Machine Learning and Symbolic Computation
Part I of II

Matthew England
Coventry University

Thematic Seminar on Symbolic Computation
Shenzhen, China
5 – 12 January 2025

Speaker supported by EPSRC grant EP/T015748/1.

Outline for the Course

Session 1:

1. Introduction
2. Case Study 1:
   ML for Symbolic Integration Algorithm Selection in Maple

Session 2:

3. Case Study 2:
   ML to Optimise Cylindrical Algebraic Decomposition
4. Future Progress from Explainable AI?

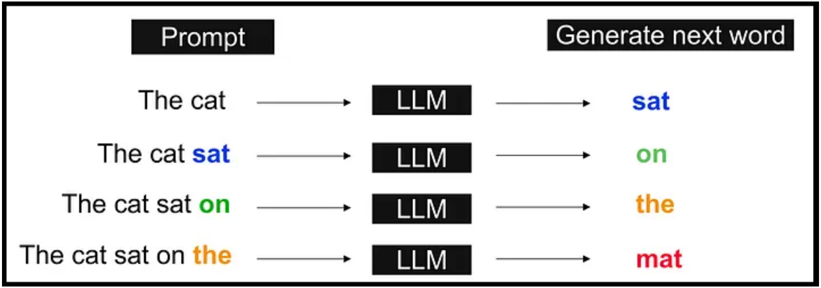
Includes joint work with: Rashid Barket, James Bridge, Kelly Cohen, James Davenport, Tereso del Río, Dorian Florescu, Zongyan Huang, Larry Paulson, Lynn Pickering.

Outline

Outline

LLMs

Recall how LLMs work: next word prediction following training on copious amounts of data.



LLMs are incredible feats of engineering resting on the powerful computer science idea of the *transformer* architecture to process big data in parallel. But can they perform mathematics yet?

Chain-of-Thought LLMs

- **Original idea:** Use prompt engineering to have the model show its working rather than going directly to the answer.
- **2024 OpenAI o1 model:** uses reinforcement learning to train GPT to use a chain-of-thought process.
    - The internal chain is not trained for "*policy compliance or user preferences*" in the way the actual output is.
    - The actual chain of thought is kept internal and not revealed.

Wei, Wang, Schuurmans, Bosma, Ichter, Xia, Chi, Le, Zhou (Google).
Chain-of-thought prompting elicits reasoning in large language models
Proc. NIPS 2022, article 1800
https://openreview.net/forum?id=_VjQlMeSB_J

OpenAI
Learning to Reason with LLMs.
https://openai.com/index/learning-to-reason-with-llms/

How good are LLMs at basic mathematics? (Slide 3/45)

OpenAI: "*o1 ranks in the 89th percentile on competitive programming questions and places among the top 500 students in the US in a qualifier for the USA Math Olympiad*. However:

Mirzadeh, Alizadeh, Shahrokhi, Tuzel, Bengio, Farajtabar (Apple)
GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.
Under Review for ICLR 2025. Preprint:
https://arxiv.org/abs/2410.05229

Creates variations of the standard benchmarks (e.g. changing the numbers / names) which reveals huge performance drops in most LLMs; only modest ones for o1.

Reports that even o1 drops off in the face of irrelevant information; but the examples in the paper can now be handled when I try!

How good are LLMs at basic mathematics? (Slide 3/45)

OpenAI: "*o1 ranks in the 89th percentile on competitive programming questions and places among the top 500 students in the US in a qualifier for the USA Math Olympiad*. However:

Mirzadeh, Alizadeh, Shahrokhi, Tuzel, Bengio, Farajtabar (Apple)
GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.
Under Review for ICLR 2025. Preprint:
https://arxiv.org/abs/2410.05229

Creates variations of the standard benchmarks (e.g. changing the numbers / names) which reveals huge performance drops in most LLMs; only modest ones for o1.

Reports that even o1 drops off in the face of irrelevant information; but the examples in the paper can now be handled when I try!

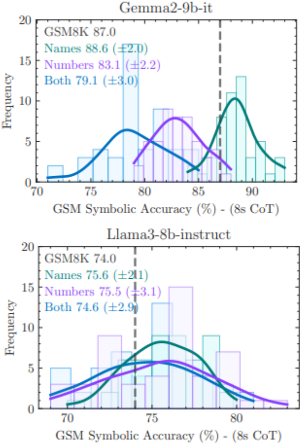
2024 GSM-Symbolic Evaluation

GSM Symbolic Template

```
When {name} watches her {family}, she gets out a variety
of toys for him.  The bag of building blocks has {x}
blocks in it.  The bin of stuffed animals has {y} stuffed
animals inside.The tower of stacking rings has {z}
multicolored rings on it.{name} recently bought a tube
of bouncy balls, bringing her total number of toys she
bought for her {family} up to {total}.  How many bouncy
balls came in the tube?

#variables:
- name = sample(names)
- family = sample(["nephew", "cousin", "brother"])
- x = range(5, 100)
- y = range(5, 100)
- z = range(5, 100)
- total = range(100, 500)
- ans = range(85, 200)

#conditions:
- x + y + z + ans == total
```



2024 GSM-Symbolic Evaluation

GSM-NoOp

Oliver picks 44 kiwis on Friday. Then he picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, but five of them were a bit smaller than average. How many kiwis does Oliver have?

---

o1-mini: Oliver picks kiwis over three days as follows:
- Friday: 44 kiwis
- Saturday: 58 kiwis
- Sunday: Double the number he picked on Friday, which is $2 \times 44 = 88$ kiwis

However, on Sunday, 5 of these kiwis were smaller than average. We need to subtract them from the Sunday total: 88 (Sunday's kiwis) - 5 (smaller kiwis) = 83 kiwis Now, summing up the kiwis from all three days: 44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis Oliver has a total of 185 kiwis.

GSM-NoOp

Liam wants to buy some school supplies. He buys 24 erasers that now cost \$6.75 each, 10 notebooks that now cost \$11.0 each, and a ream of bond paper that now costs \$19. How much should Liam pay now, assuming that due to inflation, prices were 10% cheaper last year?

---

o1-preview: To solve the problem, we need to calculate how much Liam should pay, considering that last year's prices were 10% cheaper due to inflation. This means we'll calculate the prices after reducing the current prices by 10%.

Step 1: Calculate last year's prices by reducing the current prices by 10%.
- Eraser price last year: $\$6.75 \times 0.90 = \$6.075$
- Notebook price last year: $\$11.00 \times 0.90 = \$9.90$
- Bond paper price last year: $\$19.00 \times 0.90 = \$17.10$

Step 2: Calculate the total cost using last year's prices.
- Total cost for erasers: 24 erasers $\times \$6.075 = \$145.80$
- Total cost for notebooks: 10 notebooks $\times \$9.90 = \$99.00$
- Total cost for bond paper: 1 ream $\times \$17.10 = \$17.10$

Step 3: Sum up all the costs to get the total amount Liam should pay.
$\$145.80 + \$99.00 + \$17.10 = \$261.90$

Answer:\$261.90

Will LLMs do Mathematics in the Future?

Hard to predict. It seems they will be able to do arithmetic and logic reasoning as we teach humans to do by hand in school.

But can they do symbolic computation?

LLM to Create a Gröbner Basis I

Copilot
Please calculate a Groebner Ba...

December 10, 2024 at 12:43 PM

Please calculate a Groebner Basis for the set of polynomials {x^2 - 2*x*z + 5, y*z^3 + x*y^2, -8*z^3 + 3*y^2}

I calculated the Groebner Basis for the given set of polynomials ({x^2 - 2x z + 5, y z^3 + x y^2, -8 z^3 + 3 y^2}). Here is the result:

$$ \begin{align*}

1. & \quad x^2 - 2x z + 5 \
2. & \quad 640x z^3 + 9z^8 - 96z^7 + 120z^5 \
3. & \quad 3y^2 - 8z^3 \
4. & \quad 80y z^3 - 3z^8 + 32z^7 - 40z^5 \
5. & \quad 9z^9 - 96z^8 + 240z^6 + 1600z^3 \end{align*} $$

If you have any further questions or need additional assistance, feel free to ask!

