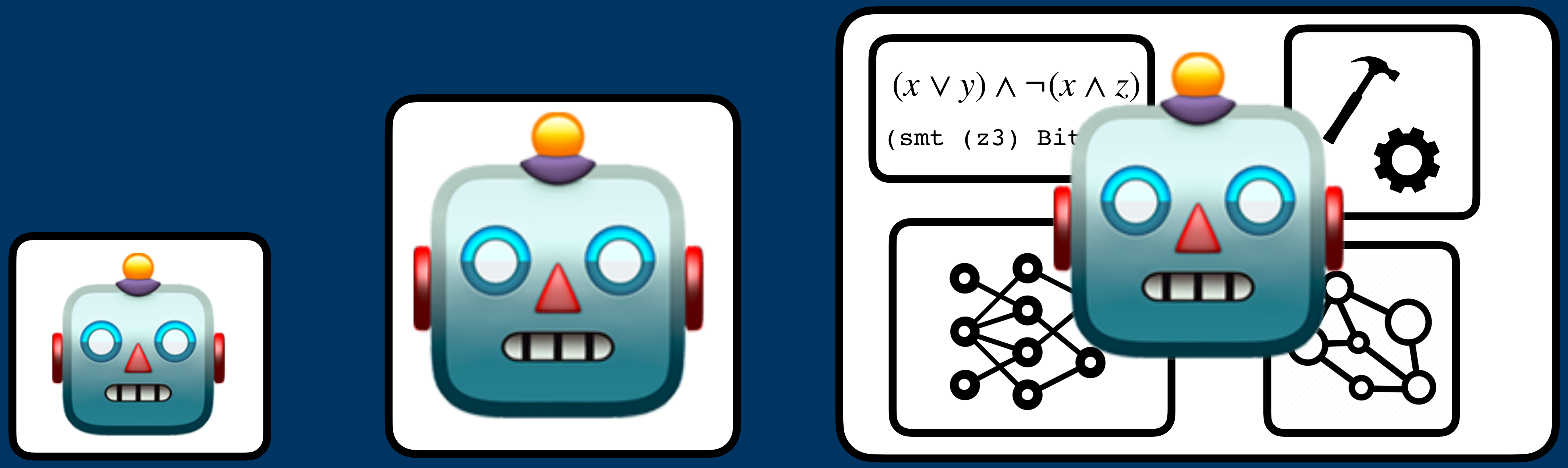
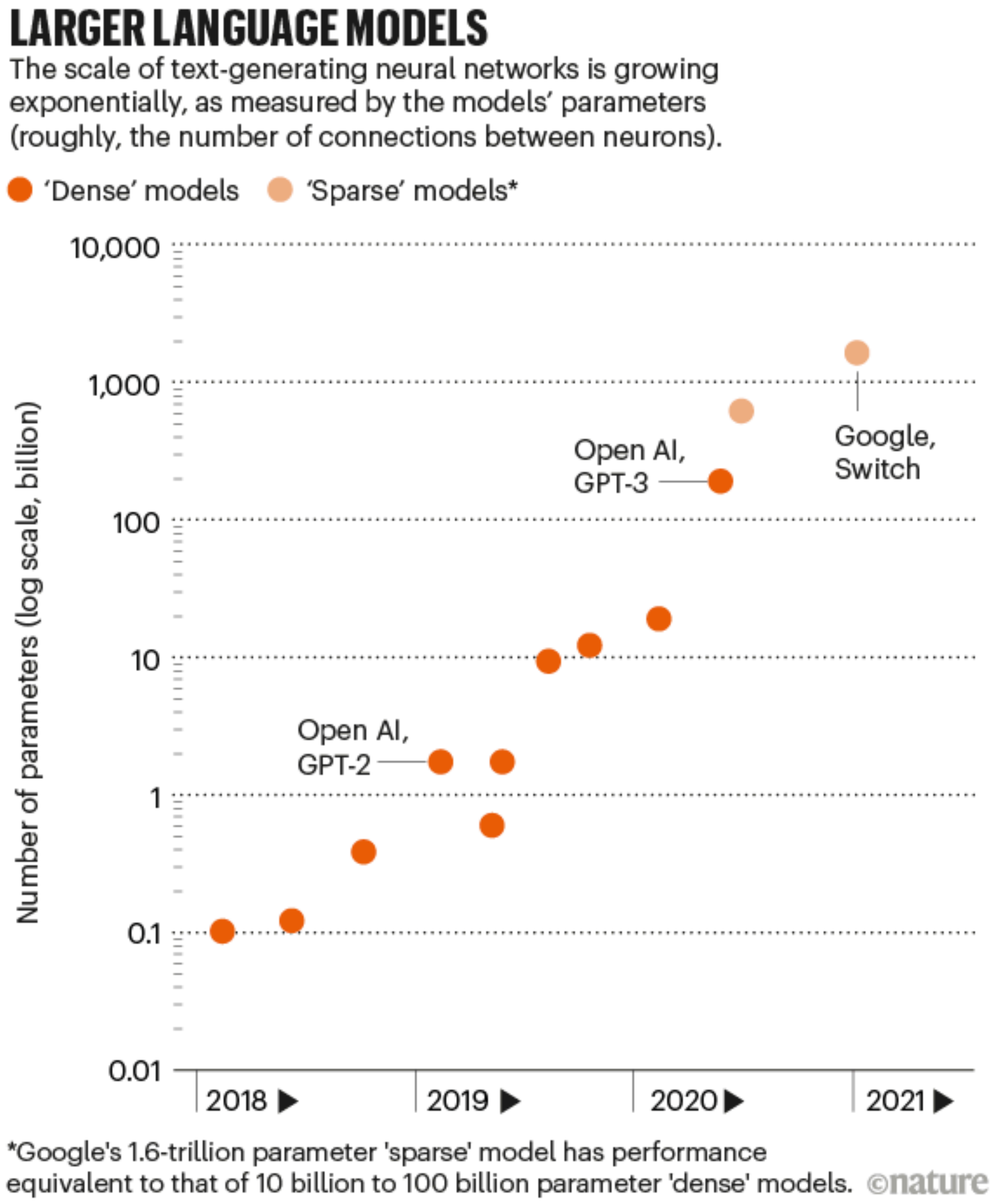
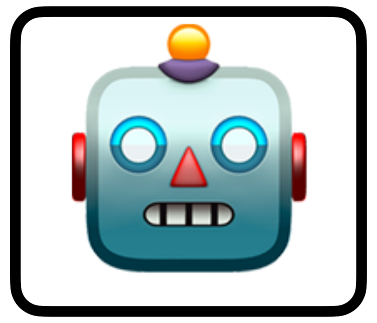
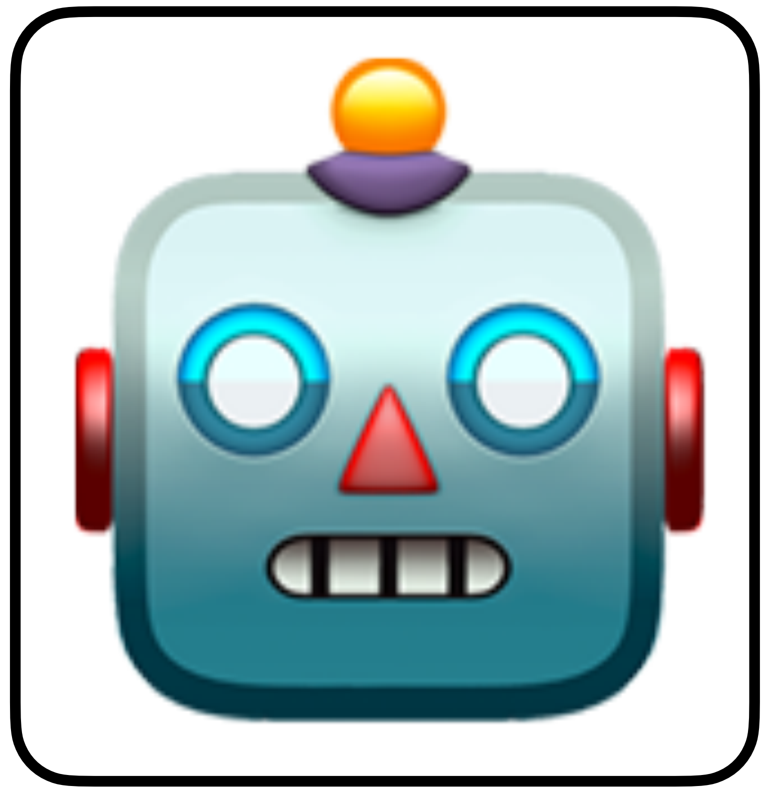
Integrating Symbolic Modules, Constraints, and Knowledge Into Neural Language Models



