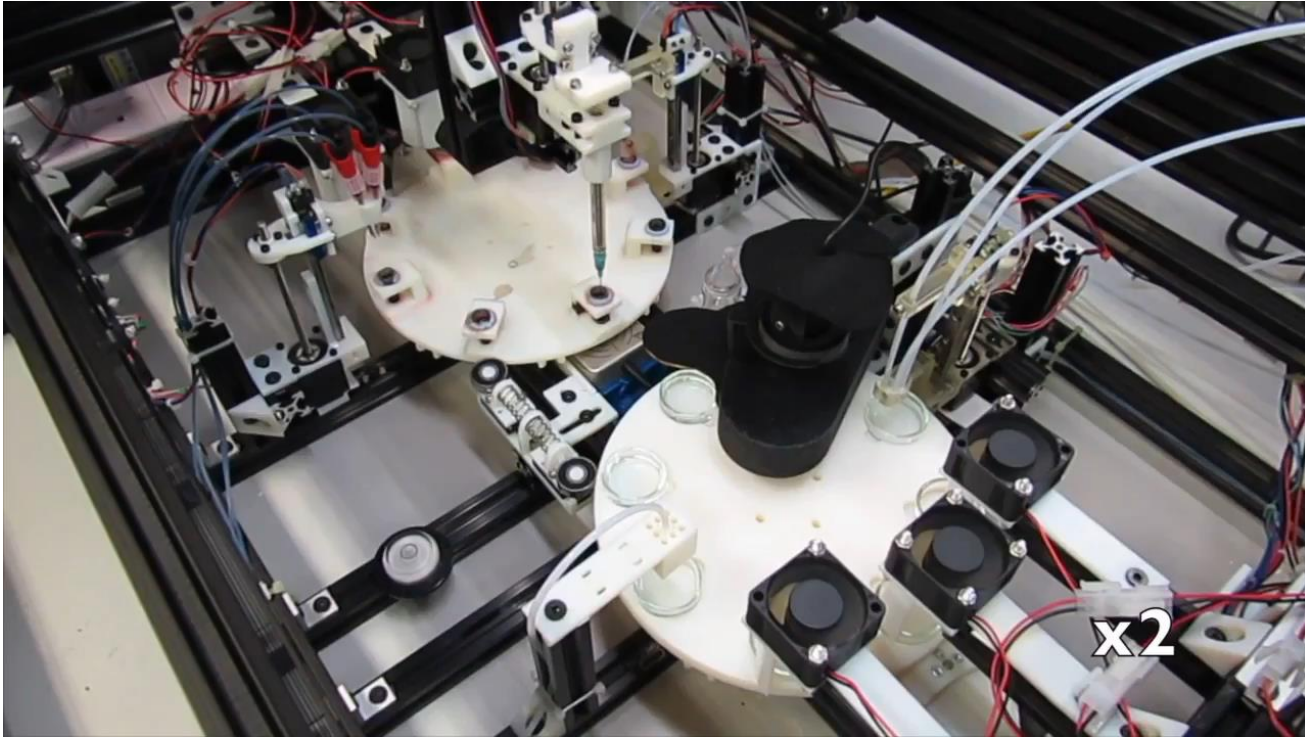
A curious formulation robot enables the discovery of a novel protocell behavior

Jonathan Grizou, Laurie J. Points, Abhishek Sharma, Leroy Cronin*

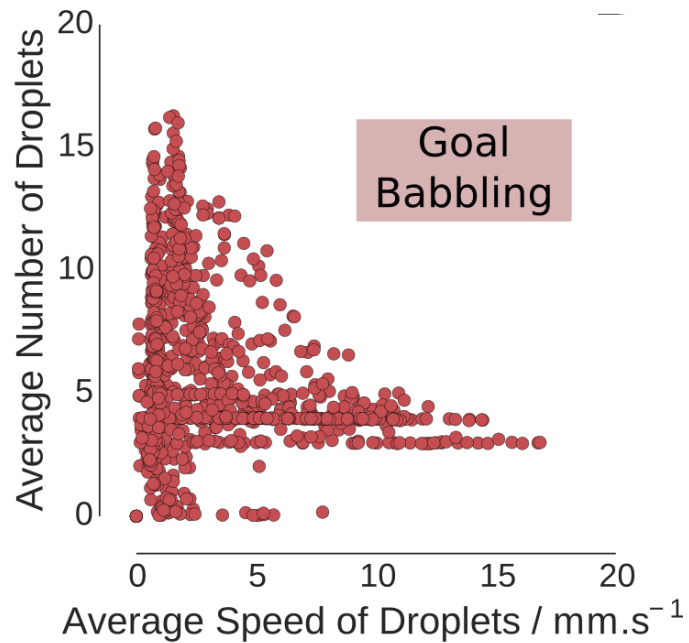
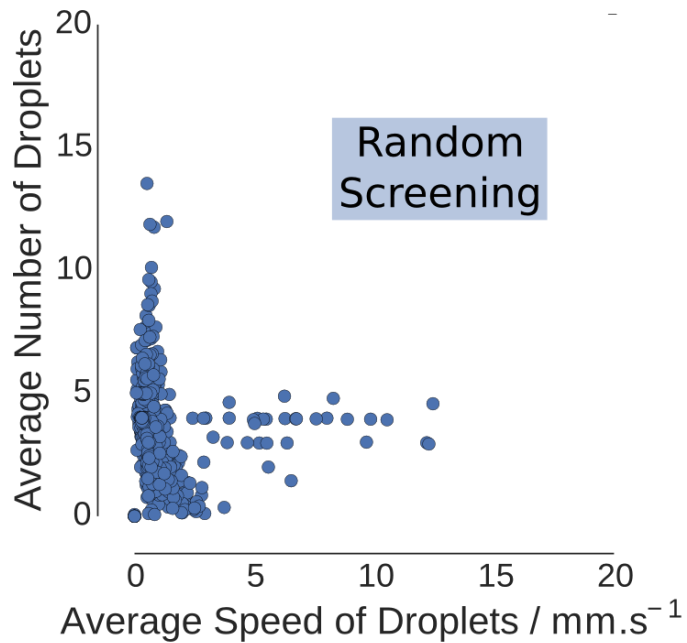
<https://www.science.org/doi/pdf/10.1126/sciadv.aay4237>



