(Neural) program synthesis from examples

Gabriele Libardi

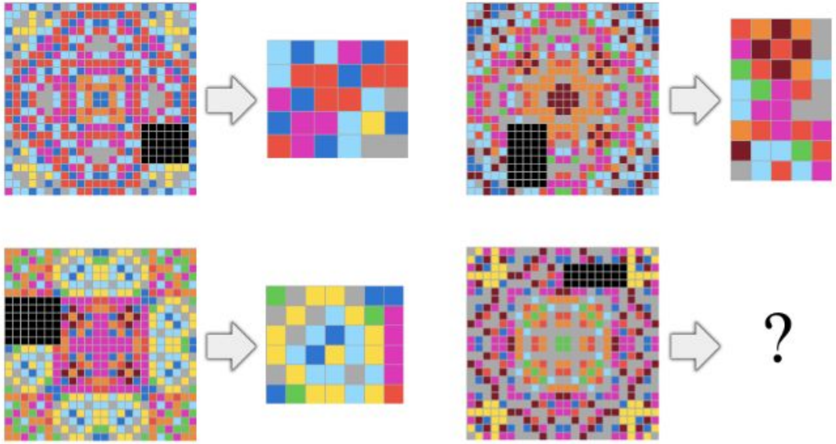
Program synthesis from examples

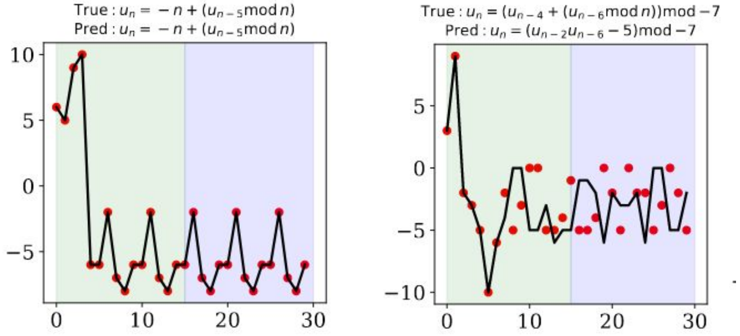
Program 1:
```
w ← [int]
t ← [int]
c ← MAP (*3) w
d ← ZIPWITH (+) c t
e ← MAXIMUM d
```

Input-output example:
Input:
[6 2 4 7 9],
[5 3 6 1 0]
Output:
27

Program 2:
```
a ← [int]
b ← [int]
c ← ZIPWITH (-) b a
d ← COUNT (>0) c
```

Input-output example:
Input:
[6 2 4 7 9],
[5 3 2 1 0]
Output:
4





Balog et al. - 2016
Chollet - 2019
d'Ascoli, Kamienny et al. - 2022

Weak vs Strong generalization

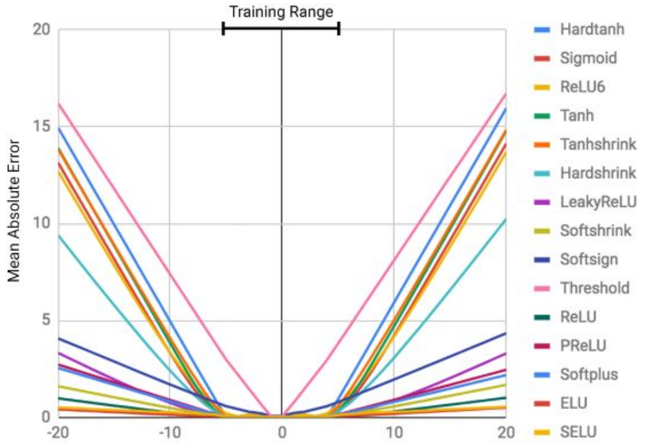


Figure 1: MLPs learn the identity function only for the range of values they are trained on. The mean error ramps up severely both below and above the range of numbers seen during training.

Trask et al. - 2018

Other advantages

- Explainable and verifiable
- Sample efficiency
- Generalization
- Rule learning in humans

RESEARCH

RESEARCH ARTICLES

COGNITIVE SCIENCE

Human-level concept learning through probabilistic program induction

Brenden M. Lake,[1*] Ruslan Salakhutdinov,[2] Joshua B. Tenenbaum[3]

People learning new concepts can often generalize successfully from just a single example, yet machine learning algorithms typically require tens or hundreds of examples to perform with similar accuracy. People can also use learned concepts in richer ways than conventional algorithms—for action, imagination, and explanation. We present a computational model that captures these human learning abilities for a large class of