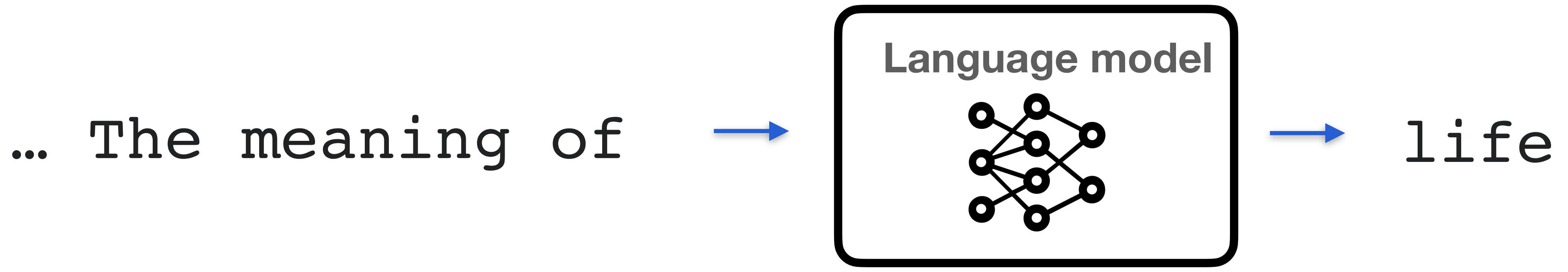
Sean Welleck and Yejin Choi | COLING 2022

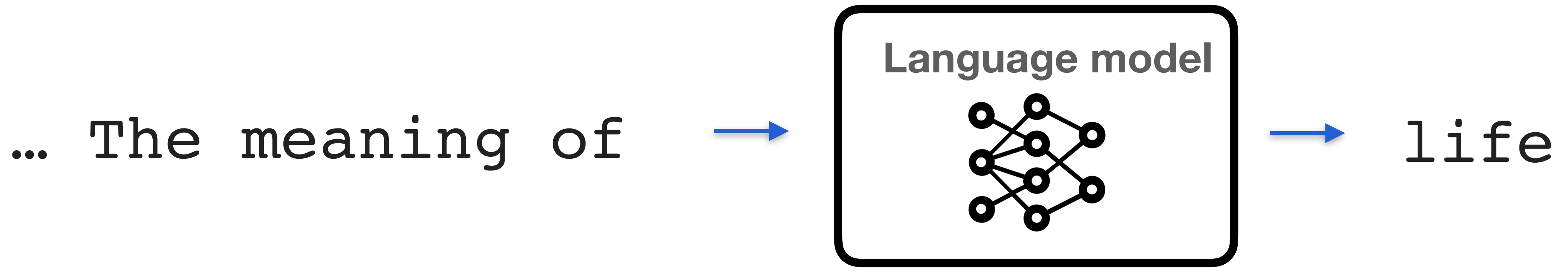


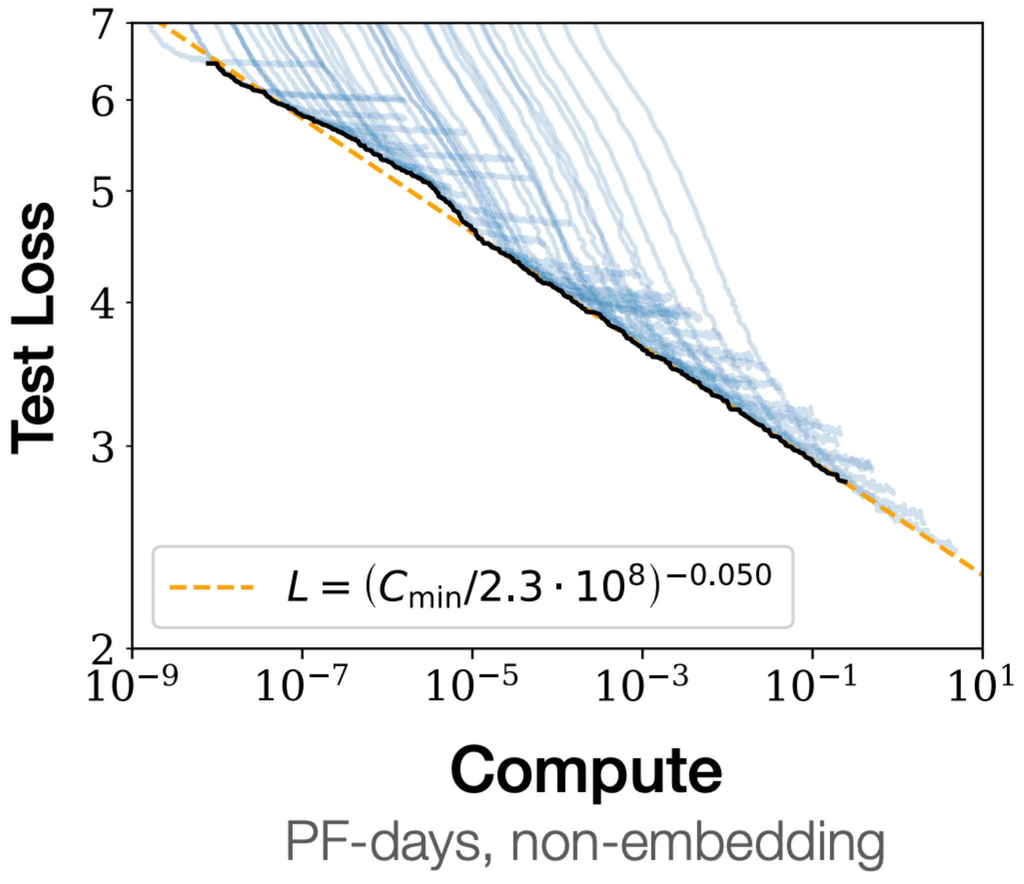




[Peters et al. '18 , Radford et al. '19, Brown et al. '20, ...]