Automatized robot experiments



- 8 experiments running in parallel
- Specialized and stationary working stations
- Oils and surfactant handled separately



Limit: hand defined single goal space



Generating a Diversity of Challenging Programming Puzzles with Autotelic Generative Models

Julien Pourcel,
Inria

Cédric Colas
MIT, Inria

Gaia Molinaro
University of California, Berkeley

Pierre-Yves Oudeyer
Inria

Laetitia Teodorescu
Inria

Python Programming Puzzles (P3)

```
1 def f(x: float, a=1020) -> bool:
2     """Find a number that when squared is close to a."""
3
4     return abs(x ** 2 - a) < 10 ** -3
5
6 def g(a=1020) -> float:
7     return a ** 0.5
8
9 assert f(g()) == True
10
11 f(g())
```

[14] ✓ 0.2s

... True

Problem to solve and test function
(test units) return True/False
True: correct solution,
False: incorrect solution

Python Programming Puzzles (P3)

```
1 def f(x: float, a=1020) -> bool:
2     """Find a number that when squared is close to a."""
3
4     return abs(x ** 2 - a) < 10 ** -3
5
6 def g(a=1020) -> float:
7     return a ** 0.5
8
9 assert f(g()) == True
10
11 f(g())
```

[14] ✓ 0.2s

Problem to solve and test function
(test units) return True/False
True: correct solution,
False: incorrect solution

Solution function
return a solution

... True

Python Programming Puzzles (P3)

```
1 def f(x: float, a=1020) -> bool:
2     """Find a number that when squared is close to a."""
3
4     return abs(x ** 2 - a) < 10 ** -3
5
6 def g(a=1020) -> float:
7     return a ** 0.5
8
9 assert f(g()) == True
10
11 f(g())
```

[14] ✓ 0.2s

... True

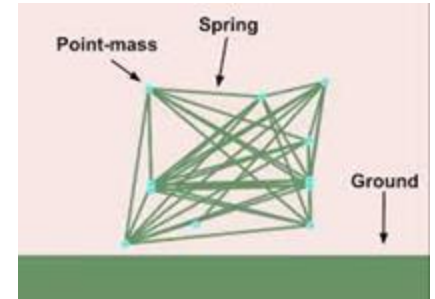
Problem to solve and test function (test units) return True/False
True: correct solution,
False: incorrect solution

Solution function
return a solution

Check syntax of the puzzle and **correctness**

Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

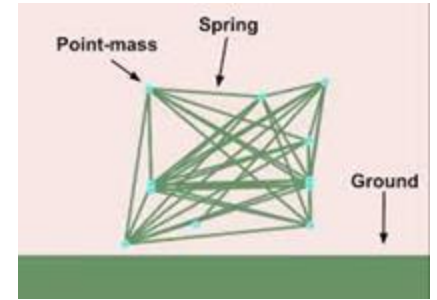
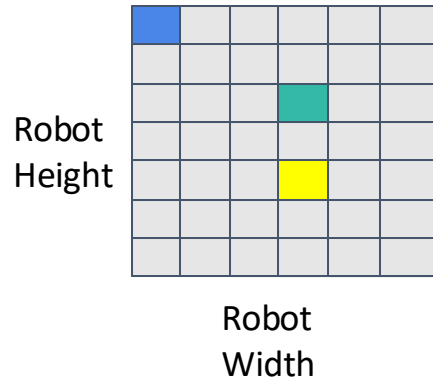
```
1 from walk_creator import walker_creator
2
3 def make_square(wc, x0, y0, x1, y1):
4     """ Make a square with top left x0,y0 and top right x1,y1 """
5     j0 = wc.add_joint(x0, y0)
6     j1 = wc.add_joint(x0, y1)
7     j2 = wc.add_joint(x1, y1)
8     j3 = wc.add_joint(x1, y0)
9
10    return j0, j1, j2, j3
11
12
13 def make_walker():
14     wc = walker_creator()
15
16     # the main body is a square
17     sides = make_square(wc, 0, 0, 10, 10)
18     center = wc.add_joint(5, 5)
19
20     # connect the square with distance muscles
21     for k in range(len(sides)-1):
22         wc.add_muscle(sides[k], sides[k+1])
23     wc.add_muscle(sides[3], sides[0])
24
25     # one prong of the square is a distance muscle
26     wc.add_muscle(sides[3], center)
27
28     # the other prongs from the center of the square are active
29     wc.add_muscle(sides[0], center, False, 5.0, 0.0)
30     wc.add_muscle(sides[1], center, False, 10.0, 0.0)
31     wc.add_muscle(sides[2], center, False, 2.0, 0.0)
32
33     return wc.get_walker()
```



Lehman, J., Gordon, J., Jain, S., Ndousse, K., Yeh, C., & Stanley, K. O. (2023). Evolution through large models. In *Handbook of Evolutionary Machine Learning* (pp. 331-366). Singapore: Springer Nature Singapore.

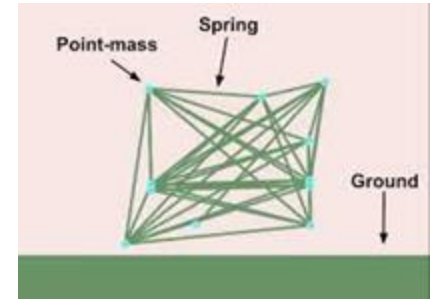
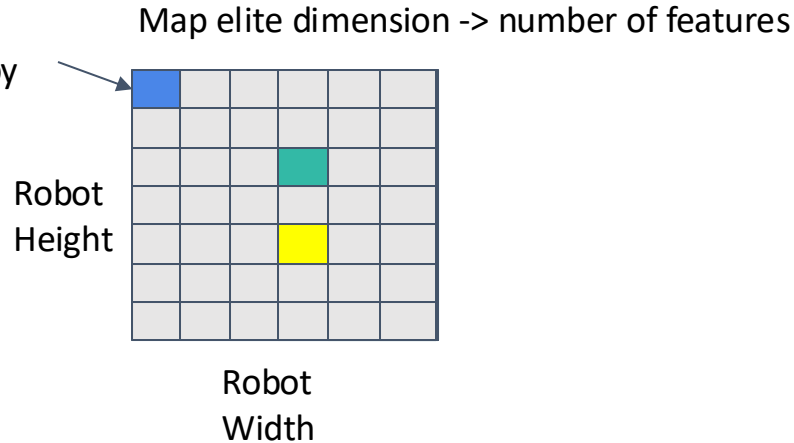
Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