Lake et al. - Science - 2015

Flash Fill

	A	B	C
1	Name and ID	First name and last name	ID #
2	Thomas, Rhonda 82132	Rhonda Thomas	
3	Emmett, Keara 34231	Keara Emmett	
4	Vogel, James 32493	James Vogel	
5	Jelen, Bill 23911	Bill Jelen	
6	Miller, Sylvia 78356	Sylvia Miller	
7	Lambert, Bobby 25900	Bobby Lambert	
8	Sweet, Julie 65477	Julie Sweet	
9	Williams, Don 43920	Don Williams	
10	Spake, Deborah 33488	Deborah Spake	

Domain Specific Language (DSL)

top-level expr $T := C \mid ifThenElse(B, C, T)$

condition-free expr $C := A \mid Concatenate(A, C)$

atomic expr $A := SubString(x, P, P) \mid Const(w)$

position expr $P := K \mid RegPos(x, R, K)$

Gulwani - 2011

Symbolic back-propagation

Spec: "Luc de Raedt, Luc.deraedt@kuleuven.be" → "raedt-Luc"

Grammar: $Concatenate$(e1, e2)

Backprop: $Concatenate^{-1}$("raedt-Luc") =
{ ("r", "aedt-Luc"), …, ("raedt-", "Luc") ,…, ("raedt-Lu", "c") }

Gulwani - 2011

Symbolic back-propagation

Spec: “Luc de Raedt, Luc.deraedt@kuleuven.be” → “Luc”

Grammar: $SubString$(input, p1, p2)

Backprop: $SubString^{-1}$(“Luc”) = { (input,1,4), (input,15,18) }

Gulwani - 2011

Symbolic back-propagation

Spec: “Luc De Raedt, luc.deraedt@kuleuven.be” → “raedt” (last name)

Grammar: $ToLower$(s)

Backprop: $ToLower^{-1}$(raedt) = { raedt, Raedt, RAedt, RAEdt, ... }

With forwardprop filtering: $ToLower^{-1}$(raedt) = { Raedt, raedt }

Spec: “Price $24.58, Units 25.6 gm” → “25” (intent: round price up)

Grammar: $RoundNum$(num, roundDesc)

Backprop: $RoundNum^{-1}$(25) = { (24.01, Up), (24.02, Up), ..., (25.02, Down), (24.51, Near), ...

Gulwani - 2011

MarkovJunior

For example, consider this Markov algorithm in the alphabet `{0, 1, x}` (ε is the empty word):

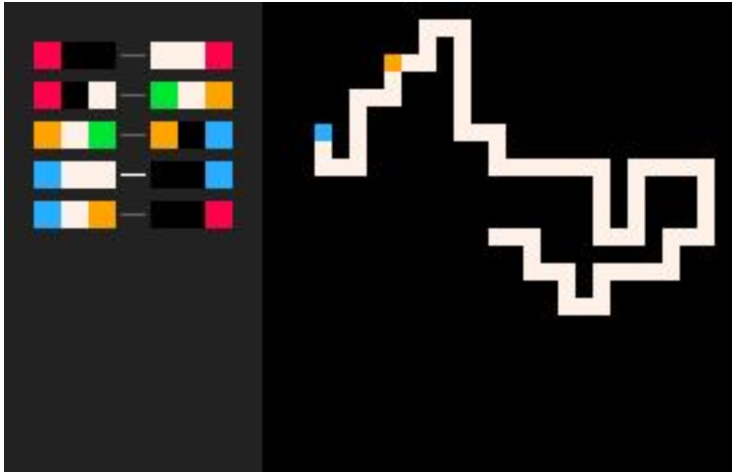
```
1=0x
x0=0xx
0=ε
```

If we apply it to the string `110` we get this sequence of strings:

```
110 -> 0x10 -> 0x0x0 -> 00xxx0 -> 00xx0xx -> 00x0xxxx -> 000xxxxxx -> 00xxxxxx -> 0xxxxxx -> xxxxxx
```