Edit in Pages | Code | ···

1 of 30 responses | AI-generated content may be incorrect

LLM to Create a Gröbner Basis 2

How did you calculate this?

Copilot
Please calculate a Groebner Ba...

I used the **SymPy** library in Python to calculate the Groebner Basis. SymPy is a powerful tool for symbolic mathematics, and it includes functions for working with polynomials and computing Groebner Bases.

Here's a step-by-step outline of the process:

1. **Define the Variables**: I defined the variables (x), (y), and (z) using SymPy's `symbols` function.
2. **Define the Polynomials**: I created a list of the given polynomials.
3. **Calculate the Groebner Basis**: I used the `groebner` function from SymPy, specifying the polynomials and the variable order (lexicographic by default).
4. **Output the Result**: I printed the resulting Groebner Basis.

Here's the Python code I used:

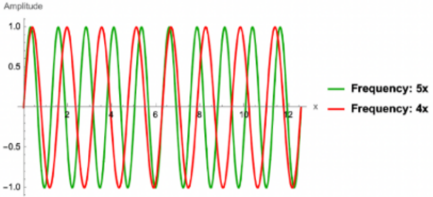
```
from sympy import symbols, groebner
# Define the variables
x, y, z = symbols('x y z')
# Define the polynomials
polynomials = [x**2 - 2*x*z + 5, y*z**3 + x*y**2, -8*z**3 + 3*y**2]
# Calculate the Groebner basis
G = groebner(polynomials, x, y, z)
# Print the Groebner basis
print("The Groebner basis for the given set of polynomials is:")
for g in G:
    print(g)
```

Mathematica 14: LLM Integration Example 1 (Slide 8/45)

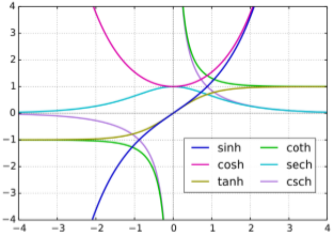
Plot two sine curves with a frequency ratio corresponding to a perfect third. Show them in the colors of the Portuguese flag.

A perfect third in music consists of two frequencies with a ratio of 5:4. To plot two sine curves with this frequency ratio and the colors of the Portuguese flag (green and red), you can use the following code:

```
Plot[
 {Sin[5 x], Sin[4 x]},
 {x, 0, 4 Pi},
 PlotStyle → {Darker[Green], Red},
 AxesLabel → {"x", "Amplitude"},
 PlotLegends → {"Frequency: 5x", "Frequency: 4x"}
]
```



Mathematica 14: LLM Integration Example 2



Create a plot that looks similar to that

```
Plot[
  {Sinh[x], Cosh[x], Tanh[x], Coth[x], Sech[x], Csch[x]},
  {x, -4, 4},
  PlotLegends → "Expressions",
  PlotRange → {-4, 4},
  AxesLabel → {"x", "y"}
]
```

LLMs and Computer Algebra

It looks unlikely that LLMs will replace symbolic computation any time soon. Rather, they seek to work in collaboration with CASs.

S. Wolfram
WolframAlpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT.
https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/

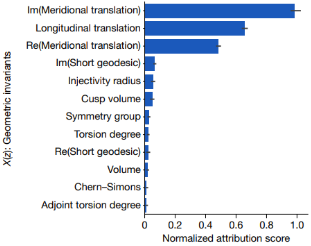
S. Wolfram
New in 14: LLM & AI .
https://blog.wolfram.com/2024/01/16/new-in-14-llm-ai/

LLMs and CASs complement each other. But is there a role for other machine learning technology in advanced mathematics and programming? We give a quick survey.

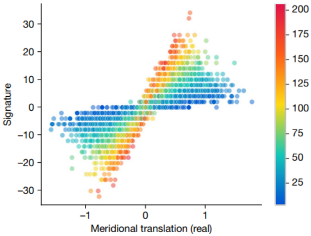
Machine Learning Pure Mathematics (Slide 11/45)

There have been a variety of papers suggesting that ML can guide pure mathematicians to potential conjectures. E.g.

Davies, Veličković, Buesing, Blackwell, Zheng, Tomašev, Tanburn, Battaglia, Blundell, Juhász, Lackenby, Williamson, Hassabis, Kohli.
Advancing mathematics by guiding human intuition with AI.
Nature, 600, pp. 70––74, 2021.
https://doi.org/10.1038/s41586-021-04086-x



Attribution values for geometric invariant features.



The real part of the meridional translation against signature, coloured by the longitudinal translation

AlphaGeometry and AlphaProof

`AlphaGeometry` is a tool for Euclidean plane geometry:

- Uses LLM trained on large-scale synthetic data, to guide a symbolic deduction engine.
- Evaluated on 30 recent Olympiad-level problems: solved 25; approaching IMO gold medallist level.
- Produces human-readable proofs.

Trinh, Wu, Le, He, Luong (Google DeepMind).
Solving Olympiad geometry without human demonstrations.
Nature, 625, pp. 476––482, 2024.
https://doi.org/10.1038/s41586-023-06747-5

There also exists `AlphaProof`: a system that trains itself to prove mathematical statements in the formal language Lean.

AlphaGeometry

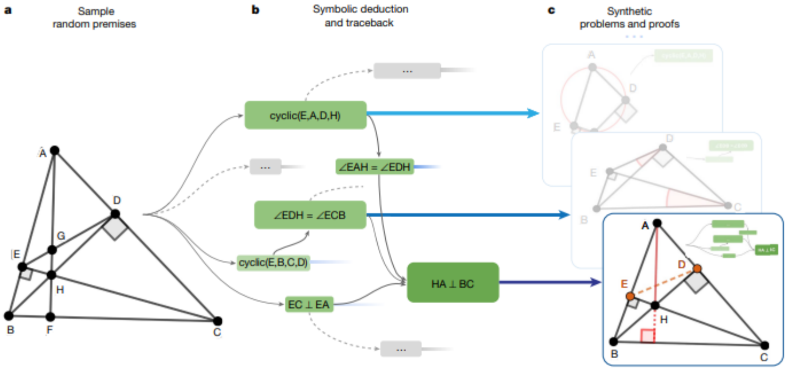


Fig. 3 | AlphaGeometry synthetic-data-generation process. a, We first sample a large set of random theorem premises. **b**, We use the symbolic deduction engine to obtain a deduction closure. This returns a directed acyclic graph of statements. For each node in the graph, we perform traceback to find its minimal set of necessary premise and dependency deductions. For example, for the rightmost node 'HA ⊥ BC', traceback returns the green subgraph. **c**, The minimal premise and the corresponding subgraph constitute a synthetic problem and its solution. In the bottom example, points E and D took part in the proof despite being irrelevant to the construction of HA and BC; therefore, they are learned by the language model as auxiliary constructions.

Machine Learning Matrix Multiplication (Slide 14/45)

Reinforcement learning found an improvement on Strassen's algorithm to multiply matrices:

A. Fawzi and M. Balog and A. Huang and T. Hubert and B. Romera-Paredes and M. Barekatain and A. Novikov and F.J.R. Ruiz and J. Schrittwieser and G. Swirszcz and D. Silver and D. Hassabis and P. Kohli.
Discovering faster matrix multiplication algorithms with reinforcement learning.
Nature, 610, pp. 47—–53, 2022.
https://doi.org/10.1038/s41586-022-05172-4

That solution seeded a new random walk based algorithm which found an even better one!