Map elite dimension -> number of features



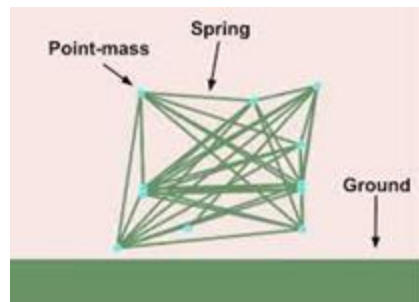
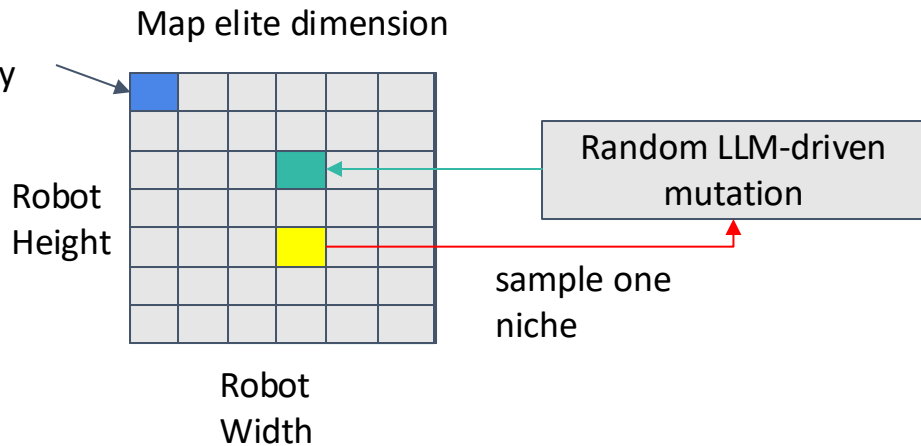
Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate



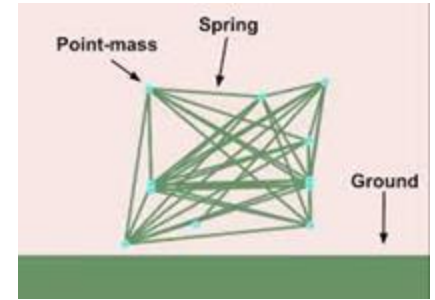
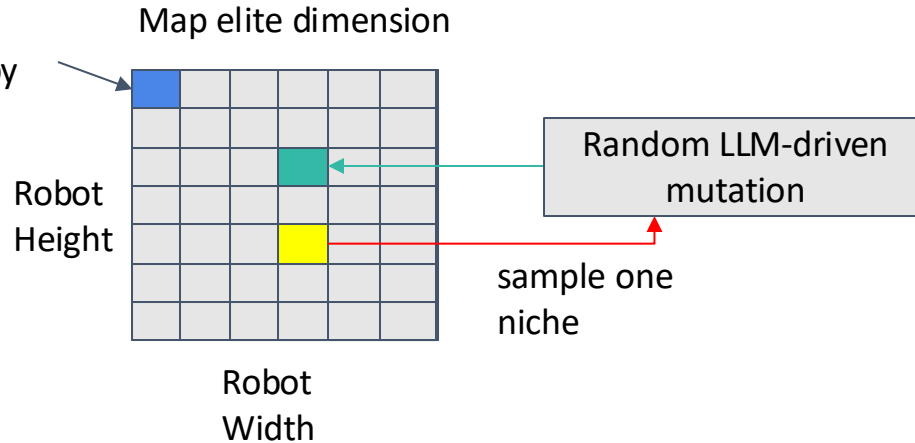
Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate



Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate

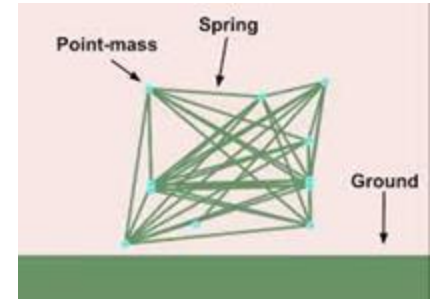
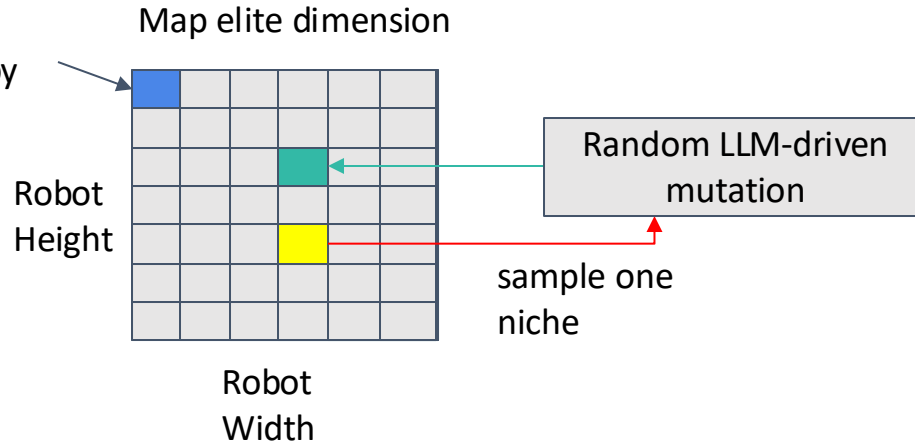


What descriptor can we use for P3 ?

What quality measure can we use for P3 ?

Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate

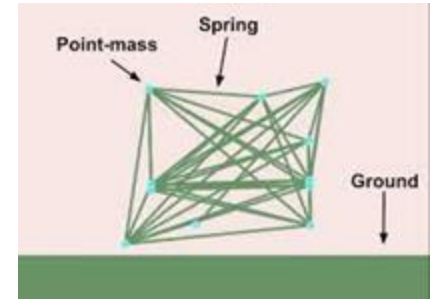
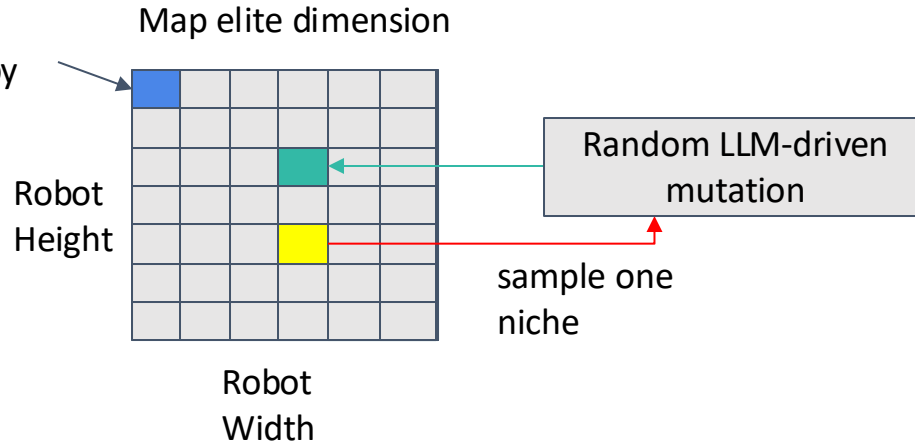


What descriptor can we use for P3 ?
What quality measure can we use for P3 ?

How to maximize diversity given limited resources?

Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate



What descriptor can we use for P3 ?

Semantic descriptors (recursion, set operations, ...)

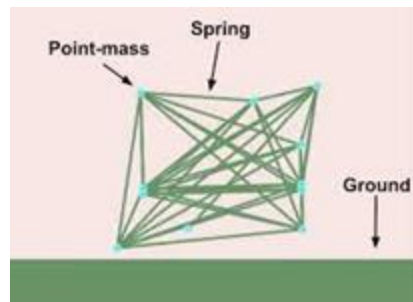
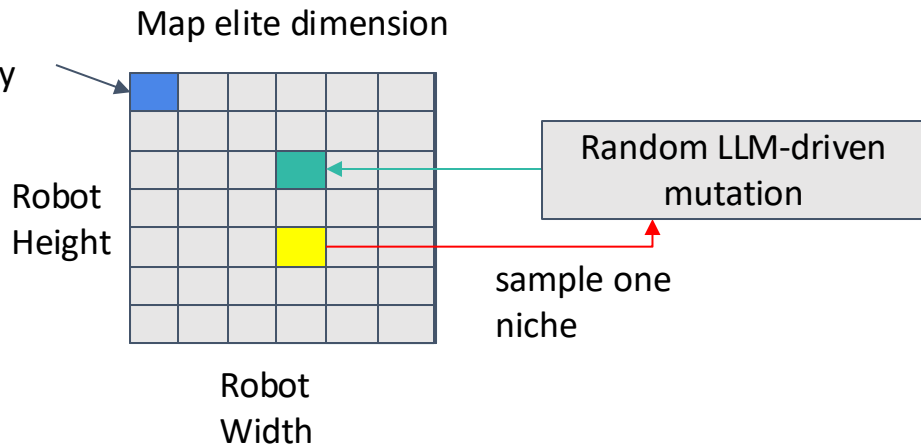
What quality measure can we use for P3 ?

Difficulty as measured by an external LLM solver

How to maximize diversity given limited resources?

Quality-Diversity algorithms (MAP Elites,...) as particular IMGEPs

discretize the map into
Niches
Niches are defined by
their coordinate



What descriptor can we use for P3 ?

Semantic descriptors (recursion, set operations, ...)

What quality measure can we use for P3 ?

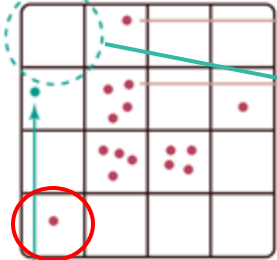
Difficulty as measured by an external LLM solver

How to maximize diversity given limited resources? Targeting new goals

Our proposed approach

sample target
cell [1, 2]

1



puzzle archive

List of semantic descriptors

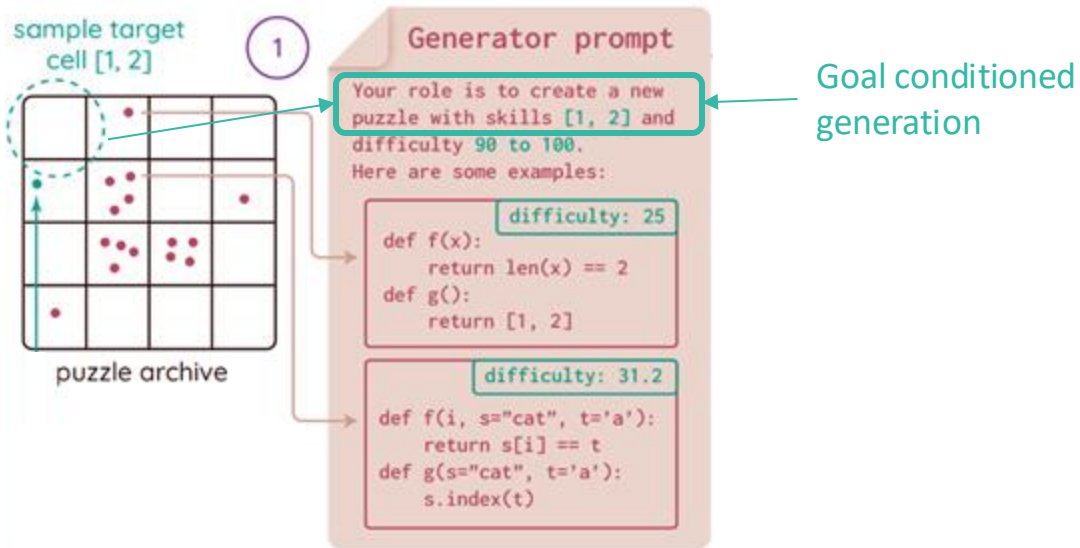
- 0 - String Manipulation
- 1 - Mathematical Operations
- 2 - Conditional Logic
- 3 - Recursion
- 4 - Brute Force Search
- 5 - Dynamic Programming
- 6 - Greedy Algorithms
- 7 - Backtracking
- 8 - Set Operations
- 9 - Permutations and Combinations
- 10 - Probability and Statistics
- 11 - Pattern Recognition
- 12 - Sorting and Ordering
- 13 - Binary Operations (bitwise shifting, AND, OR)
- 14 - Geometry and Coordinate Manipulation
- 15 - Algorithm Optimization
- 16 - Number Theory (factors, primes, etc.)
- 17 - Graph Theory (paths, edges, vertices)
- 18 - Array Indexing
- 19 - Hashing