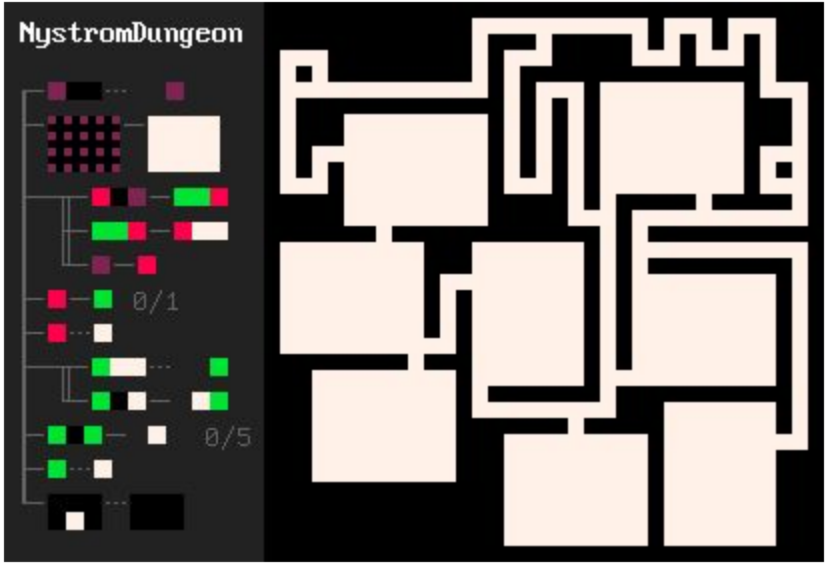
Gumin - 2022

MarkovJunior



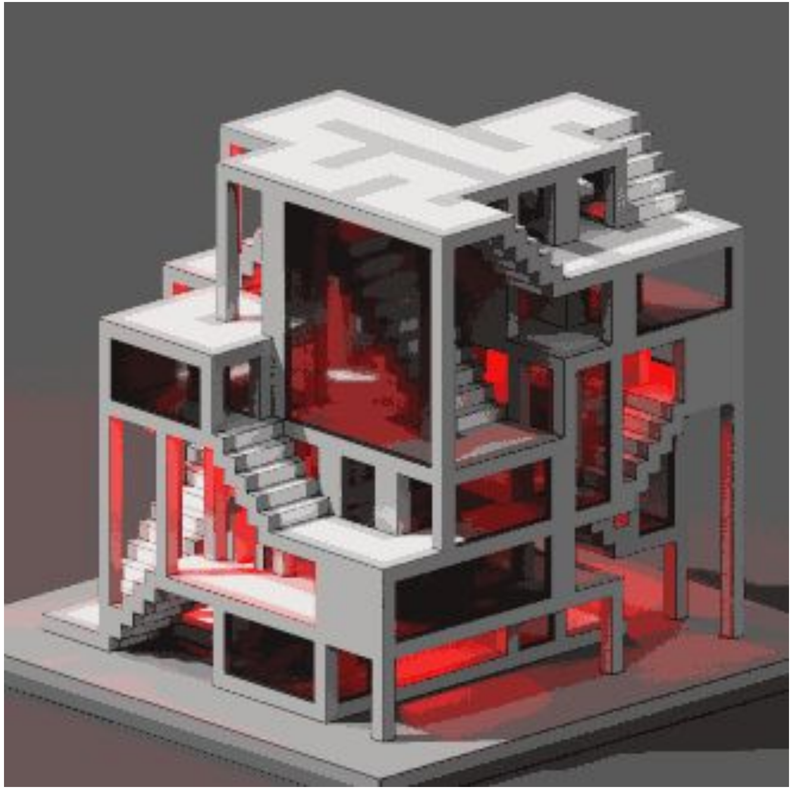
Gumin - 2022

MarkovJunior



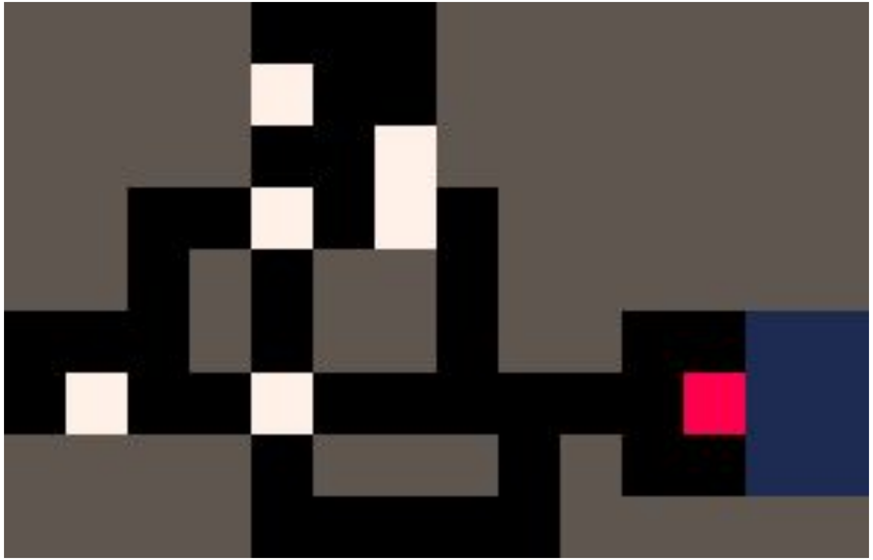
Gumin - 2022

MarkovJunior



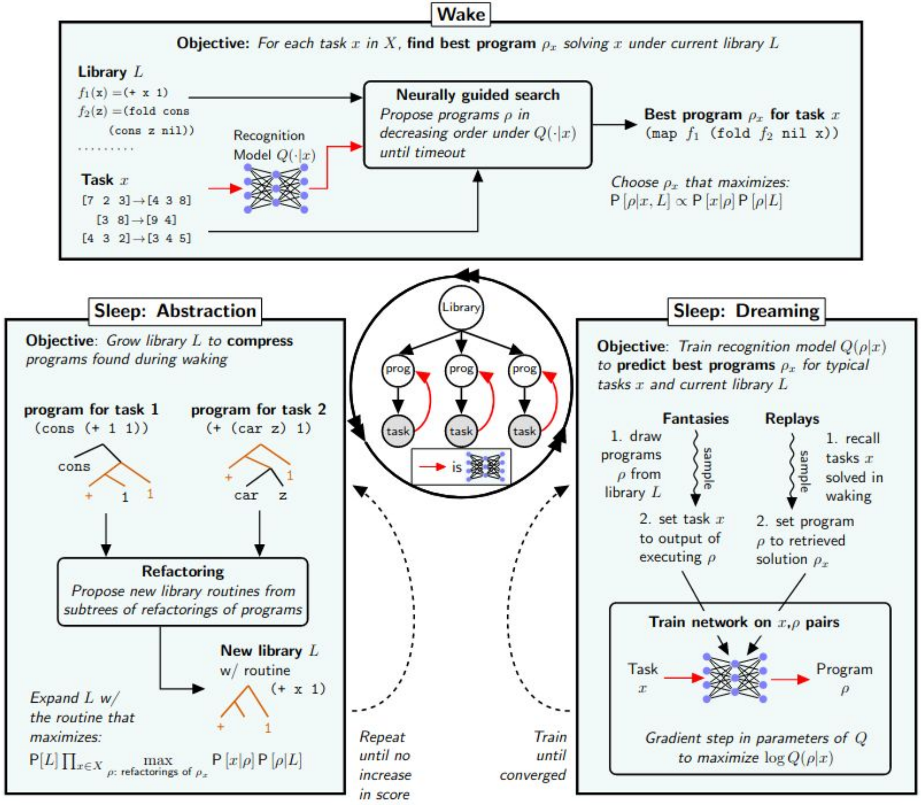
Gumin - 2022

MarkovJunior