Manuel Kauers and Jakob Moosbauer.
Flip Graphs for Matrix Multiplication.
Proc. ISSAC 2023, pp. 381 – –388. ACM, 2023.
https://doi.org/10.1145/3597066.3597120

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} \end{pmatrix}$$

$h_1 = a_{3,2}(-b_{2,1} - b_{2,5} - b_{3,1})$
$h_2 = (a_{2,2} + a_{2,5} - a_{3,5})(-b_{2,5} - b_{5,1})$
$h_3 = (-a_{3,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,5})$
$h_4 = (a_{1,2} + a_{1,4} + a_{3,4})(-b_{2,5} - b_{4,1})$
$h_5 = (a_{1,5} + a_{2,2} + a_{2,5})(-b_{2,4} + b_{5,1})$
$h_6 = (-a_{2,2} - a_{2,5} - a_{4,5})(b_{2,3} + b_{5,1})$
$h_7 = (-a_{1,1} + a_{4,1} - a_{4,2})(b_{1,1} + b_{2,4})$
$h_8 = (a_{3,2} - a_{3,3} - a_{4,3})(-b_{2,3} + b_{3,1})$
$h_9 = (-a_{1,2} - a_{1,4} + a_{4,4})(b_{2,3} + b_{4,1})$
$h_{10} = (a_{2,2} + a_{2,5})b_{5,1}$
$h_{11} = (-a_{2,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,2})$
$h_{12} = (a_{4,1} - a_{4,2})b_{1,1}$
$h_{13} = (a_{1,2} + a_{1,4} + a_{2,4})(b_{2,2} + b_{4,1})$
$h_{14} = (a_{1,3} - a_{3,2} + a_{3,3})(b_{2,4} + b_{3,1})$
$h_{15} = (-a_{1,2} - a_{1,4})b_{4,1}$
$h_{16} = (-a_{3,2} + a_{3,3})b_{3,1}$
$h_{17} = (a_{1,2} + a_{1,4} - a_{2,1} + a_{2,2} - a_{2,3} + a_{2,4} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,2}$
$h_{18} = a_{2,1}(b_{1,1} + b_{1,2} + b_{5,2})$
$h_{19} = -a_{2,3}(b_{3,1} + b_{3,2} + b_{5,2})$
$h_{20} = (-a_{1,5} + a_{2,1} + a_{2,3} - a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4} - b_{5,2})$
$h_{21} = (a_{2,1} + a_{2,3} - a_{2,5})b_{5,2}$
$h_{22} = (a_{1,3} - a_{1,4} - a_{2,4})(b_{1,1} + b_{1,2} - b_{1,4} - b_{3,1} - b_{3,2} + b_{3,4} + b_{4,4})$
$h_{23} = a_{1,3}(-b_{3,1} + b_{3,4} + b_{4,4})$
$h_{24} = a_{1,5}(-b_{4,4} - b_{5,1} + b_{5,4})$
$h_{25} = -a_{1,1}(b_{1,1} - b_{1,4})$
$h_{26} = (-a_{1,3} + a_{1,4} + a_{1,5})b_{4,4}$
$h_{27} = (a_{1,3} - a_{3,1} + a_{3,3})(b_{1,1} - b_{1,4} + b_{1,5} + b_{3,5})$
$h_{28} = -a_{3,4}(-b_{3,5} - b_{4,1} - b_{4,5})$
$h_{29} = a_{3,1}(b_{1,1} + b_{1,5} + b_{3,5})$
$h_{30} = (a_{3,1} - a_{3,3} + a_{3,4})b_{3,5}$
$h_{31} = (-a_{1,4} - a_{1,5} - a_{3,4})(-b_{4,4} - b_{5,1} + b_{5,4} - b_{5,5})$
$h_{32} = (a_{2,1} + a_{4,1} + a_{4,4})(b_{1,3} - b_{4,1} - b_{4,2} - b_{4,3})$
$h_{33} = a_{4,3}(-b_{3,1} - b_{3,3})$
$h_{34} = a_{4,4}(-b_{1,3} + b_{4,1} + b_{4,3})$
$h_{35} = -a_{4,5}(b_{1,3} + b_{5,1} + b_{5,3})$
$h_{36} = (a_{2,3} - a_{2,5} - a_{4,5})(b_{3,1} + b_{3,2} + b_{3,3} + b_{5,2})$
$h_{37} = (-a_{4,1} - a_{4,4} + a_{4,5})b_{1,3}$
$h_{38} = (-a_{2,3} - a_{3,1} + a_{3,3} - a_{3,4})(b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5})$
$h_{39} = (-a_{3,1} - a_{4,1} - a_{4,4} + a_{4,5})(b_{1,3} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{40} = (-a_{1,3} + a_{1,4} + a_{1,5} - a_{4,4})(-b_{3,1} - b_{3,3} + b_{3,4} + b_{4,4})$
$h_{41} = (-a_{1,1} + a_{4,1} - a_{4,5})(b_{1,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4})$
$h_{42} = (-a_{2,1} + a_{2,5} - a_{3,5})(-b_{1,1} - b_{1,2} - b_{1,5} + b_{4,1} + b_{4,2} + b_{4,5} - b_{5,2})$
$h_{43} = a_{2,4}(b_{4,1} + b_{4,2})$
$h_{44} = (a_{2,3} + a_{3,2} - a_{3,3})(b_{2,2} - b_{3,1})$
$h_{45} = (-a_{3,3} + a_{3,4} - a_{4,3})(b_{3,5} + b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{46} = -a_{3,5}(-b_{5,1} - b_{5,5})$
$h_{47} = (a_{2,1} - a_{2,5} - a_{3,1} + a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{4,1} - b_{4,2} - b_{4,5})$
$h_{48} = (-a_{2,3} + a_{3,3})(b_{2,2} + b_{3,2} + b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5})$
$h_{49} = (-a_{1,1} - a_{1,3} + a_{1,4} + a_{1,5} - a_{2,1} - a_{2,3} + a_{2,4} + a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4})$
$h_{50} = (-a_{1,4} - a_{2,4})(b_{2,2} - b_{3,1} - b_{3,2} + b_{3,4} - b_{4,2} + b_{4,4})$
$h_{51} = a_{2,2}(b_{2,1} + b_{2,2} - b_{5,1})$
$h_{52} = a_{4,2}(b_{1,1} + b_{2,1} + b_{2,3})$
$h_{53} = -a_{1,2}(-b_{2,1} + b_{2,4} + b_{4,1})$
$h_{54} = (a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,2} + a_{4,3} - a_{4,4} - a_{4,5})b_{2,3}$
$h_{55} = (a_{1,4} - a_{4,4})(-b_{2,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{4,3} - b_{4,4})$
$h_{56} = (a_{1,1} - a_{1,5} - a_{4,1} + a_{4,5})(b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4})$
$h_{57} = (-a_{3,1} - a_{4,1})(-b_{1,3} - b_{1,5} - b_{2,5} - b_{5,1} - b_{5,3} - b_{5,5})$
$h_{58} = (-a_{1,4} - a_{1,5} - a_{3,4} - a_{3,5})(-b_{5,1} + b_{5,4} - b_{5,5})$
$h_{59} = (-a_{3,3} + a_{3,4} - a_{4,3} + a_{4,4})(b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{60} = (a_{2,5} + a_{4,5})(b_{2,3} - b_{3,1} - b_{3,2} - b_{3,3} - b_{5,2} - b_{5,3})$
$h_{61} = (a_{1,4} + a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5} - b_{2,5} - b_{4,4} + b_{4,5} - b_{5,1} + b_{5,4} - b_{5,5})$
$h_{62} = (a_{2,1} + a_{4,1})(b_{1,2} + b_{1,3} + b_{2,2} - b_{4,1} - b_{4,2} - b_{4,3})$
$h_{63} = (-a_{3,3} - a_{4,3})(-b_{2,3} - b_{3,3} - b_{3,5} - b_{4,1} - b_{4,3} - b_{4,5})$
$h_{64} = (a_{1,1} - a_{1,3} - a_{1,4} + a_{3,1} - a_{3,3} - a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5})$
$h_{65} = (-a_{1,1} + a_{4,1})(-b_{1,3} + b_{1,4} + b_{2,4} - b_{5,1} - b_{5,3} + b_{5,4})$
$h_{66} = (a_{1,1} - a_{1,2} + a_{1,3} - a_{1,5} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,4}$
$h_{67} = (a_{2,5} - a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{2,5} - b_{4,1} - b_{4,2} - b_{4,5} + b_{5,2} + b_{5,5})$
$h_{68} = (a_{1,1} + a_{1,3} - a_{1,4} - a_{1,5} - a_{4,1} - a_{4,3} + a_{4,4} + a_{4,5})(-b_{3,1} - b_{3,3} + b_{3,4})$
$h_{69} = (-a_{1,3} + a_{1,4} - a_{2,3} + a_{2,4})(-b_{2,4} - b_{3,1} - b_{3,2} + b_{3,4} - b_{5,2} + b_{5,4})$
$h_{70} = (a_{2,3} - a_{2,5} + a_{4,3} - a_{4,5})(-b_{3,1} - b_{3,2} - b_{3,3})$
$h_{71} = (-a_{3,1} + a_{3,3} - a_{3,4} + a_{3,5} - a_{4,1} + a_{4,3} - a_{4,4} + a_{4,5})(-b_{5,1} - b_{5,3} - b_{5,5})$
$h_{72} = (-a_{2,1} - a_{2,4} - a_{4,1} - a_{4,4})(b_{4,1} + b_{4,2} + b_{4,3})$
$h_{73} = (a_{1,3} - a_{1,4} - a_{1,5} + a_{2,3} - a_{2,4} - a_{2,5})(b_{1,1} + b_{1,2} - b_{1,4} + b_{2,4} + b_{5,2} - b_{5,4})$
$h_{74} = (a_{2,1} - a_{2,3} + a_{2,4} - a_{3,1} + a_{3,3} - a_{3,4})(b_{4,1} + b_{4,2} + b_{4,5})$
$h_{75} = -(a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,1} + a_{3,2} + a_{3,4} + a_{3,5} - a_{4,1} + a_{4,2})b_{2,5}$
$h_{76} = (a_{1,3} + a_{3,3})(-b_{1,1} + b_{1,4} - b_{1,5} + b_{2,4} + b_{3,4} - b_{3,5})$
$c_{1,1} = -h_{10} + h_{12} + h_{14} - h_{15} - h_{16} + h_{53} + h_5 - h_{66} - h_7$
$c_{2,1} = h_{10} + h_{11} - h_{12} + h_{13} + h_{15} + h_{16} - h_{17} - h_{44} + h_{51}$
$c_{3,1} = h_{10} - h_{12} + h_{15} + h_{16} - h_1 + h_2 + h_3 - h_4 + h_{75}$
$c_{4,1} = -h_{10} + h_{12} - h_{15} - h_{16} + h_{52} + h_{54} - h_6 - h_8 + h_9$
$c_{1,2} = h_{13} + h_{15} + h_{20} + h_{21} - h_{22} + h_{23} + h_{25} - h_{43} + h_{49} + h_{50}$
$c_{2,2} = -h_{11} + h_{12} - h_{13} - h_{15} - h_{16} + h_{17} + h_{18} - h_{19} - h_{21} + h_{43} + h_{44}$
$c_{3,2} = -h_{16} - h_{19} - h_{21} - h_{28} - h_{29} - h_{38} + h_{42} + h_{44} - h_{47} + h_{48}$
$c_{4,2} = h_{11} - h_{12} - h_{18} + h_{21} - h_{32} + h_{33} - h_{34} - h_{36} + h_{62} - h_{70}$
$c_{1,3} = h_{15} + h_{23} + h_{24} + h_{34} - h_{37} + h_{40} - h_{41} + h_{55} - h_{56} - h_9$
$c_{2,3} = -h_{10} + h_{19} + h_{32} + h_{35} + h_{36} + h_{37} - h_{43} - h_{60} - h_6 - h_{72}$
$c_{3,3} = -h_{16} - h_{28} + h_{33} + h_{37} - h_{39} + h_{45} - h_{46} + h_{63} - h_{71} - h_8$
$c_{4,3} = h_{10} + h_{15} + h_{16} - h_{33} + h_{34} - h_{35} - h_{37} - h_{54} + h_6 + h_8 - h_9$
$c_{1,4} = -h_{10} + h_{12} + h_{14} - h_{16} + h_{23} + h_{24} + h_{25} + h_{26} + h_5 - h_{66} - h_7$
$c_{2,4} = h_{10} + h_{18} - h_{19} + h_{20} - h_{22} - h_{24} - h_{26} - h_5 - h_{69} + h_{73}$
$c_{3,4} = -h_{14} + h_{16} - h_{23} - h_{26} + h_{27} + h_{29} + h_{31} + h_{46} - h_{58} + h_{76}$
$c_{4,4} = h_{12} + h_{25} + h_{26} - h_{33} - h_{35} - h_{40} + h_{41} + h_{65} - h_{68} - h_7$
$c_{1,5} = h_{15} + h_{24} + h_{25} + h_{27} - h_{28} + h_{30} + h_{31} - h_4 + h_{61} + h_{64}$
$c_{2,5} = -h_{10} - h_{18} - h_2 - h_{30} - h_{38} + h_{42} - h_{43} + h_{46} + h_{67} + h_{74}$
$c_{3,5} = -h_{10} + h_{12} - h_{15} + h_{28} + h_{29} - h_2 - h_{30} - h_3 + h_{46} + h_4 - h_{75}$
$c_{4,5} = -h_{12} - h_{29} + h_{30} - h_{34} + h_{35} + h_{39} + h_3 - h_{45} + h_{57} + h_{59}$