coordinate of a niche described by a
combination of skills

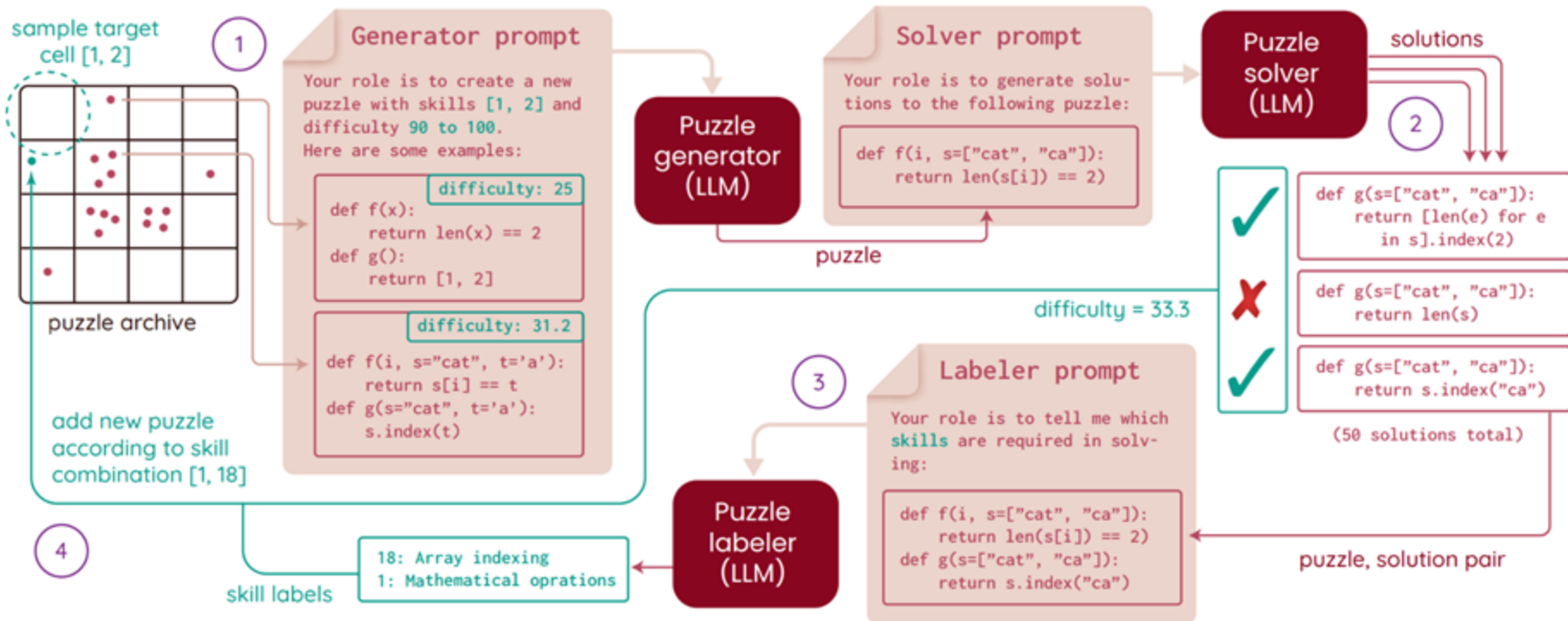


binary skill
vector

Our proposed approach

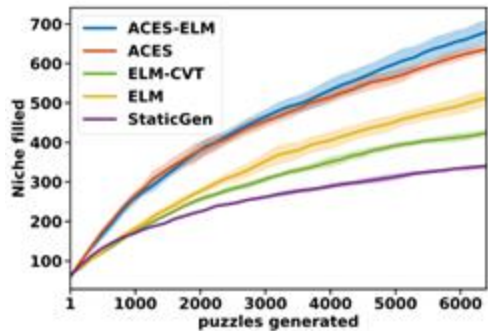


Our proposed approach

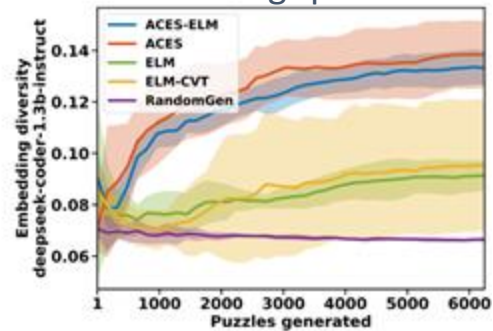
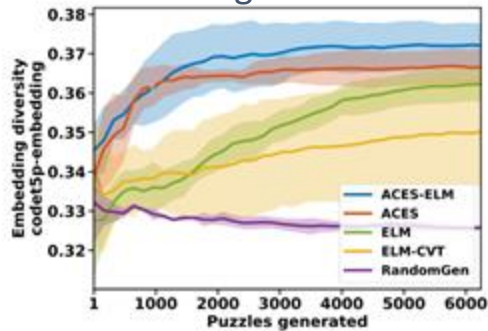


Quality-Diversity metrics

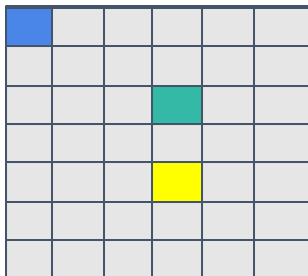
Niches Filled



Average Pairwise Distance in embedding space

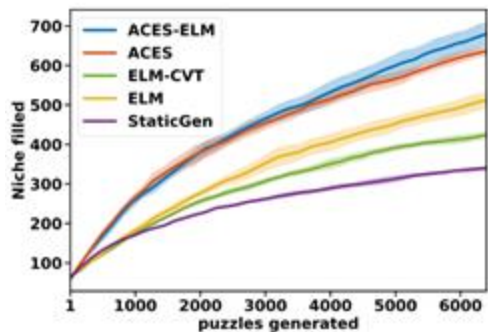


(higher is better)