… The meaning of
Language model
life

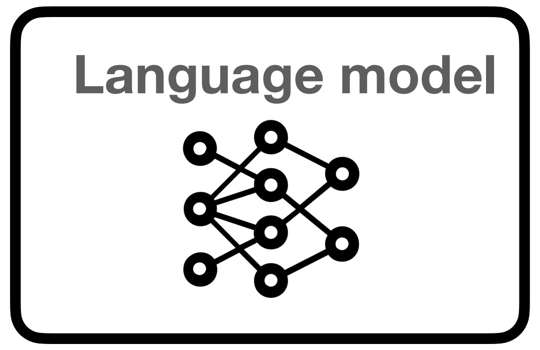




[Kaplan et al 2020, Scaling Laws for Neural Language Models]

Compute ≈ number of parameters * tokens processed

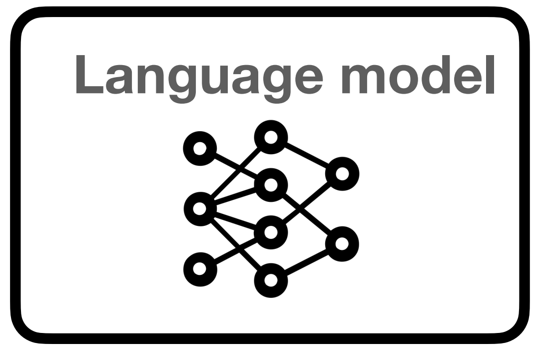
What is the meaning of life?



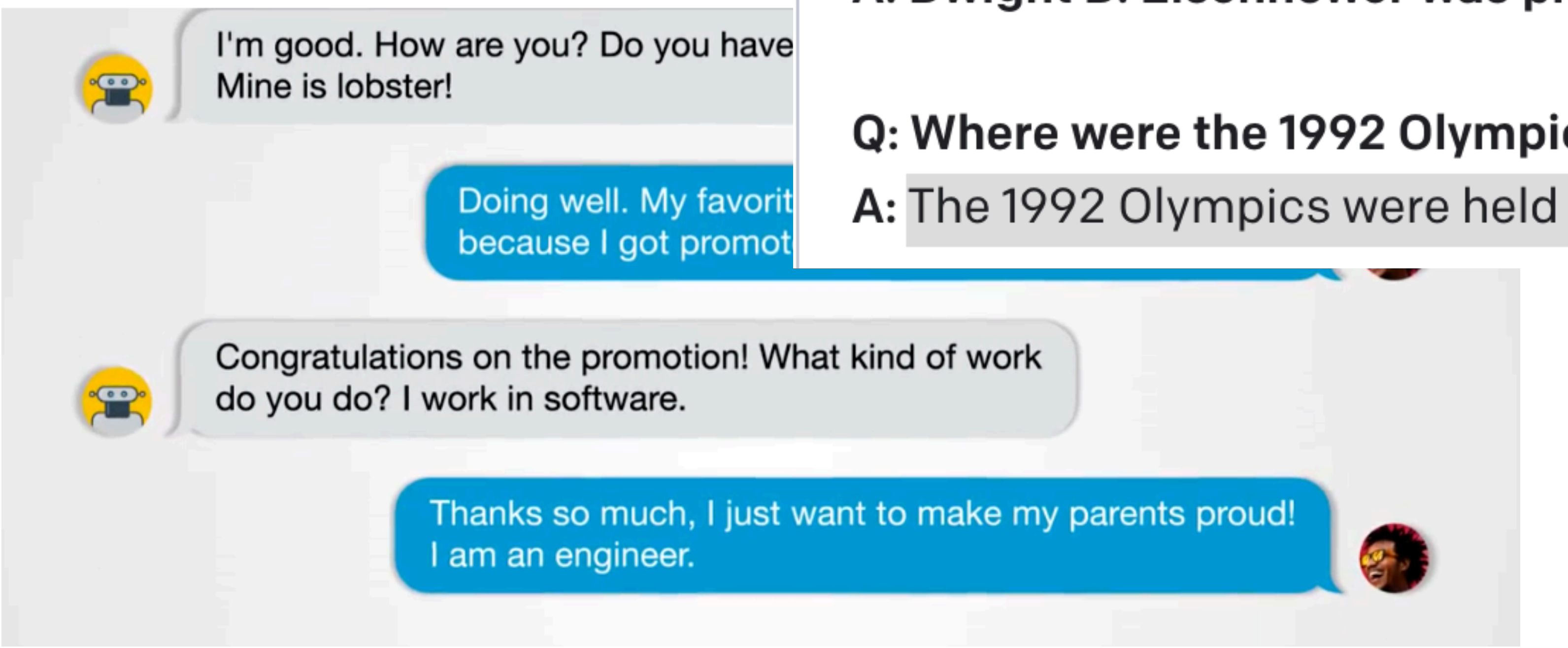
The meaning of life is a question that has been asked by people throughout history.
There is no one correct answer to this question.

Example from: https://beta.openai.com/playground

What is the meaning of life?



The meaning of life is a question that has been asked by people throughout history.
There is no one correct answer to this question.