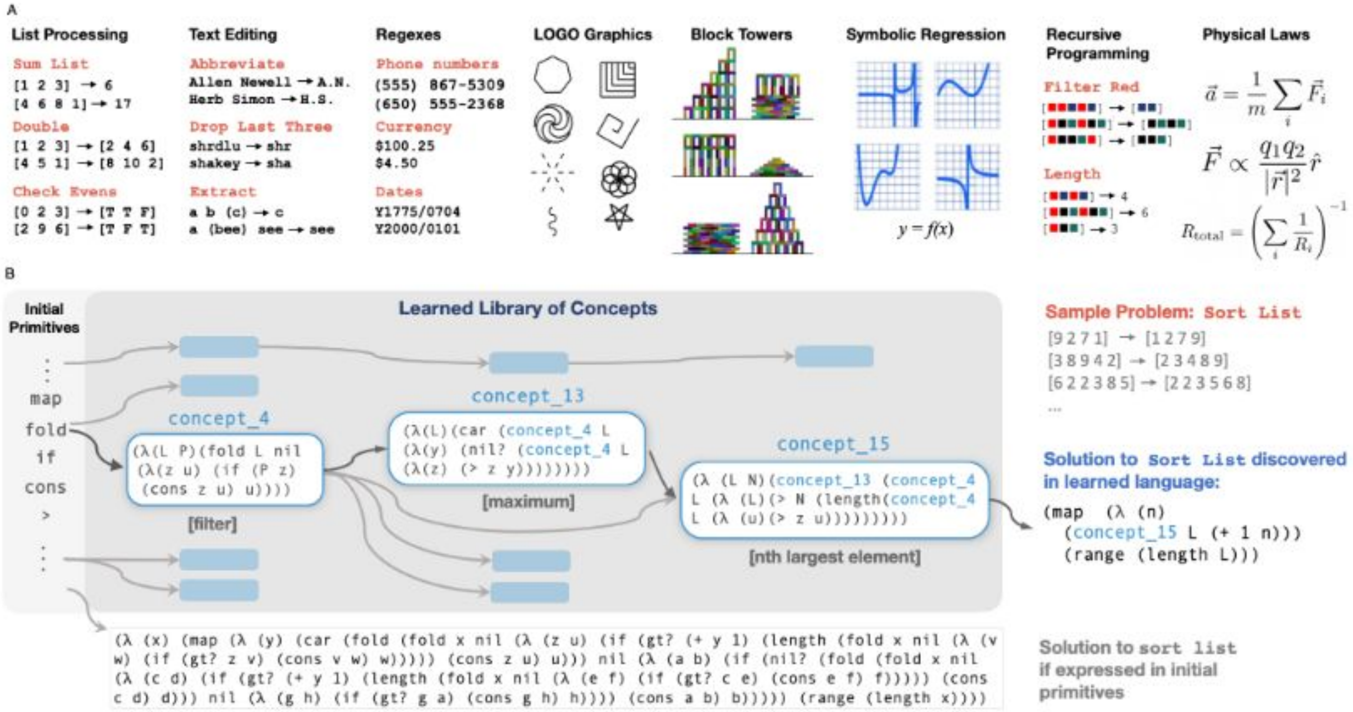
Gumin - 2022

DreamCoder



Ellis et al. - ACM SIGPLAN - 2021

DreamCoder



Ellis et al. - ACM SIGPLAN - 2021

DreamCoder

- Hierarchy of abstractions
- Gradually builds a library of composite functions
- Generates its own training data