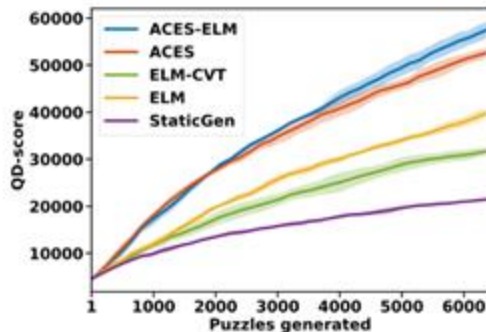
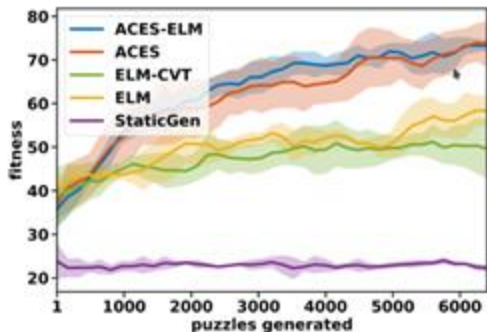
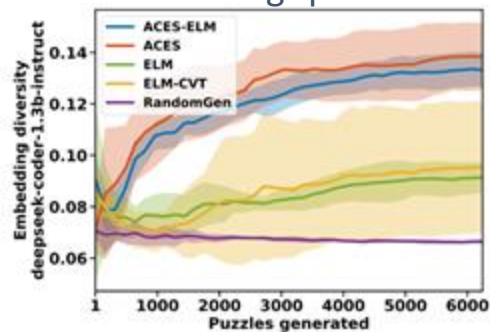
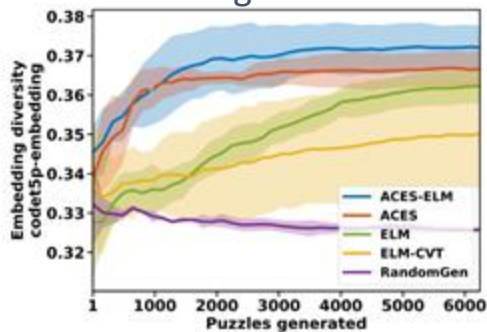