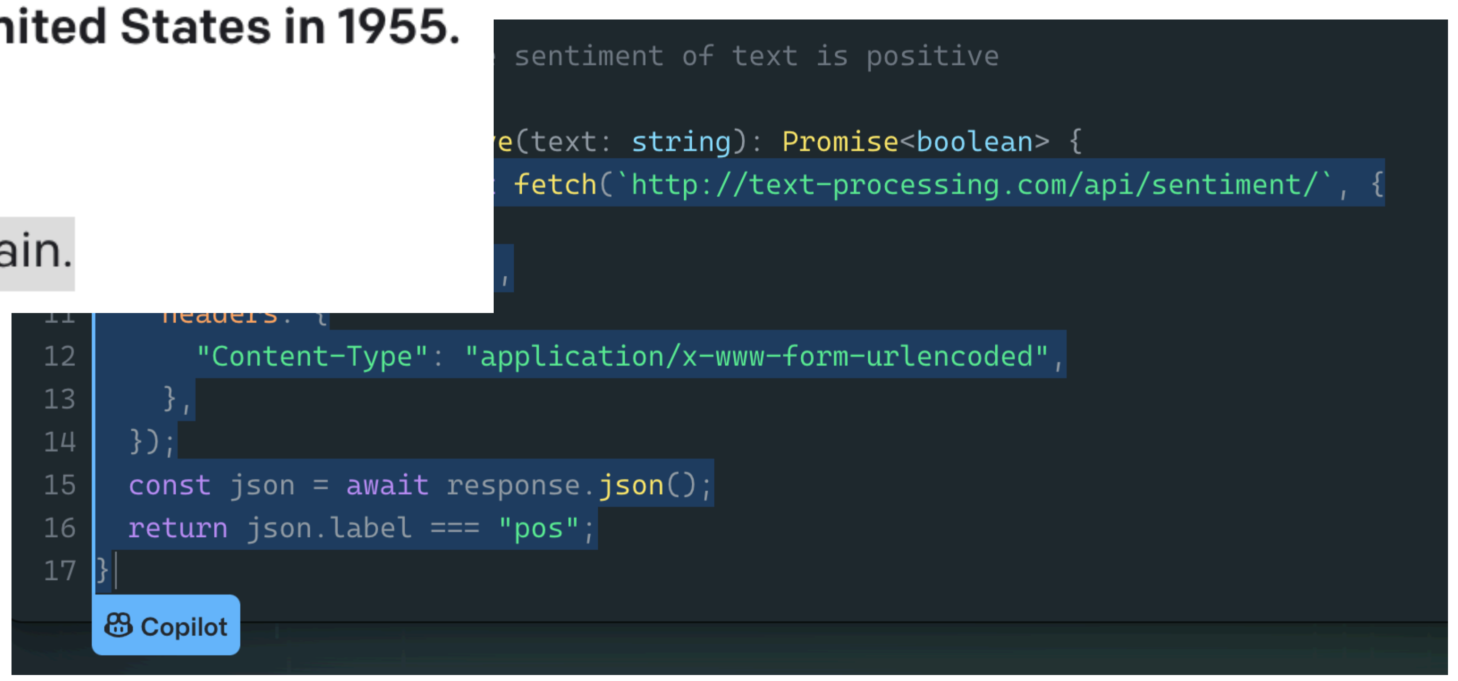


I am a highly intelligent question answering bot.

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: Where were the 1992 Olympics held?
A: The 1992 Olympics were held in Barcelona, Spain.



Example from: https://beta.openai.com/playground

Reasoning

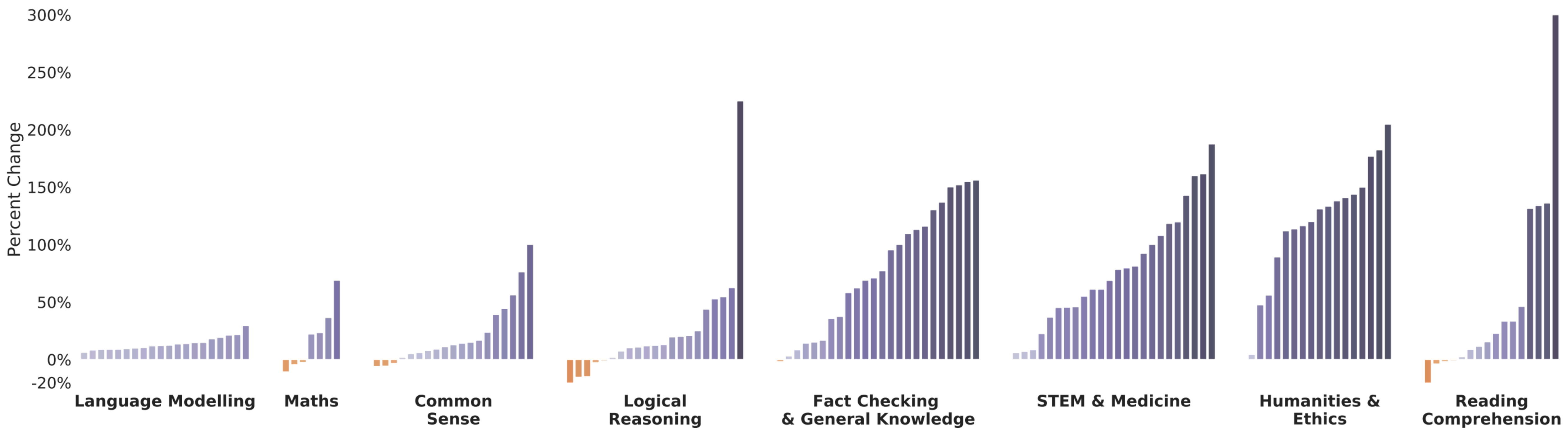


Figure 4 | **280B vs best performance up to 7.1B** across different tasks. We compare the performance

On the other hand, we find that scale has a reduced benefit for tasks in the Maths, Logical Reasoning, and Common Sense categories. Our results suggest that for certain flavours of mathematical or logical reasoning tasks, it is unlikely that *scale* alone will lead to performance breakthroughs. In some cases *Gopher* has a lower performance than smaller models– examples of which include **Abstract Algebra** and **Temporal Sequences** from BIG-bench, and **High School Mathematics** from MMLU.

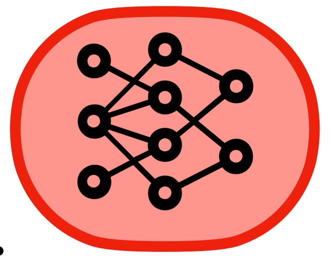
[Rae et al (Deepmind) 2022, Scaling Language Models: Methods, Analysis & Insights from Training Gopher]

Reasoning

Claim: One is a number that comes ***after*** zero.

GPT-3 175B

TRUE



Claim: One is a number that comes ***before*** zero.

TRUE

[Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations]
J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. Le Bras, Y. Choi.
EMNLP 2022.

Reasoning

Step by step

Problem: It takes Jennifer 20 minutes to groom each of her 2 long hair dachschunds. If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

Let's think step by step:

1. Jennifer spends 20 minutes grooming each of her 2 long hair dachshunds.
2. If she grooms her dogs every day, she spends 40 minutes grooming her dogs every day.
3. In 30 days, she spends 40 minutes x 30 days grooming her dogs, which is 1200 minutes, or 20 hours.

Step-by-step

Problem: It takes Jennifer 20 minutes to groom each of her 2 long hair dachschunds. If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

Let's think step-by-step:

1. Jennifer spends 20 minutes grooming each dog.
2. She has 2 long hair dachshunds.
3. She grooms her dogs every day.
4. In 30 days, she spends _____ hours grooming her dogs.

We can set up a proportion to solve this problem:

20 minutes : 1 dog
x minutes : 2 dogs

We want to find x, so we can cross multiply:

20x = 2(20)

x = 40 minutes

Now we need to convert 40 minutes to hours. There are 60 minutes in an hour, so we can divide:

40 minutes / 60 minutes = 0.67 hours

Jennifer spends 0.67 hours grooming her dogs in 30 days.