- Learns to search

Write, Execute, Assess: Program Synthesis with a REPL

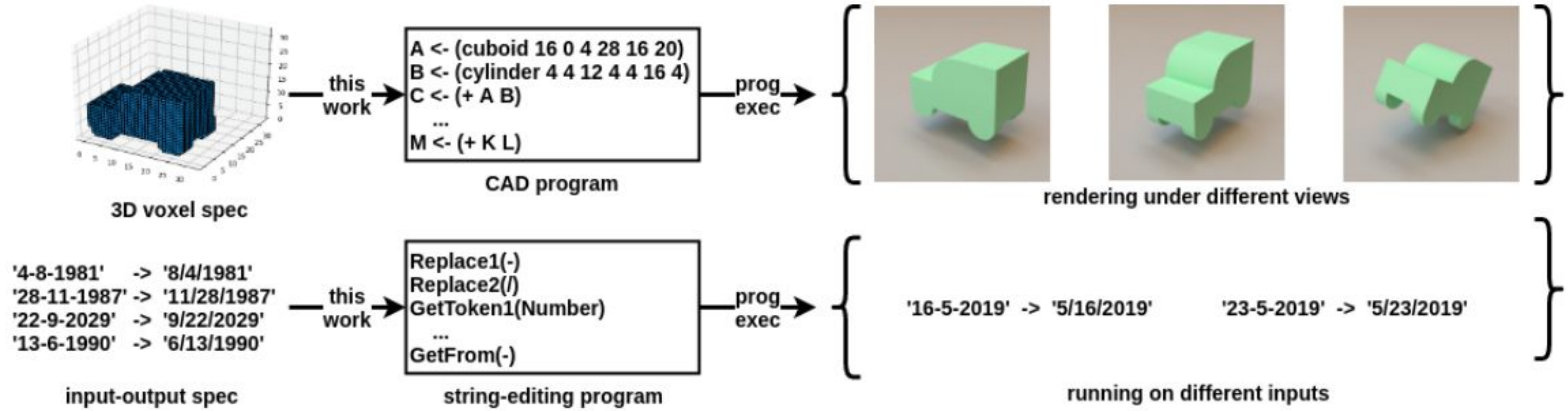


Figure 1: Examples of programs synthesized by our system. Top, graphics program from voxel specification. Bottom, string editing program from input-output specification.