Extended Data Fig. 2 | Algorithm for multiplying 4 × 5 by 5 × 5 matrices in standard arithmetic with 76 multiplications. This outperforms the previously best known algorithm, which involves 80 multiplications.

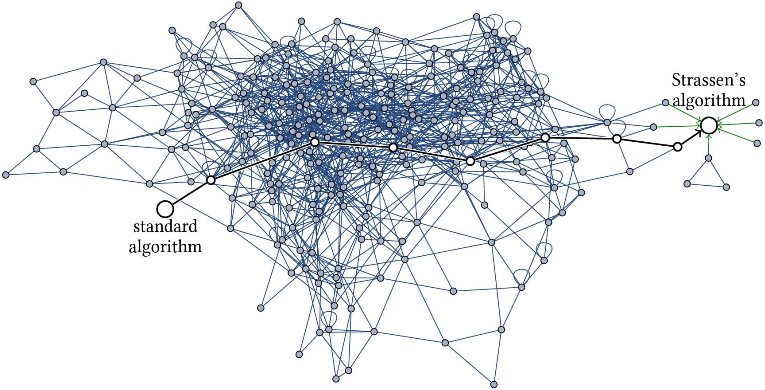
Machine Learning Matrix Multiplication

Reinforcement learning found an improvement on Strassen's algorithm to multiply matrices:

A. Fawzi and M. Balog and A. Huang and T. Hubert and B. Romera-Paredes and M. Barekatain and A. Novikov and F.J.R. Ruiz and J. Schrittwieser and G. Swirszcz and D. Silver and D. Hassabis and P. Kohli.
Discovering faster matrix multiplication algorithms with reinforcement learning.
Nature, 610, pp. 47——53, 2022.
https://doi.org/10.1038/s41586-022-05172-4

That solution seeded a new random walk based algorithm which found an even better one!

Manuel Kauers and Jakob Moosbauer.
Flip Graphs for Matrix Multiplication.
Proc. ISSAC 2023, pp. 381 – –388. ACM, 2023.
https://doi.org/10.1145/3597066.3597120

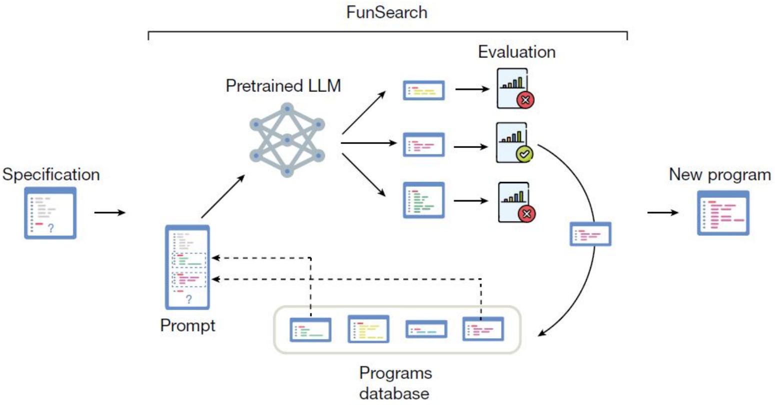

Strassen's
algorithm
standard
algorithm

LLMs to Evolve Heuristic Algorithms

Romera-Paredes, Barekatain, Novikov, Balog, Kumar, Dupont, Ruiz, Ellenberg, Wang, Fawzi, Kohli, Fawzi (DeepMind).
Mathematical discoveries from program search with large language models.
Nature, 625, pp. 468––475, 2023.
https://doi.org/10.1038/s41586-023-06924-6

They start with a database of heuristic methods for a hard problem; and evolve them by feeding two at a time as a prompt to an LLM (Codey) which is asked to merge them; with the outputs evaluated for fitness and the best added into the database.