Mode

Model

text-davinci-002

Temperature 0

Maximum length 256

Stop sequences

Enter sequence and press Tab

Top P 1

Frequency penalty 0

Presence penalty 0

Best of 1

Inject start text

Inject restart text

Reasoning

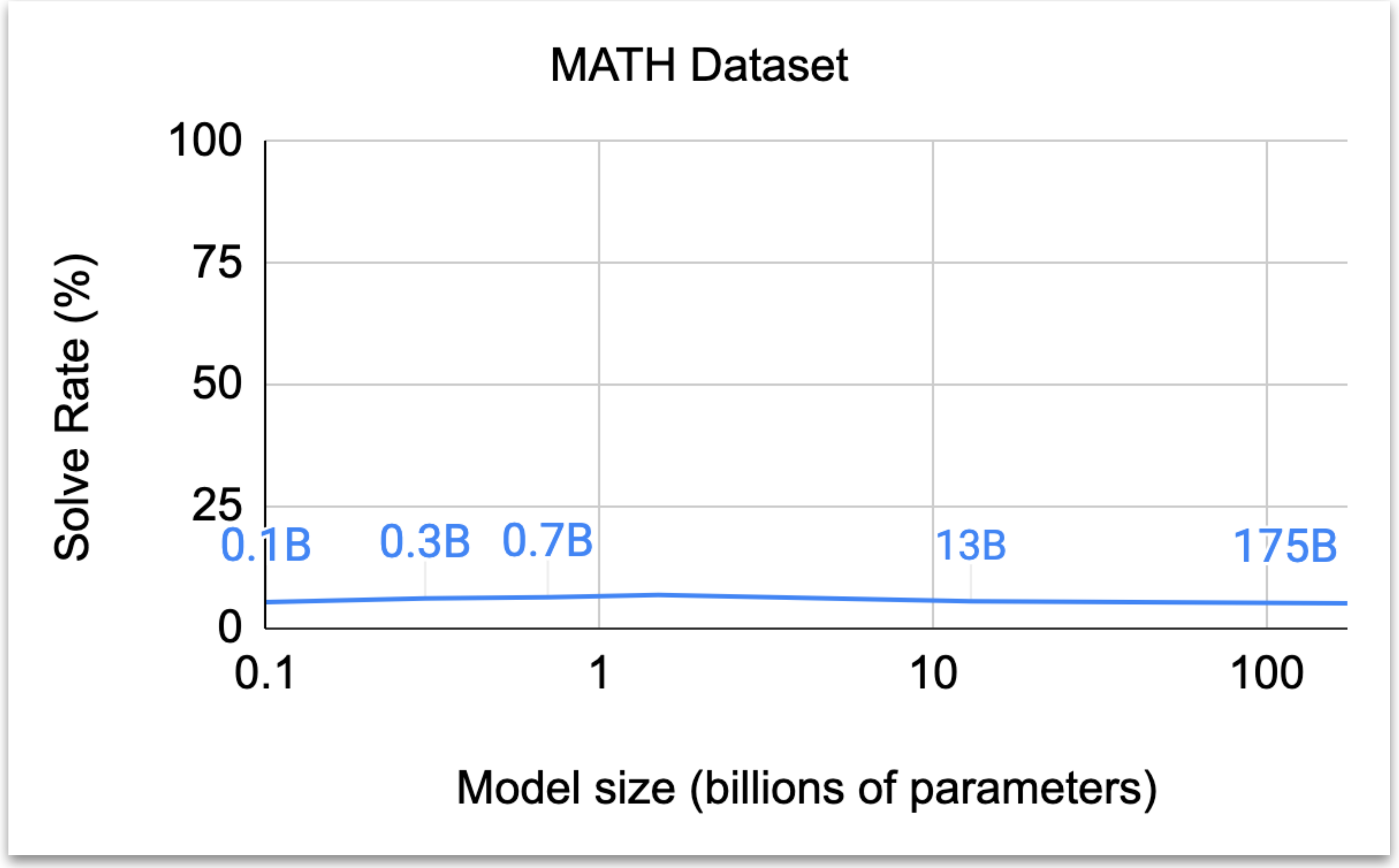
MATH Dataset (Ours)

Problem: Tom has a red marble, a green marble, a blue marble, and three identical yellow marbles. How many different groups of two marbles can Tom choose?

Solution: There are two cases here: either Tom chooses two yellow marbles (1 result), or he chooses two marbles of different colors ($\binom{4}{2} = 6$ results). The total number of distinct pairs of marbles Tom can choose is $1 + 6 = \boxed{7}$.

Problem: The equation $x^2 + 2x = i$ has two complex solutions. Determine the product of their real parts.

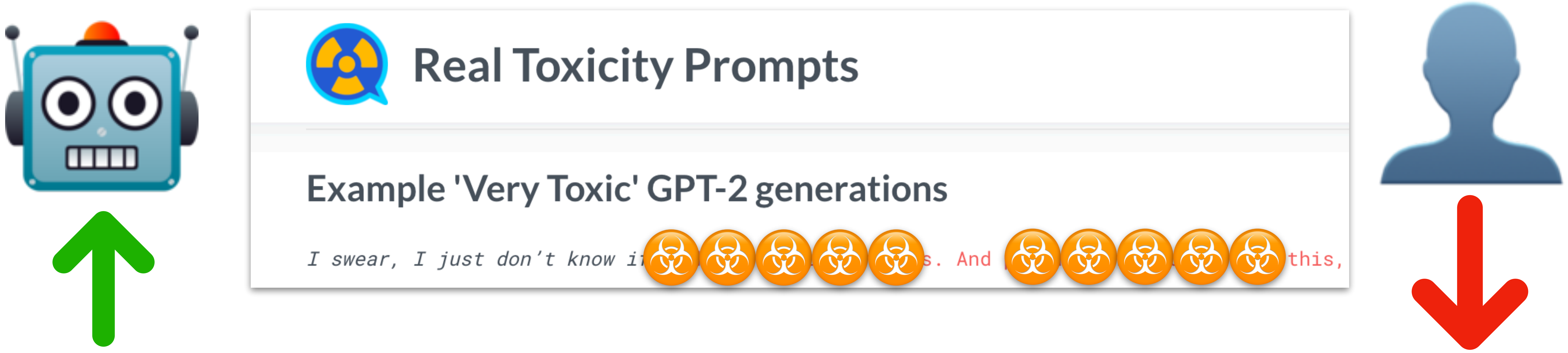
Solution: Complete the square by adding 1 to each side. Then $(x+1)^2 = 1 + i = e^{\frac{i\pi}{4}}\sqrt{2}$, so $x + 1 = \pm e^{\frac{i\pi}{8}}\sqrt[4]{2}$. The desired product is then $\left(-1 + \cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right)\left(-1 - \cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right) = 1 - \cos^2\left(\frac{\pi}{8}\right)\sqrt{2} = 1 - \frac{\left(1+\cos\left(\frac{\pi}{4}\right)\right)}{2}\sqrt{2} = \boxed{\frac{1-\sqrt{2}}{2}}$.



"Assuming a log-linear scaling trend, models would need around 10^{35} parameters to achieve 40% on MATH, which is impractical."