Ellis et al. - NeurIPS - 2019

Write, Execute, Assess: Program Synthesis with a REPL

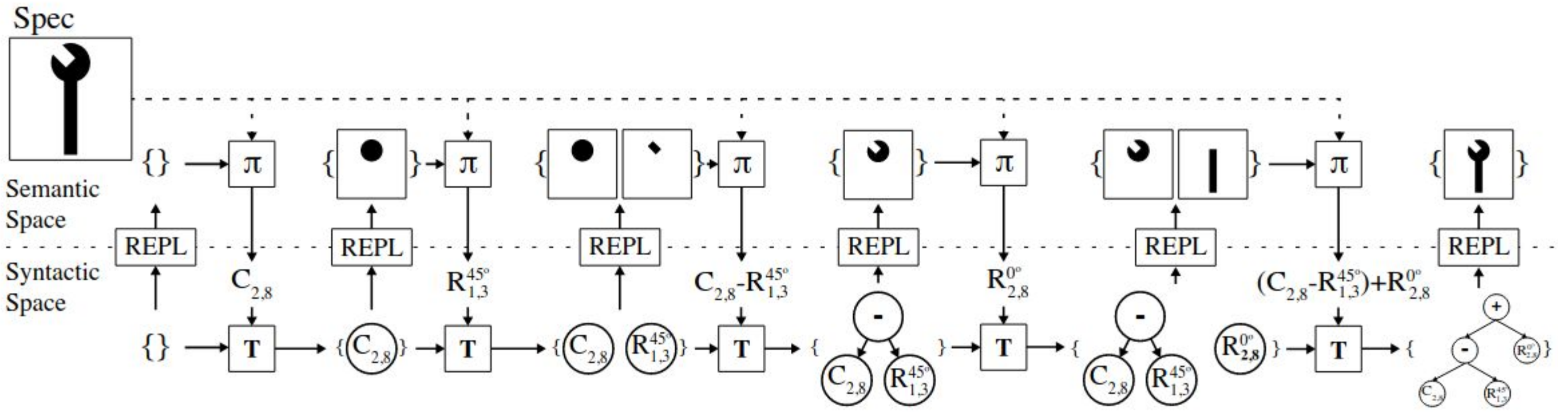


Figure 2: A particular trajectory of the policy building a 2D wrench. At each step, the REPL renders the set of partial programs pp into the semantic (image) space. These images are fed into the policy π which proposes how to extend the program via an action a, which is incorporated into pp via the transition T.

Ellis et al. - NeurIPS - 2019

Program synthesis disguised..

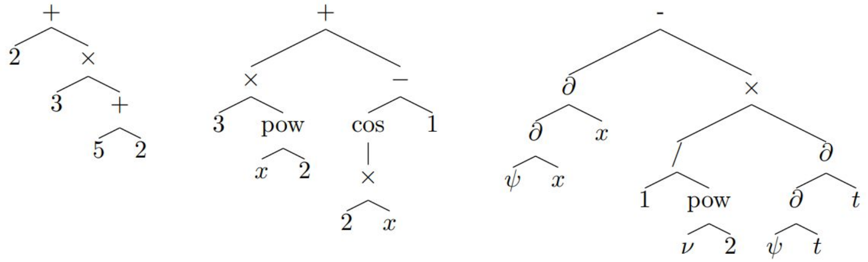
DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Guillaume Lampe*
Facebook AI Research
glample@fb.com

François Charton*
Facebook AI Research
fcharton@fb.com

ABSTRACT

Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica.



Lample et al. - 2019

Formal mathematics statement curriculum learning

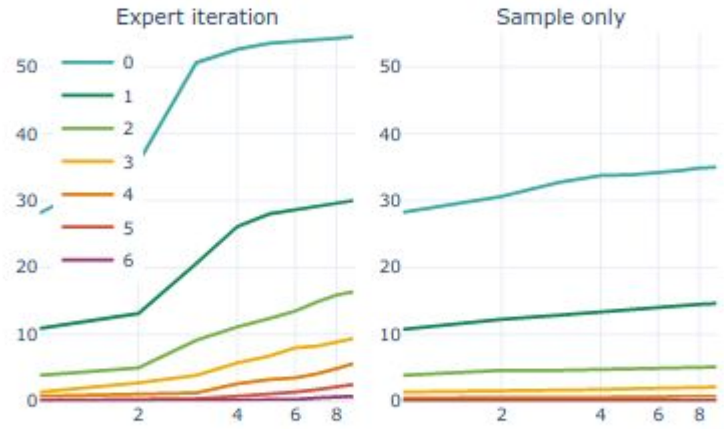


Figure 3. Cumulative pass rate for our *expert iteration* loop as well as a *sample only* loop where we skip re-training the model between iterations. Pass rates are reported for each value of N_D (pooling together $0 \leq N_S \leq 7$).

- Expert iteration
- Automated curriculum learning
- Outperform proof search only