Does not train the LLM themselves! The LLM is simply the merger tool. Note they breed programs to solve the problem (genetic programming), rather than breeding solutions (genetic algorithms).


FunSearch
Evaluation
Pretrained LLM
Specification
New program
Prompt
Programs database

LLMs to Evolve Heuristic Algorithms

Romera-Paredes, Barekatain, Novikov, Balog, Kumar, Dupont, Ruiz, Ellenberg, Wang, Fawzi, Kohli, Fawzi (DeepMind).
Mathematical discoveries from program search with large language models.
Nature, 625, pp. 468––475, 2023.
https://doi.org/10.1038/s41586-023-06924-6

They start with a database of heuristic methods for a hard problem; and evolve them by feeding two at a time as a prompt to an LLM (Codey) which is asked to merge them; with the outputs evaluated for fitness and the best added into the database.

Does not train the LLM themselves! The LLM is simply the merger tool. Note they reed programs to solve the problem (genetic programming), rather than breeding solutions (genetic algorithms).

Outline

1. Introduction
   - AI and Mathematics
   - Machine Learning and Symbolic Computation
2. ML for Symbolic Integration Algorithm Selection in Maple
   - Problem and Data Generation
   - ML Experiments

Symbolic Computation via Machine Learning I (Slide 16/45)

(Q) So can we train ML models to replace symbolic computation?

The first attempt was in 2020 which demonstrated transformers to perform symbolic integration and ODE solving.

G. Lample and D. Charton
Deep Learning for Symbolic Mathematics.
Proc. ICLR 2020.
https://doi.org/10.48550/arXiv.1912.01412

	Integration (BWD)	ODE (order 1)	ODE (order 2)
Mathematica (30s)	84.0	77.2	61.6
Matlab	65.2	-	-
Maple	67.4	-	-
Beam size 1	98.4	81.2	40.8
Beam size 10	99.6	94.0	73.2
Beam size 50	99.6	97.0	81.0

Table 3: **Comparison of our model with Mathematica, Maple and Matlab on a test set of 500 equations.** For Mathematica we report results by setting a timeout of 30 seconds per equation. On a given equation, our model typically finds the solution in less than a second.

Symbolic Computation via Machine Learning II (Slide 17/45)

More recently, there has been an attempt at transformers to directly produce Gröbner Bases.

H. Kera, Y. Ishihara, Y. Kambe, T. Vaccon and K. Yokoyama
Learning to compute Gröbner bases.
Proc. NIPS 2024.
https://openreview.net/forum?id=ZRz7XlxBzQ

Accuracy [%] / support accuracy [%] of Gröbner basis by Transformer

Coeff.		Shape position			
		$n = 2$	$n = 3$	$n = 4$	$n = 5$
$\mathbb{Q}$	*disc.*	93.7 / 95.4	88.7 / 92.0	90.8 / 94.0	86.5 / 90.6
	hyb.	66.8 / 87.3	69.0 / 89.8	62.7 / 86.8	0.0 / 84.9
$\mathbb{F}_7$	*disc.*	72.3 / 79.1	78.1 / 83.2	71.3 / 84.6	84.3 / 88.5
	hyb.	54.1 / 78.7	55.8 / 84.3	46.1 / 81.8	54.4 / 81.5
$\mathbb{F}_{31}$	*disc.*	46.8 / 77.3	50.2 / 80.9	51.1 / 83.7	28.6 / 77.9
	hyb.	6.1 / 75.3	5.8 / 80.4	0.1 / 73.0	0.1 / 76.9
$\mathbb{R}$	*hyb.*	57.2 / 85.0	61.0 / 88.0	61.7 / 87.5	45.6 / 82.9

How to combine Symbolic Computation and ML?

ML can only offer probabilistic guidance, but CASs prize exact results. So how to combine ML and CASs? Possibilities:

- Algorithms to validate correctness of ML results?
- Algorithms to fix "*close*" results from ML?
- Guarantees on "*in-distribution*" accuracy of ML?

Another option is to integrate the technologies. CASs often contain multiple symbolic computation algorithms for the same task to choose between (Case Study 1). Further, those individual algorithms often come with settings to choose in how they are run (Case Study 2). These do not effect the correctness of the final result, but do effect resources required and result presentation.

Hypothesis: ML can make such choices better than developers or users.

How to combine Symbolic Computation and ML?

ML can only offer probabilistic guidance, but CASs prize exact results. So how to combine ML and CASs? Possibilities:

- Algorithms to validate correctness of ML results?
- Algorithms to fix "*close*" results from ML?
- Guarantees on "*in-distribution*" accuracy of ML?

Another option is to integrate the technologies. CASs often contain multiple symbolic computation algorithms for the same task to choose between (Case Study 1). Further, those individual algorithms often come with settings to choose in how they are run (Case Study 2). These do not effect the correctness of the final result, but do effect resources required and result presentation.

Hypothesis: ML can make such choices better than developers or users.

Early Examples of ML to Optimise CASs

- **HE+14** was the first use of ML for computer algebra: used a support vector machine to select a CAS variable ordering.
- **KUV15** used a Monte-Carlo tree search to find the representation of polynomials that are most efficient to evaluate numerically.
- **SYK16** used ML to choose which algorithm to compute the resultant with for a given problem instance.
- **HE+16** used a support vector machine to decide whether or not to precondition a symbolic computation input.
- **BD20** used a neural network to select the order of polynomial constraints to process in their solver.
- **CZC20** uses a neural network for selecting a variable ordering.
- **PSH20** applied reinforcement learning to select which S-Pair to process next when building a Gröbner Basis.

Huang, England, Wilson, Dabenport, Paulson, Bridge.
Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition.
Proc. CICM '14, LNCS 8542, pp. 92——107, 2013.
http://dx.doi.org/10.1007/978-3-319-08434-3_8

Kuipers, Ueda, Vermaseren
Code optimization in FORM.
Computer Physics Communications, 189, 1-19, 2015.
https://doi.org/10.1016/j.cpc.2014.08.008

Simpson, Yi, Kalita.
Automatic Algorithm Selection in Computational Software Using Machine Learning.
Proc. ICMLA '16, pp. 355–360, 2016
https://doi.org/10.1109/ICMLA.2016.0064

Huang, England, Davenport, Paulson.
Using Machine Learning to decide when to Precondition Cylindrical Algebraic Decomposition with Groebner Bases.
Proc. SYNASC '16, pp. 45–52, 2016
https://doi.org/10.1109/SYNASC.2016.020

Brown, Daves.
Applying Machine Learning to Heuristics for Real Polynomial Constraint Solving.
Proc. ICMS '20, LNCS 12097, pp. 292—301, 2020
https://doi.org/10.1007/978-3-030-52200-1_29

Chen, Zhu, Chi.
Variable Ordering Selection for Cylindrical Algebraic Decomposition with Artificial Neural Networks.
Proc. ICMS '20, LNCS 12097, pp. 281—291, 2020
https://doi.org/10.1007/978-3-030-52200-1_28

Brown, Daves.
Learning Selection Strategies in Buchberger's Algorithm.
Proc. ML Research, 119, pp. 7575—7585, 2020.
https://proceedings.mlr.press/v119/peifer20a.html

Challenging / Interesting Machine Learning (Slide 20/45)

- Problems do not always neatly fit as regression / classification.
- No a priori limit on the input space.
- How best to embed the data for a ML algorithm
- Supervised learning hard: because labelling dataset needs lots of expensive symbolic computation.
- Unsupervised learning is hard: because it is unclear if a particular outcome is good or bad without seeing the rest.
- What constitutes a meaningful and representative data set?
- Insufficient quantities of real world data for deep learning.
- How to perform data augmentation / synthetic data generation to allow for generalisability on problems of interest?