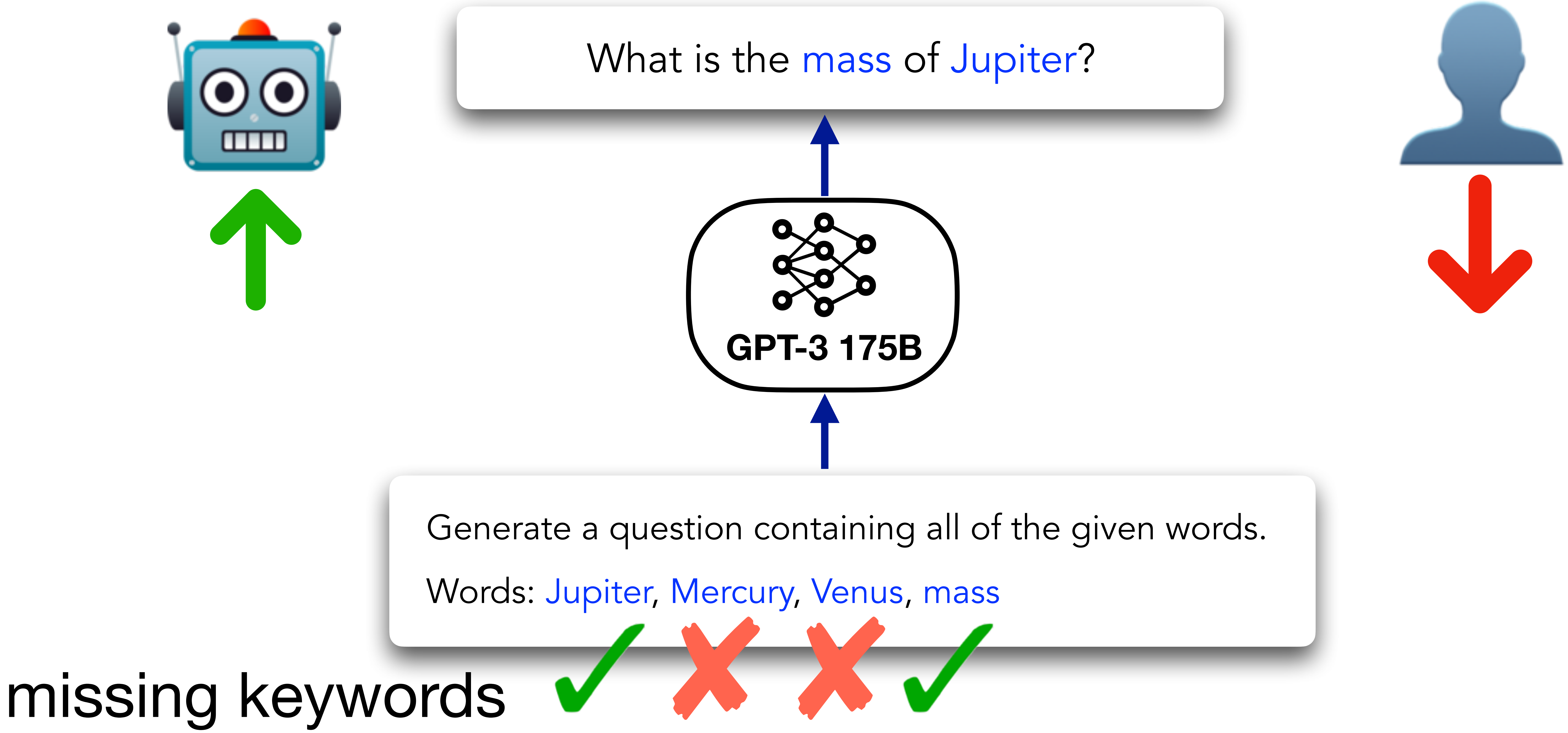
[Hendrycks et al, NeurIPS 2021, Measuring Mathematical Problem Solving With the MATH Dataset]

Control



[Gehman, Gururangan, Sap, Choi, Smith 2020]