Polu, Stanislas, et al. - 2022

Li et al. - Science - 2022

Alphacode: a hybrid

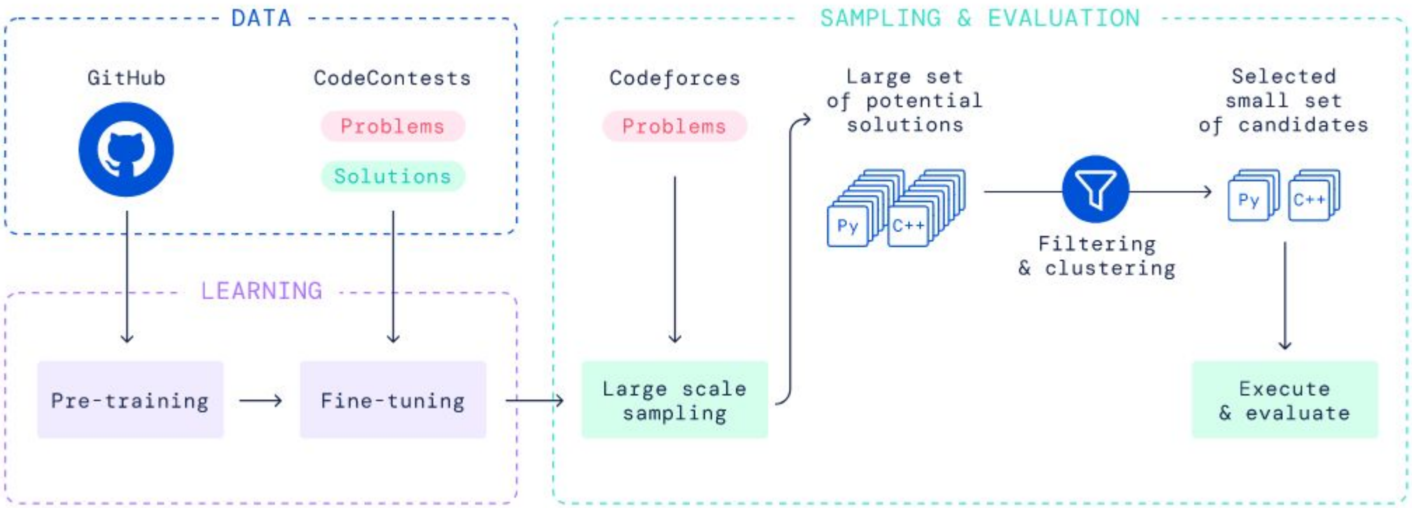


Figure 4 | **Overview of AlphaCode.**

Li et al. - Science - 2022

AlphaTensor

Article

Discovering faster matrix multiplication algorithms with reinforcement learning

https://doi.org/10.1038/s41586-022-05172-4

Received: 2 October 2021

Accepted: 2 August 2022

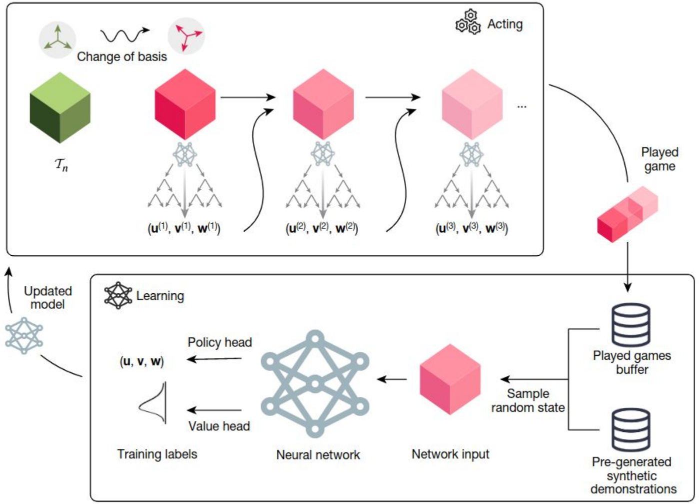
Published online: 5 October 2022

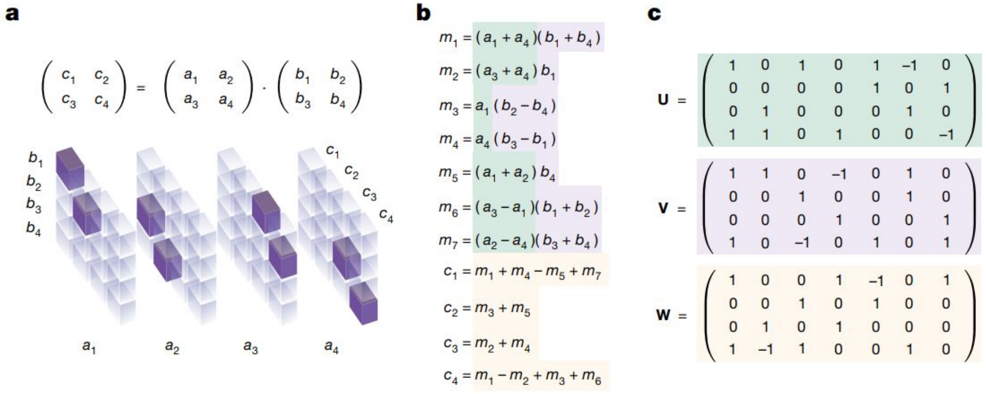
Alhussein Fawzi[1,2]✉, Matej Balog[1,2], Aja Huang[1,2], Thomas Hubert[1,2], Bernardino Romera-Paredes[1,2], Mohammadamin Barekatain[1], Alexander Novikov[1], Francisco J. R. Ruiz[1], Julian Schrittwieser[1], Grzegorz Swirszcz[1], David Silver[1], Demis Hassabis[1] & Pushmeet Kohli[1]