Quality-Diversity metrics

Niches Filled

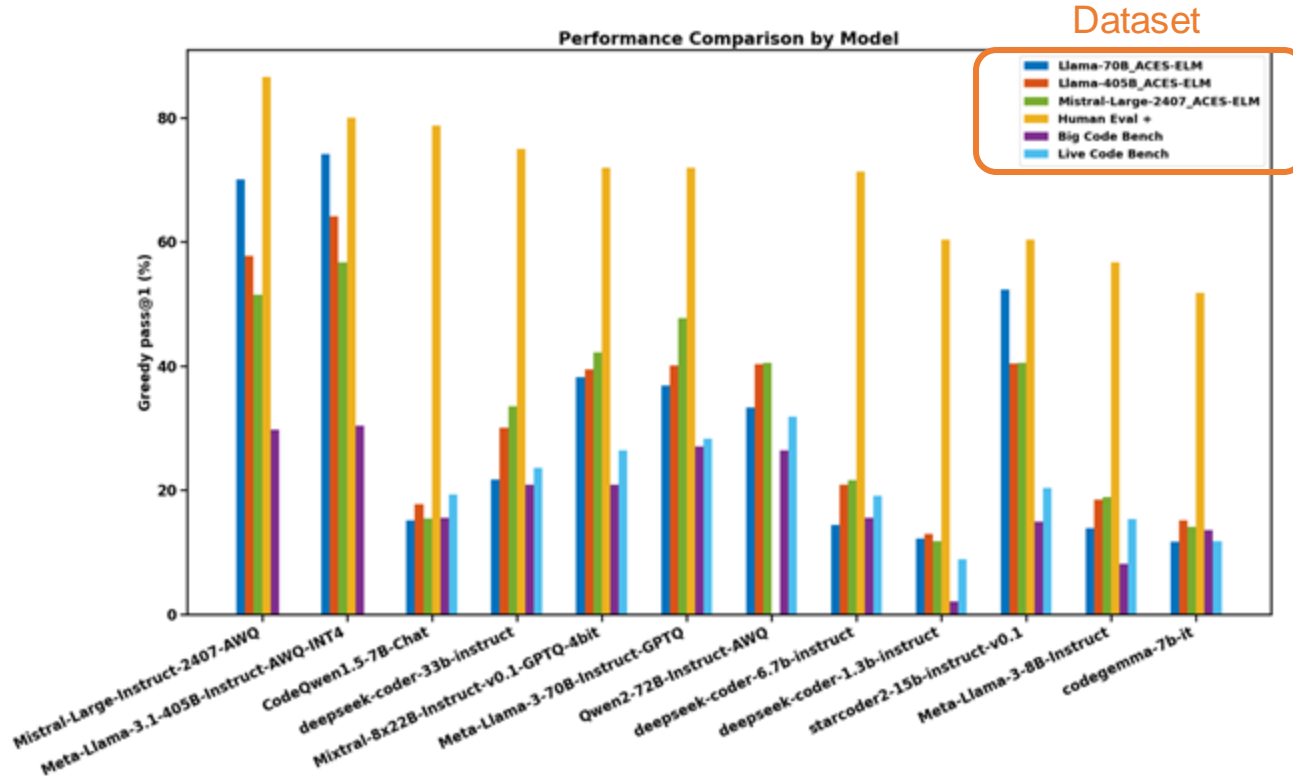


Average Pairwise Distance in embedding space



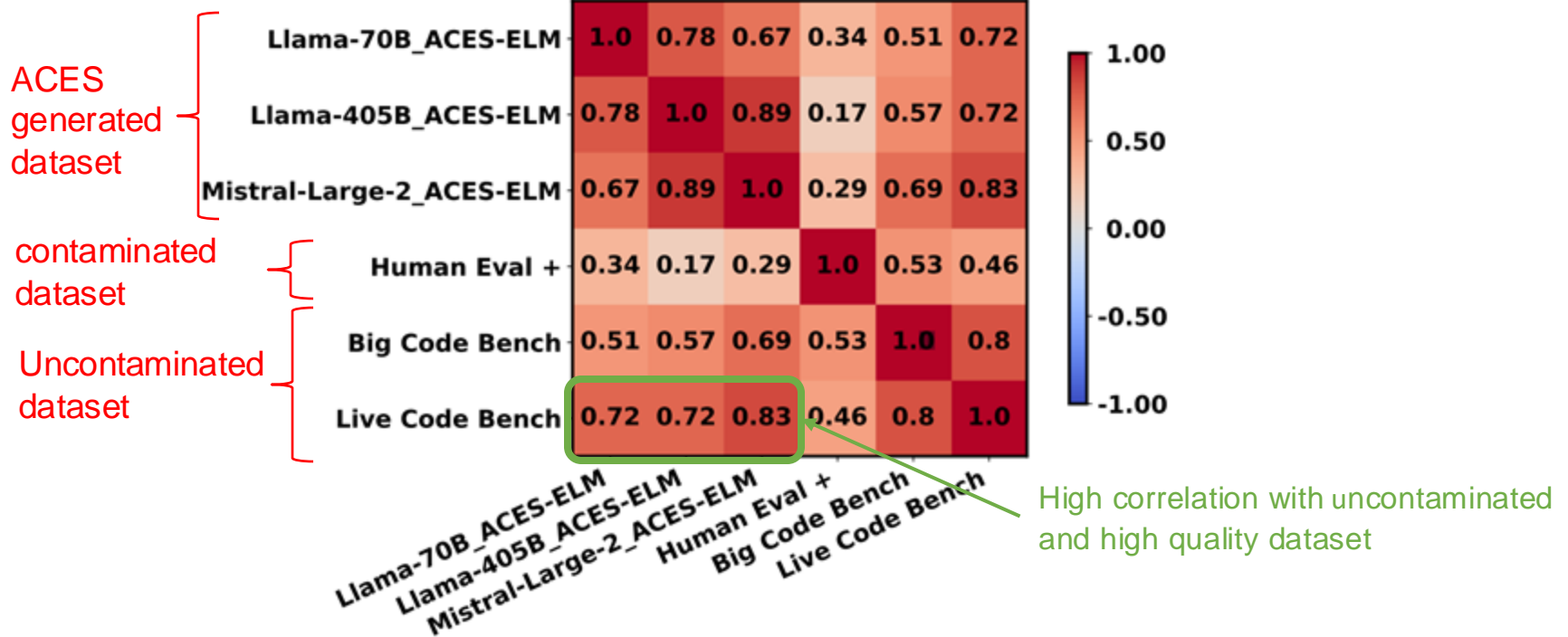
(higher is better)

Automatic benchmark generator



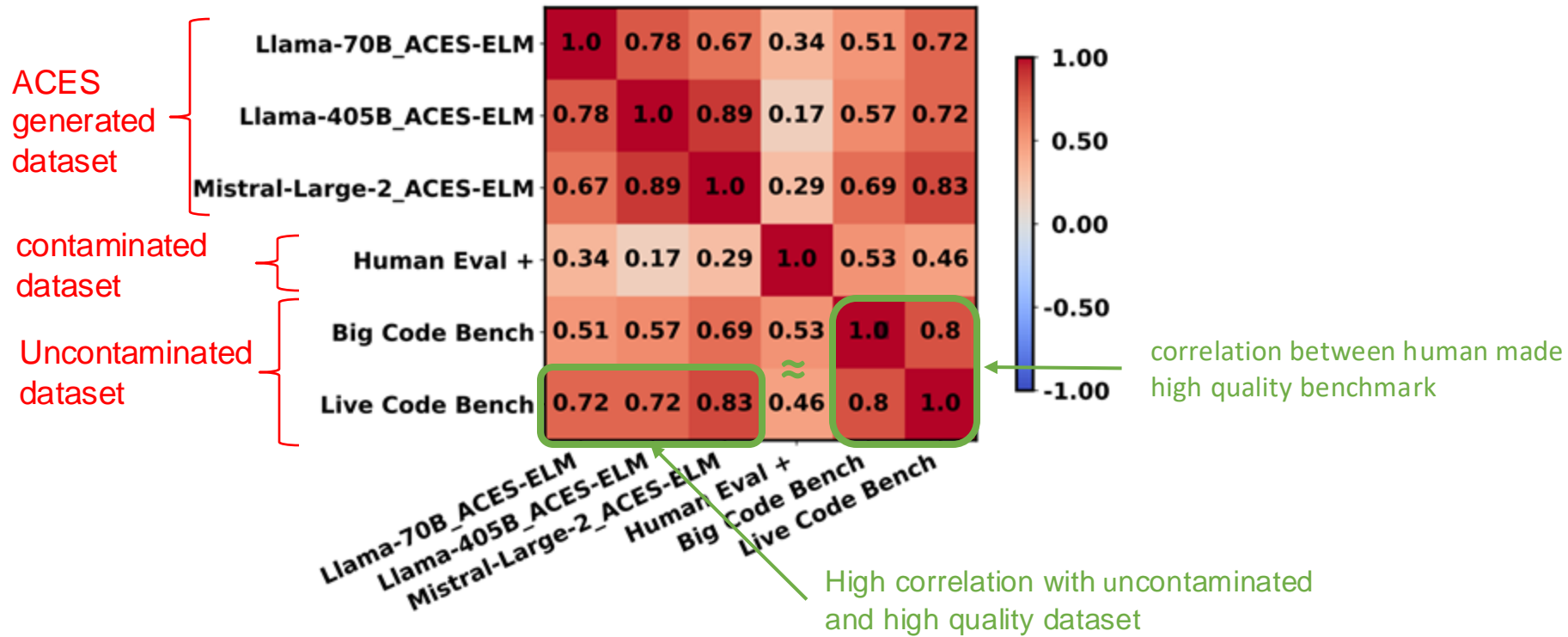
Greedy pass@1 scores for various models on multiple datasets

Automatic benchmark generator



correlation between pass@1 scores across all datasets, averaged over LLMs

Automatic benchmark generator



correlation between pass@1 scores across all datasets, averaged over LLMs

Conclusion

- We proposed a new algorithm to generate a diverse set of difficult puzzle
- Archive generated by our method seems to be the most diverse (given our diversity metrics)
- ACES seems promising as an Automatic benchmark generator (high correlation with latest human made dataset)

LLM-based Autotelic Agents in text worlds



```

Zork I: The Great Underground Empire
Behind House          Score: 0      Moves: 3
South of House
You are facing the south side of a white house. There is no door here, and all
the windows are boarded.

>go east
Behind House
You are behind the white house. A path leads into the forest to the east. In
one corner of the house there is a small window which is slightly ajar.

>open window
With great effort, you open the window far enough to allow entry.

>enter house
Kitchen
You are in the kitchen of the white house. A table seems to have been used
recently for the preparation of food. A passage leads to the west and a dark
staircase can be seen leading upward. A dark chimney leads down and to the east
is a small window which is open.
On the table is an elongated brown sack, smelling of hot peppers.
A bottle is sitting on the table.
The glass bottle contains:
  A quantity of water
>
  Do it
  ⬆ = up arrow   ⬇ = down arrow   ⬅ = left arrow   ➡ = right arrow
  Restart Game   Return to Elevator   Return to Lobby
  
```

You are playing a video game, here is what you did previously:
<trajectory>

Exercise: find a sequence of 2, 3, or 4 instructions that will help me achieve a new, interesting or creative goal in this game.