A good area for research!

Outline

1. Introduction
   - AI and Mathematics
   - Machine Learning and Symbolic Computation

2. ML for Symbolic Integration Algorithm Selection in Maple
   - Problem and Data Generation
   - ML Experiments

Acknowledgements

Collaboration with Rashid Barket and Jürgen Gerhard.
Sponsored by Maplesoft.

Outline

1. Introduction
   - AI and Mathematics
   - Machine Learning and Symbolic Computation
2. ML for Symbolic Integration Algorithm Selection in Maple
   - Problem and Data Generation
   - ML Experiments

Indefinite integration in Maple

Symbolic indefinite integration is a key feature of CASs, including Maple where it accessed via the command `int`: a meta-algorithm which does pre-processing and table lookup before accessing a particular symbolic computation algorithms for integration, of which there are currently 12 to choose from.

> $f := 7x^3 + 3x^2 + 5x :$
> $\mathrm{int}(f, x)$

$$\frac{7}{4}x^4 + x^3 + \frac{5}{2}x^2$$

> $\mathrm{int}(\sin(x), x)$

$$-\cos(x)$$

> $\mathrm{int}\left(\frac{x}{x^3 - 1}, x\right)$

$$-\frac{\ln\left(x^2 + x + 1\right)}{6} + \frac{\sqrt{3}\arctan\left(\frac{(2x+1)\sqrt{3}}{3}\right)}{3} + \frac{\ln(x-1)}{3}$$

Integration of special functions

Maple has 100s of special functions implemented, many of which can appear in integrals.

$$> \operatorname{int}\left(\exp\left(-x^2\right), x\right)$$

$$\frac{\sqrt{\pi}\operatorname{erf}(x)}{2}$$

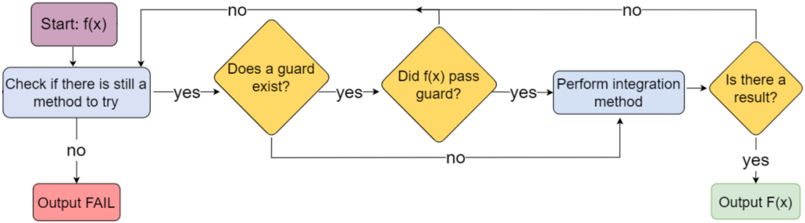
$$> \operatorname{int}\left(\frac{1}{\operatorname{sqrt}\left(2t^4-3t^2-2\right)}, t=2..3\right)$$

$$\frac{\sqrt{5}\operatorname{EllipticF}\left(\frac{\sqrt{7}}{3}, \frac{\sqrt{5}}{5}\right)}{5} - \frac{\sqrt{5}\operatorname{EllipticF}\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{5}}{5}\right)}{5}$$

This adds a layer of complexity which makes the direct ML solution difficult.

How `int` currently works (Slide 24/45)

The current default approach for `int` is to try all 12 sub-algorithms in a fixed order for an input until one of them succeeds. In some cases guard code allows for a method to be skipped early on.

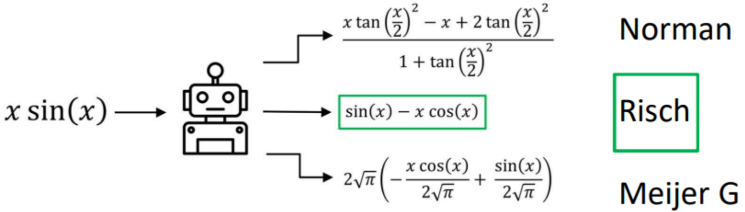


A high-level overview of how the indefinite integral command works in Maple to calculate $F(x) = \int f(x)dx$

Hypothesis: ML can make this more efficient by tailoring the `int` meta-algorithm flow to a particular problem instance.

Output Representation

There is clearly scope for computational savings from avoiding executing sub-algorithms that fail before identifying one that works. But we also aim to find algorithms that give "*nice looking*" answers.



Example from the dataset

The difference in answer sizes can be significant!

```
> f := 1 - x*cos(cos(x))*sin(x) + sin(cos(x))
```

$$f := 1 - x\cos(\cos(x))\sin(x) + \sin(\cos(x))$$

```
> # Maple's answer
  int(f, x)
```

$$\frac{x + x\tan\left(\frac{x}{2}\right)^2 + x\tan\left(\frac{1-\tan\left(\frac{x}{2}\right)^2}{2\left(1+\tan\left(\frac{x}{2}\right)^2\right)}\right)^2 + x\tan\left(\frac{x}{2}\right)^2\tan\left(\frac{1-\tan\left(\frac{x}{2}\right)^2}{2\left(1+\tan\left(\frac{x}{2}\right)^2\right)}\right)^2 + 2x\tan\left(\frac{1-\tan\left(\frac{x}{2}\right)^2}{2\left(1+\tan\left(\frac{x}{2}\right)^2\right)}\right) + 2x\tan\left(\frac{x}{2}\right)^2\tan\left(\frac{1-\tan\left(\frac{x}{2}\right)^2}{2\left(1+\tan\left(\frac{x}{2}\right)^2\right)}\right)}{\left(1+\tan\left(\frac{1-\tan\left(\frac{x}{2}\right)^2}{2\left(1+\tan\left(\frac{x}{2}\right)^2\right)}\right)^2\right)\left(1+\tan\left(\frac{x}{2}\right)^2\right)}$$

```
> # ML-suggested answer

  int(f, x, method=risch)
```

$$x + \sin(\cos(x))\,x$$

Data Generation

Maplesoft have an internal test suite developed over the years: $\sim 47k$ examples. Significant but not sufficient for deep learning. We use this as the validation dataset and seek to use an alternative – synthetic – dataset for testing.

We start with the same synthetic data as Lample and Charton:

G. Lample and D. Charton
Deep Learning for Symbolic Mathematics.
Proc. ICLR 2020.
https://doi.org/10.48550/arXiv.1912.01412

Lample & Carton (2020) Data Generation

Lample and Charton identified three approaches to generating synthetic (integrand, integral) pairs:

FWD: Integrate a random expression f using a CAS to get F and add (f, F) to dataset.

BWD: Differentiate a random expression f with a CAS to get f' and add (f', f) to the dataset.

IBP: Differentiate two random expressions f, g to get f', g'. Then if $\int f'g$ is "*known*" we may calculate

$$\int fg' := fg - \int f'g$$

and add the pair $(fg', fg - \int f'g)$ to the dataset.

Problems with Lample & Carton (2020) Data (Slide 29/45)

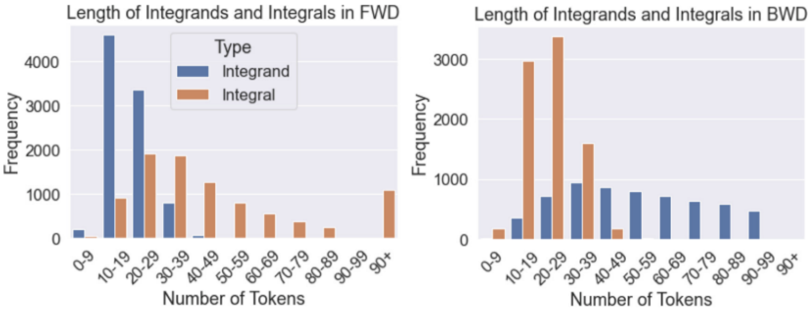
1. Potential Biases: The FWD dataset has integrands much shorter than integrals; the BWD dataset the opposite. Training one one dataset and testing on the other shows *very* poor accuracy. No independent validation dataset to test on.

2. Potential Data Leakage: There are lots of *very* similar expressions. Substituting all coefficients with a symbolic CONST token reduced the datasets to 35%, 75% and 24% of their sizes.