Fawzi et al. - Nature - 2022

AlphaTensor

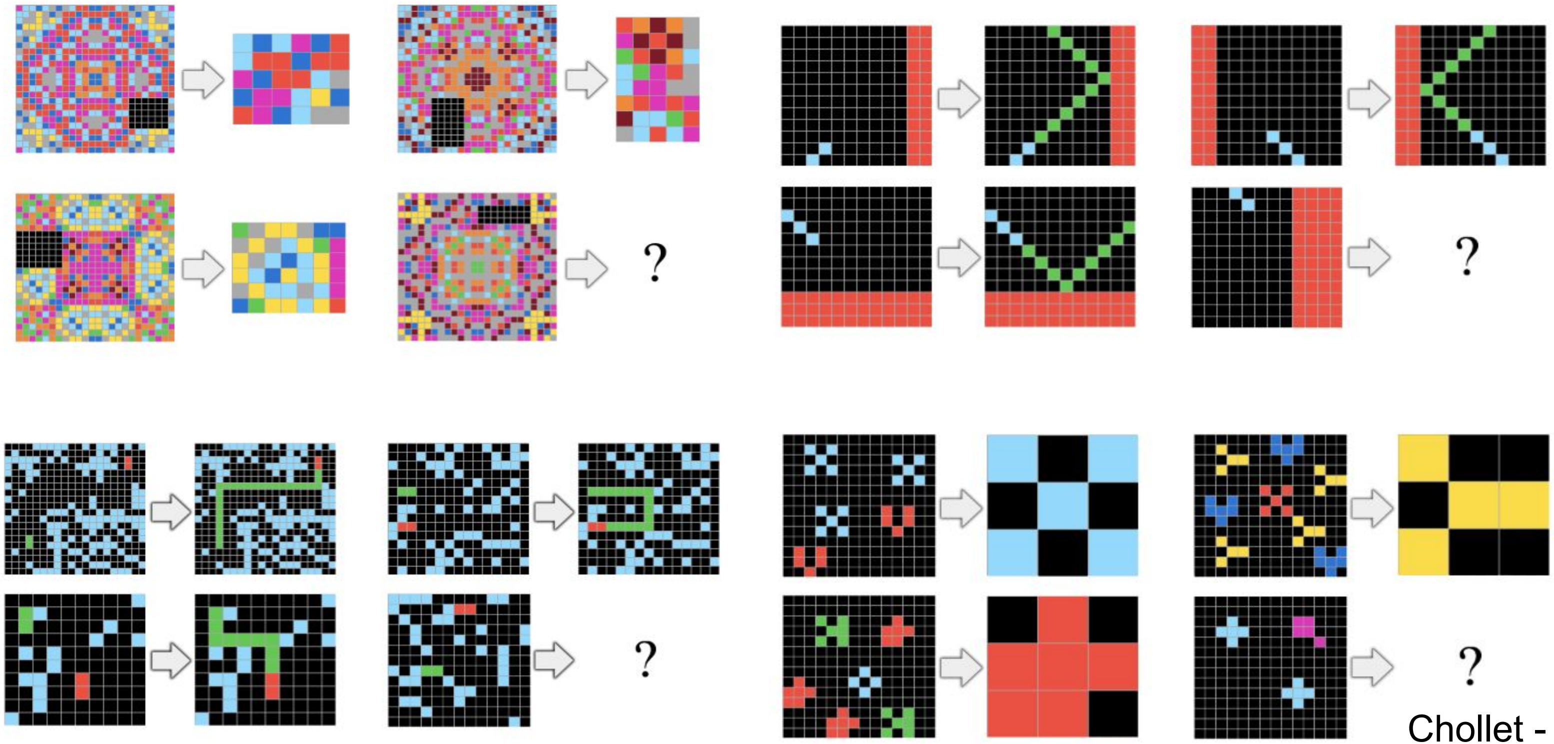
- RL + MCTS (AlphaZero)
- Synthetic data
- Is just program synthesis





Fawzi et al. - Nature - 2022

Abstraction and Reasoning Challenge (ARC)



Chollet - 2019

Abstraction and Reasoning Challenge (ARC)

- Only uses human priors (arithmetic, objectness, geometry, …)
- Meta-learning (learning to search for solution)
- Developer-aware generalization (limiting the use of symbolic methods)
- The way we measure machine intelligence is wrong
- Intelligence is data efficient
- DL struggles on ARC (but so do purely symbolic method)
- Applicable methods: dreaming, REPL, verifying solutions

Chollet - 2019

Thank you!