Instructions:
<previous goal 1>
<previous goal 2>
...

First describe the new goal starting with an imperative verb; then list the instructions.

Answer:
Goal: prepare cilantro dish
Instructions: open fridge;
pick up cilantro; pick up knife;
cut cilantro; cook cilantro

Prepare cilantro dish
Open fridge;
pick up cilantro...

Goal Generator

- Step 0: open fridge
- Step 0: you open the fridge.
- Step 1: pick up cilantro
- Step 1: you pick up the cilantro.
- Step 2: close fridge
- Step 2: you close the fridge.
- Step 3: pick up knife
- Step 3: you pick up the knife.
- Step 4: pick up pork chop
- Step 4: you pick up the pork chop.
- Step 5: chop cilantro
- Step 5: you chop the cilantro.
- Step 6: roast pork chop
- Step 6: you roast the pork chop.
- Step 7: prepare meal
- Step 7: you prepare the meal.
- Step 8: eat meal
- Step 8: you eat the meal.

Prepare cilantro dish
Open fridge;
pick up cilantro...

Rollout

- Step 6: you roast the pork chop.
- Step 7: prepare meal
- Step 7: you prepare the meal.
- Step 8: eat meal
- Step 8: you eat the meal.

Exercise: Given the description of a player's behavior in a video game, list the most interesting, impressive, novel or creative goals he achieved and, for each goal, specify when it is achieved for the first time.

Answer:
Open fridge (Step 0)
Enjoy delicious meal (Step 8)
Open fridge and then close it (Step 2)
Eat healthy (Step 8)
Clean kitchen (Step 4)

Relabeler

- Step 8: eat meal
- Step 8: you eat the meal.

Exercise: Given the description of a player's behavior in a video game, list the most interesting, impressive, novel or creative goals he achieved and, for each goal, specify when it is achieved for the first time.

Here is the list: "prepare cilantro dish", "open fridge and close it", "eat healthy", "clean kitchen"

Answer:
Enjoy delicious meal: yes (step 8)
Open fridge and then close it: yes (step 2)
Eat healthy: yes (step 8)
Clean kitchen: no
Prepare cilantro dish: yes (step 7)
Open fridge: yes (step 0)
...

Reward Function

AUGMENTING AUTOTELIC AGENTS WITH LARGE LANGUAGE MODELS



Augmenting Autotelic Agents with LLMs

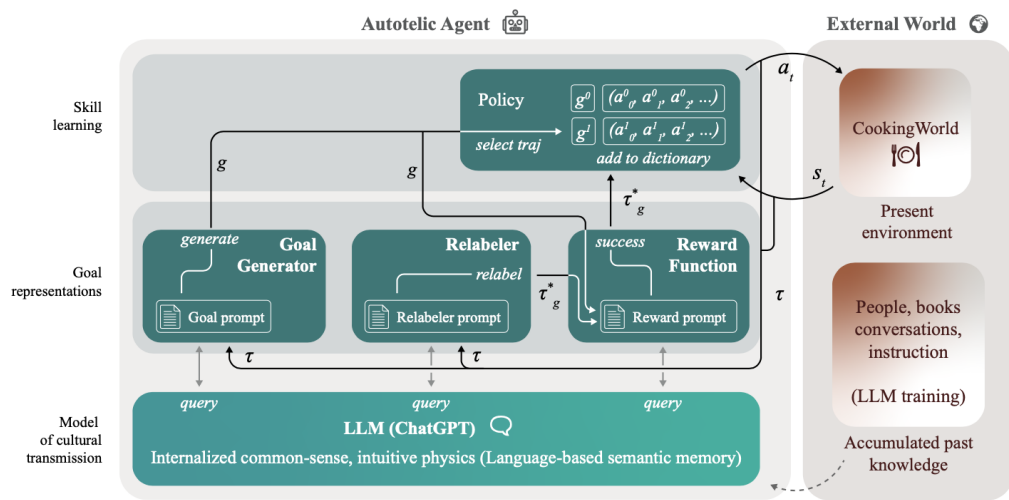
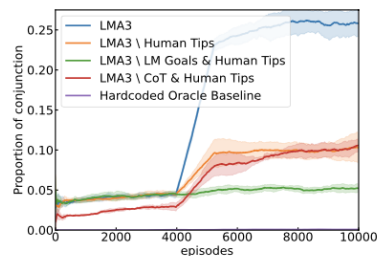


Figure 2: **General architecture of LMA3.** LMA3 assumes access to a model of cultural transmission implemented via ChatGPT (dashed line). As shown in the goal representations block, LMA3 leverages that model to generate goals g (left), relabel past trajectories $\tau = (s_0, a_0, \dots, s_T, a_T)$ as τ_g^* (middle) and compute rewards when goals are reached (right). s_t and a_t denote the state representation and the action taken at game step t . The goal-conditioned policy (top) attempts to reach its goal within *CookingWorld* (right) and uses relabels and rewards for learning.



- Step 0: open fridge
- Step 0: you open the fridge.
- Step 1: pick up cilantro
- Step 1: you pick up the cilantro.
- Step 2: close fridge
- Step 2: you close the fridge.
- Step 3: pick up knife
- Step 3: you pick up the knife.
- Step 4: pick up pork chop
- Step 4: you pick up the pork chop.
- Step 5: chop cilantro
- Step 5: you chop the cilantro.
- Step 6: roast pork chop
- Step 6: you roast the pork chop.
- Step 7: prepare meal
- Step 7: you prepare the meal.
- Step 8: eat meal
- Step 8: you eat the meal.



(a)