E. Davis.
The Use of Deep Learning for Symbolic Integration: A Review of [LC20]
https://arxiv.org/abs/1912.05752

B. Piotrowski, C. Brown, J. Urban and C. Kaliszyk.
Can Neural Networks Learn Symbolic Rewriting?.
In: Proc. AITP '19, 2019.
https://graphreason.github.io/papers/40.pdf

From Barket et al. (2023)



Problems with Lample & Carton (2020) Data (Slide 29/45)

1. Potential Biases: The FWD dataset has integrands much shorter than integrals; the BWD dataset the opposite. Training one one dataset and testing on the other shows *very* poor accuracy. No independent validation dataset to test on.

2. Potential Data Leakage: There are lots of *very* similar expressions. Substituting all coefficients with a symbolic CONST token reduced the datasets to 35%, 75% and 24% of their sizes.

E. Davis.
The Use of Deep Learning for Symbolic Integration: A Review of [LC20]
https://arxiv.org/abs/1912.05752

B. Piotrowski, C. Brown, J. Urban and C. Kaliszyk.
Can Neural Networks Learn Symbolic Rewriting?.
In: Proc. AITP '19, 2019.
https://graphreason.github.io/papers/40.pdf

New Data Generation Methods

We sought inspiration from classical results in computer algebra: the Liouville Theorem and the Risch Algorithm.

Theorem (Liouville's Theorem:). *Let D be a differential field with constant field K (whose algebraic closure is L). Suppose $f \in D$ and there exists g, elementary over D, such that $g' = f$. Then, $\exists\, v_0, \ldots, v_m \in D, c_1, \ldots, c_m \in L$ such that*

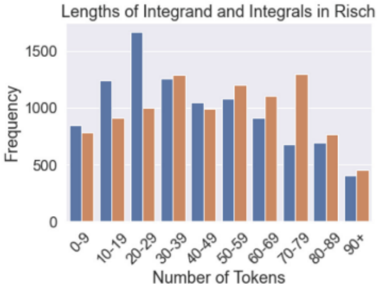
$$f = v_0' + \sum_{i=1}^{m} c_i \frac{v_i'}{v_i} \implies g = v_0 + \sum_{i=1}^{m} c_i \log(v_i).$$

Liouville's Theorem is proved by analytical means. The Risch algorithm actually construct the integrand: it doesn't always succeed, takes >100 pages to describe, and is not implemented in full; but is the basis of most CAS methods for integration.

R.H. Risch.
The Problem of Integration in Finite Terms
Transactions of the AMS, 139, pp. 167–189.
https://doi.org/10.1090/S0002-9947-1969-0237477-8

First New Attempt at Data Generation (Slide 31/45)

Basic idea: create random data; choose parameters to satisfy conditions in Risch Algorithm for integration to be possible (ensuring roots of the Trager-Rothstein resultant are constant).

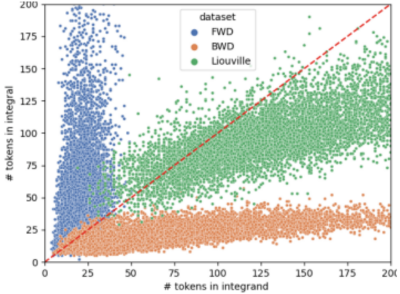


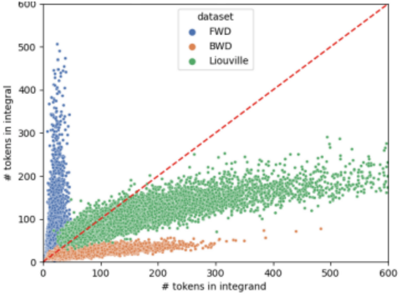
Problems: computationally expensive; relying on non-linear solvers; implementation limited to rational functions with denominator 2.

R. Barket, M. England, and J. Gerhard.
Generating Elementary Integrable Expressions
Proc. CASC '23, LNCS 14139, pp. 21 – 38, 2023.
https://doi.org/10.1007/978-3-031-41724-5_2

Second New Attempt at Data Generation (Slide 32/45)

Take inspiration from the "*Parallel Risch Algorithm*" which forms an integral hypothesis and then seeks coefficients. We create data similarly, with much lower computation costs.





R. Barket, M. England, and J. Gerhard.
The Liouville Generator for Producing Integrable Expressions
Proc. CASC '24, LNCS 14938, pp. 47 – 62, 2024.
https://doi.org/10.1007/978-3-031-69070-9_4

Our dataset in what follows

We use a large dataset drawn from all of the data generation methods. Although we hypothesise that the latter generator could actually be sufficient alone.

The examples contain constants up to three digits long. However, for ML training we:

- tokenize to extract minus signs separately;
- replace constants > 2 by symbolic tokens encoding only their number of digits: CONST1, CONST2, CONST3.
- Any duplicate examples after this replacement are discarded.

E.g. $2x^2 - 15x + 1$ becomes $2x^2 - \mathrm{CONST2}x + 1$ for ML training. This removes the data leakage fears from earlier.

Outline

Guarding integration methods

Task: train a binary classifier to predict an integration method will succeed on a particular input.

Aim: To potentially replace guard code for methods that have them; provide guards for those that do not.

Method: Transformers reading integrands as strings (in Polish notation to reduce length), similar to Lample & Charton (2020).

R. Barket, U. Shafiq, M. England, and J. Gerhard.
Transformers to Predict the Applicability of Symbolic Integration Routines
Proc. MATH-AI at NeurIPS '24, 2024.
https://openreview.net/forum?id=b2Ni828As7

Guarding integration methods – Results

Method	Accuracy (%)		Precision (%)	
	Transformer	Guard	Transformer	Guard
Trager	**98.21**	67.55	**92.21**	15.88
MeijerG	**96.78**	88.72	**89.58**	57.78
PseudoElliptic	**99.28**	62.61	**62.86**	2.03
Gosper	**94.04**	92.51	**92.08**	80.21

Results for methods **with** a guard.

Method	Accuracy (%)	
	Transformer	Guard
Default	**94.86**	82.15
DDivides	**94.13**	28.18
Parts	**93.10**	37.05
Risch	**94.53**	89.35
Norman	**95.74**	71.67
Orering	**97.21**	37.88
ParallelRisch	**97.82**	82.73

Results for methods **without** a guard.

The "guard" in this case is simply predicting to use the method every time.

Representing Mathematical Data

Is treating mathematics as a string the right approach? It does not reflect the commutativity in the expressions being integrated!
Are there alternatives?

$$3x^2 + \cos(2x) - 1$$

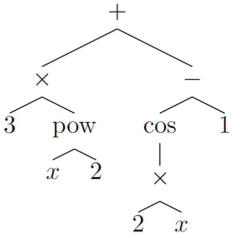
Text Sequence

['add',
'-1',
'add',
'mul',
'3',
'pow',
'x',
'2',
'cos',
'mul',
'2',
'x']

Feature Vector

	Vars	Terms	Has Trig	...
$3x^2 + \cos(2x) - 1$	1	3	True	...

Graph



Experimental Setup

Compare graph and sequential embeddings (all else kept constant):

Binary classifier for each of the 12 integration methods, this time to predict if it gives the optimal (shortest) answer for an input.

Order methods by probability of optimality given by the classifiers. Then try in turn until one succeeds, whose answer we take.

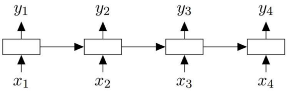
Train on 100k examples from the data generators. Evaluate on (a) separate 25k from the data generators; (b) 7413 examples from the Maple test suit (selected based on data generation function scope).