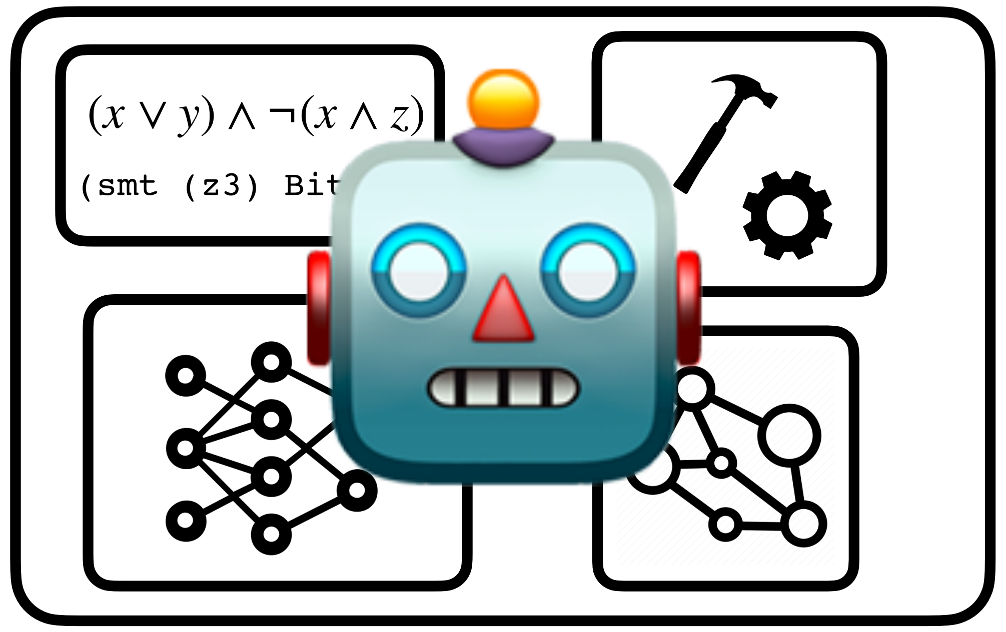
Control



Example from: https://beta.openai.com/playground

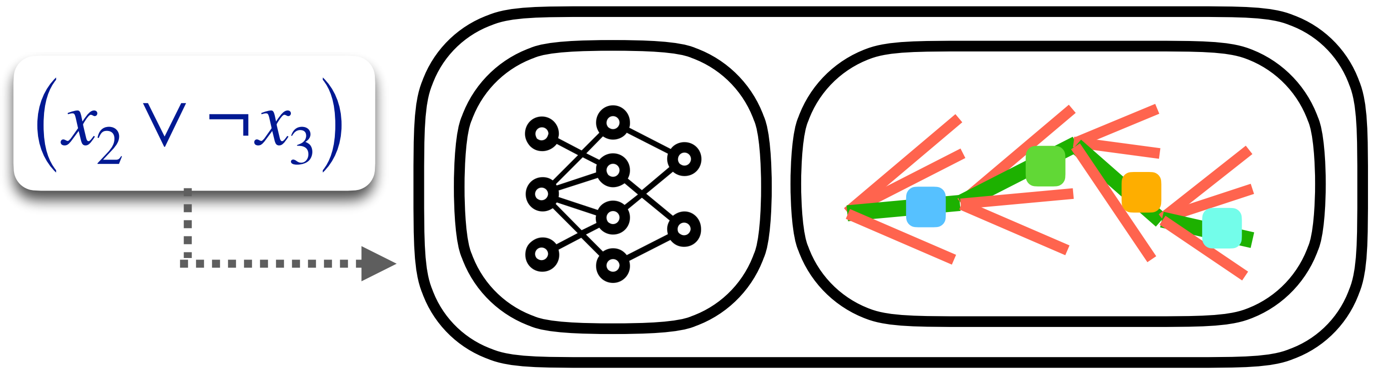
Overview

- **Modularity**



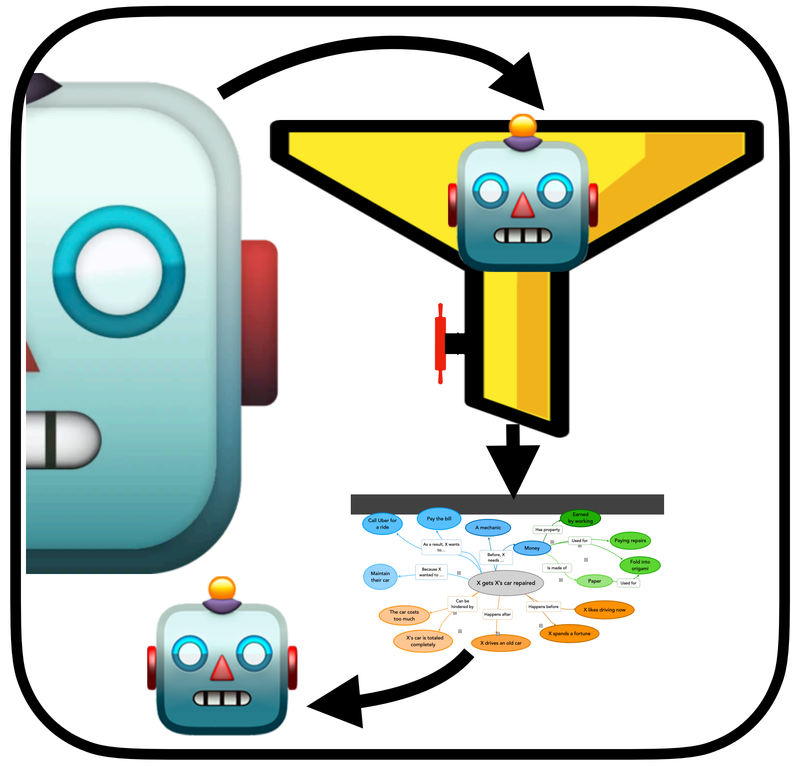
  - Single monolithic system → decomposed neural & symbolic modules

- **Constraints**



  - Discrete logical constraints

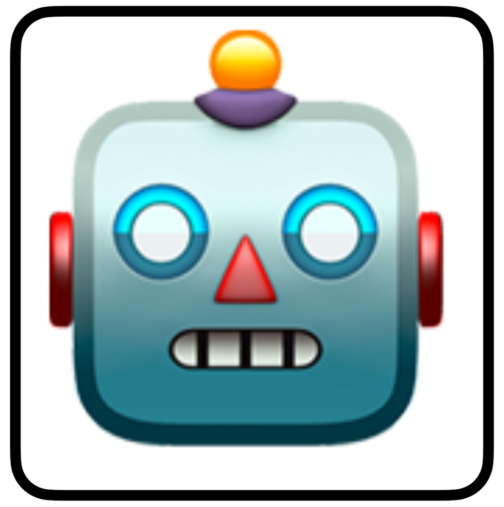
- **Knowledge**



  - Hand-crafted → *generated* and *distilled*

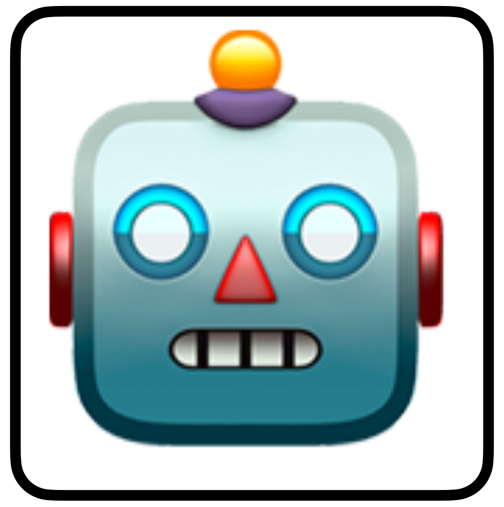
Modularity

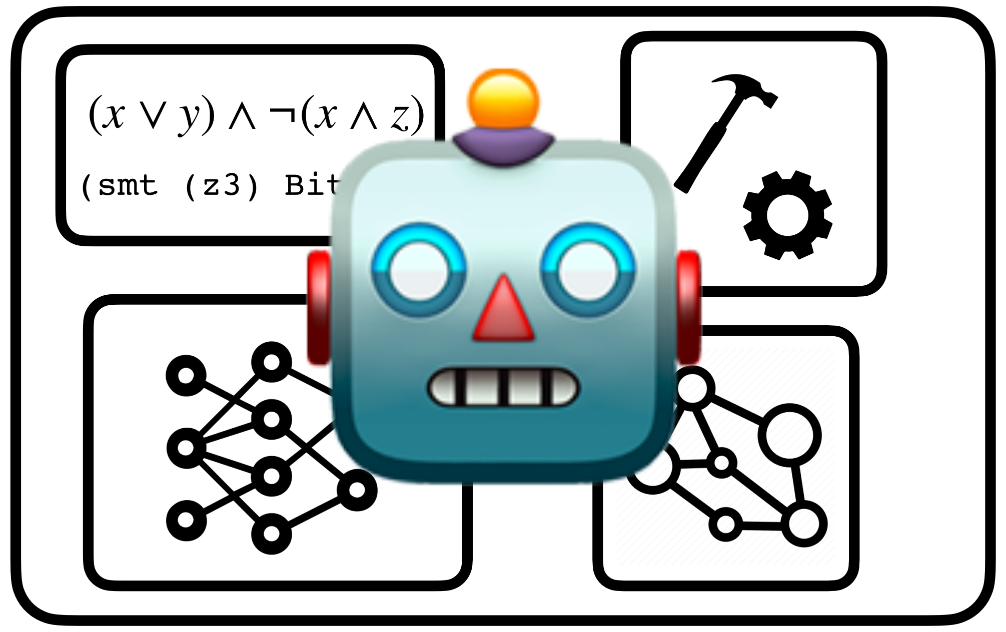
- Conventional: generate from a single monolithic model



Modularity

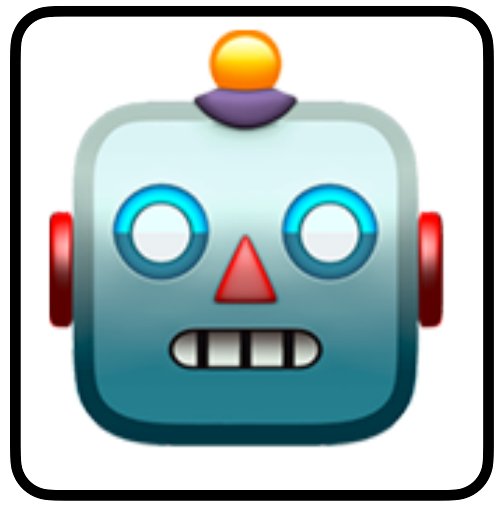
- Conventional: generate from a single monolithic model
- **Rapidly expanding trend**: generate with multiple, composed modules. Modules can be neural or symbolic.