R. Barket, M. England, and J. Gerhard.
Symbolic Integration Algorithm Selection with Machine Learning: LSTMs vs Tree LSTMs.
Proc. ICMS '24, LNCS 14749, pp. 167–175, 2024.
https://doi.org/10.1007/978-3-031-64529-7_18

LSTM vs TreeLSTM

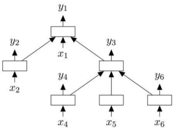
LSTM

- Sequential data processing
- The memory cell updates are dependent on the input at the current timestep and the hidden state of the previous timestep
- Math expressions can be represented as a sequence of tokens to fit the LSTM framework

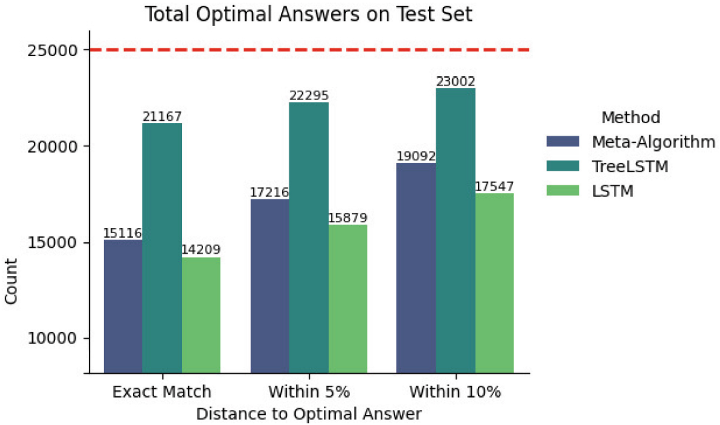


TreeLSTM

- Tree-based data processing
- The memory cell updates are dependent on the states of possibly many child units.
- A regular LSTM can be seen as a TreeLSTM where each cell has one child to form a chain.
- Math expressions can be represented as a binary tree to fit the TreeLSTM framework

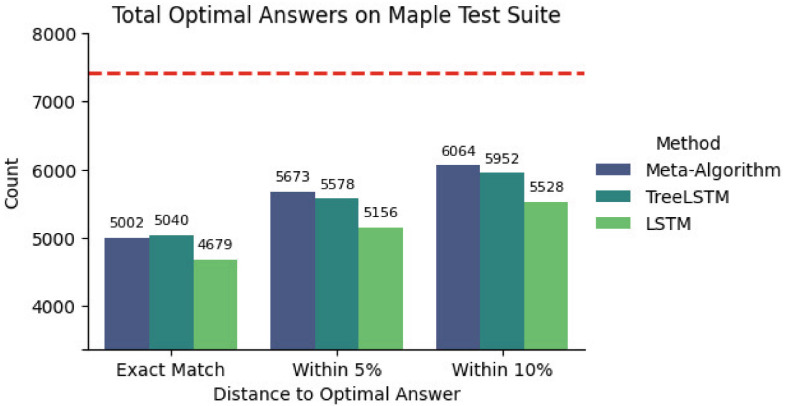


Results on hold out test data



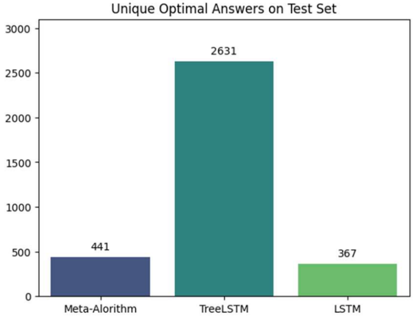
Conclusion: TreeLSTMs far superior to LSTMs. The tree structure better reflects the mathematical data.

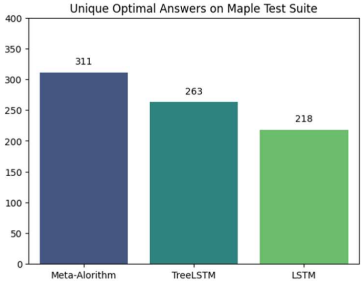
Validation on Maple test suite



Conclusion: Methodology can generalise to unseen data: the validation dataset is completely independent of the training data.

Unique Validation on Maple test suite





Conclusion: What the TreeLSTM learns is not a direct subset of what the LSTM learns.

How to Improve Performance Further

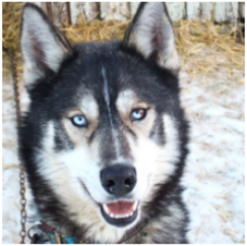
Our simple first attempt competing with the Maple meta-algorithm. Ideas to improve further:

- Scale up the data: requires more efficient architecture. Now implementing a tree transformer method.
- Classifier chains: let the prediction of the classifier for one method be used by the model to decide the next.
- Regression rather than classification: learn from not just whether a result is optimal but from how close to optimal it is.
- A two-stage approach: first a classifier to predict if a method is applicable and then regression to order the applicable ones.

Final Thought: Explainable AI (XAI)

Explainable AI (XAI) refers to AI systems that can provide explanations for their actions. useful:

- As an error check on the ML process.
- To build trust in the systems; regulatory compliance.

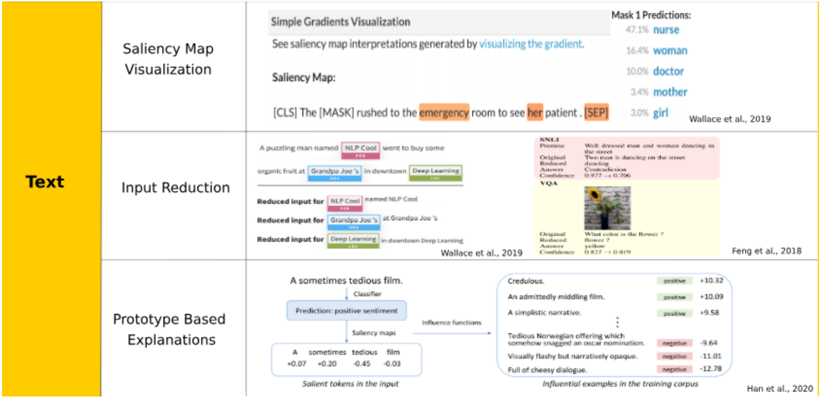


(a) Husky classified as wolf

(b) Explanation

Idea: XAI applied to ML models guide CAS development?

From Maveli 2021 (Spectra) Text



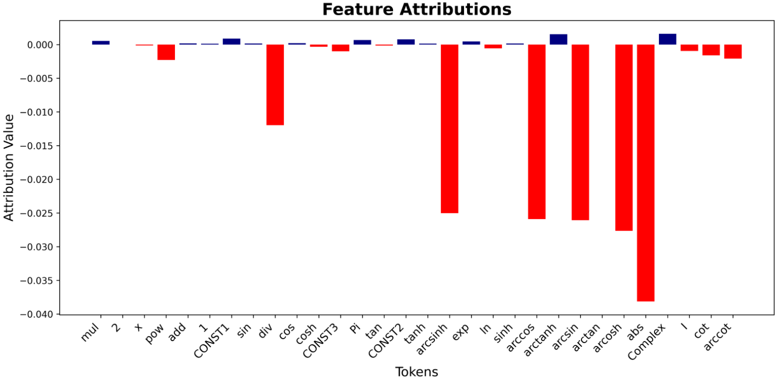
LIGs on Transformer Guards

We experimented with Layered Integrated Gradients (LIGs) applied to our embedding layer to compute an attribution score for each input token.

One pattern spotted in these explanations is the high scores applied to the abs token for the Risch sub-algorithm classifier.

[CLS] mul INT+ CONST1 mul x pow abs cos mul INT+ 2 pow x INT+ 2 INT- 1

Upon inspection: this sub-algorithm cannot be applied when there is abs in the input, but the current Maple guard code did not check for this. An improvement offered to the CAS developers independent of whether they embed the ML or not.


Feature Attributions
Attribution Value
0.000
−0.005
−0.010
−0.015
−0.020
−0.025
−0.030
−0.035
−0.040
mul
2
x
pow
add
1
CONST1
sin
div
cos
cosh
CONST3
Pi
tan
CONST2
tanh
arcsinh
exp
ln
sinh
arccos
arctanh
arcsin
arctan
arcosh
abs
Complex
I
cot
arccot
Tokens

LIGs on Transformer Guards

We experimented with Layered Integrated Gradients (LIGs) applied to our embedding layer to compute an attribution score for each input token.

One pattern spotted in these explanations is the high scores applied to the abs token for the Risch sub-algorithm classifier.

[CLS] mul INT+ CONST1 mul x pow abs cos mul INT+ 2 pow x INT+ 2 INT- 1

Upon inspection: this sub-algorithm cannot be applied when there is abs in the input, but the current Maple guard code did not check for this. An improvement offered to the CAS developers independent of whether they embed the ML or not.

Thanks for Listening

Matthew.England@coventry.ac.uk
https://matthewengland.coventry.domains