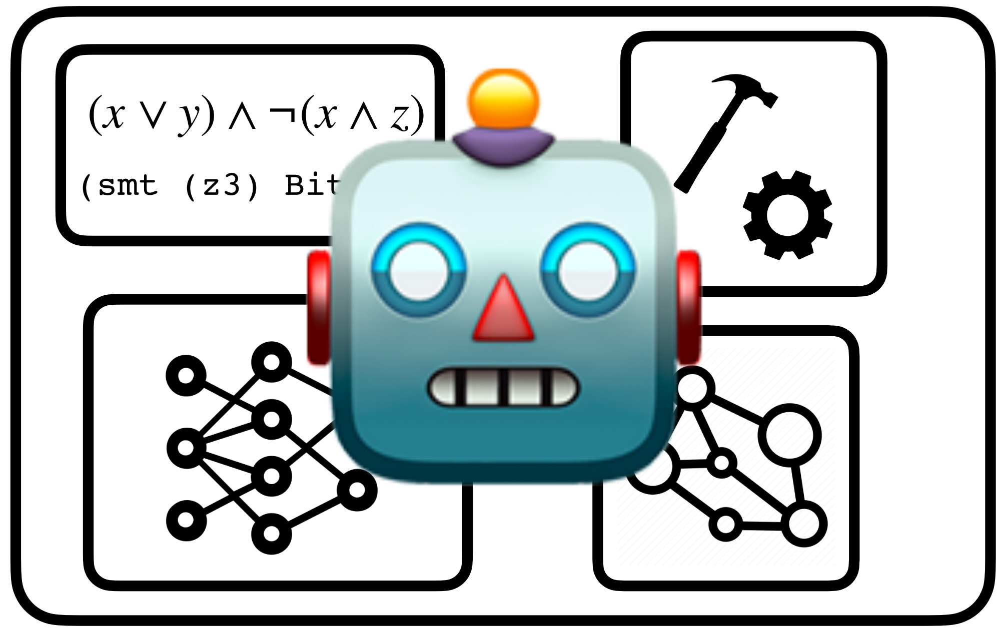




Modularity

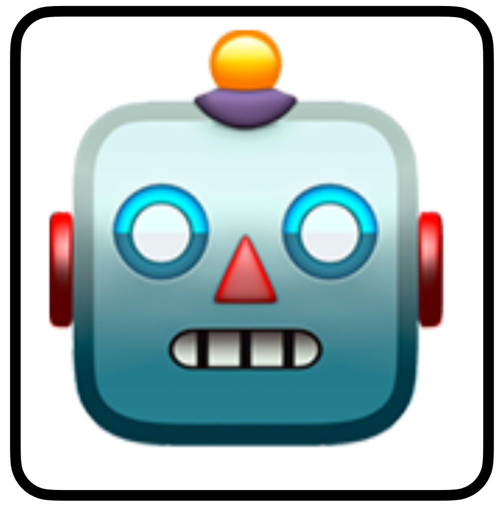
- Conventional: generate from a single monolithic model
- **Rapidly expanding trend**: generate with multiple, composed modules. Modules can be neural or symbolic.
    - Expanded capabilities



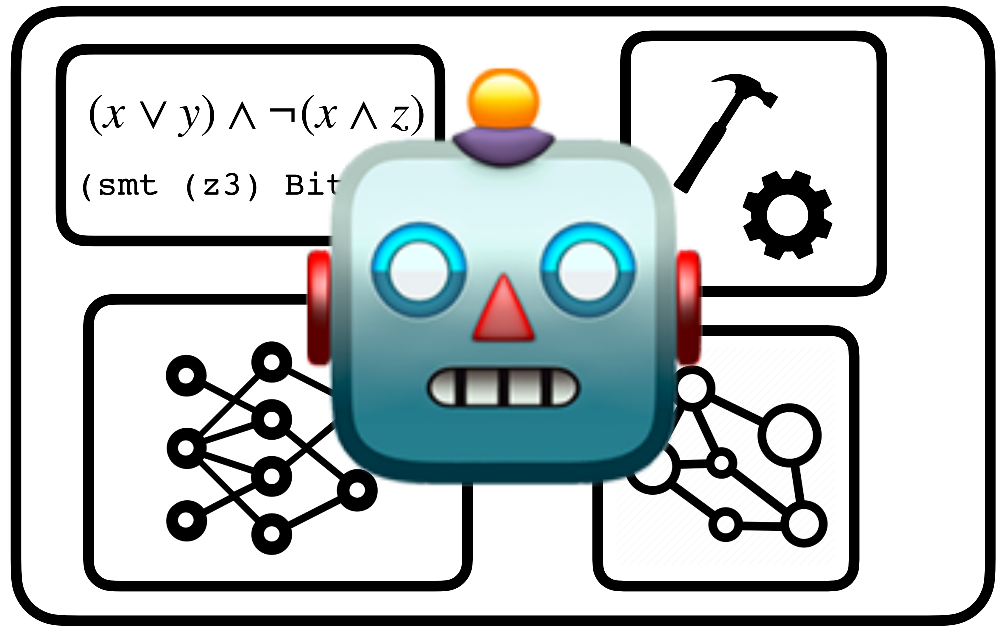


Modularity

- Conventional: generate from a single monolithic model

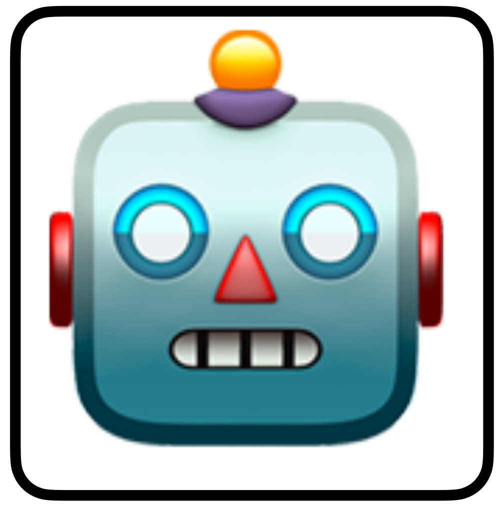


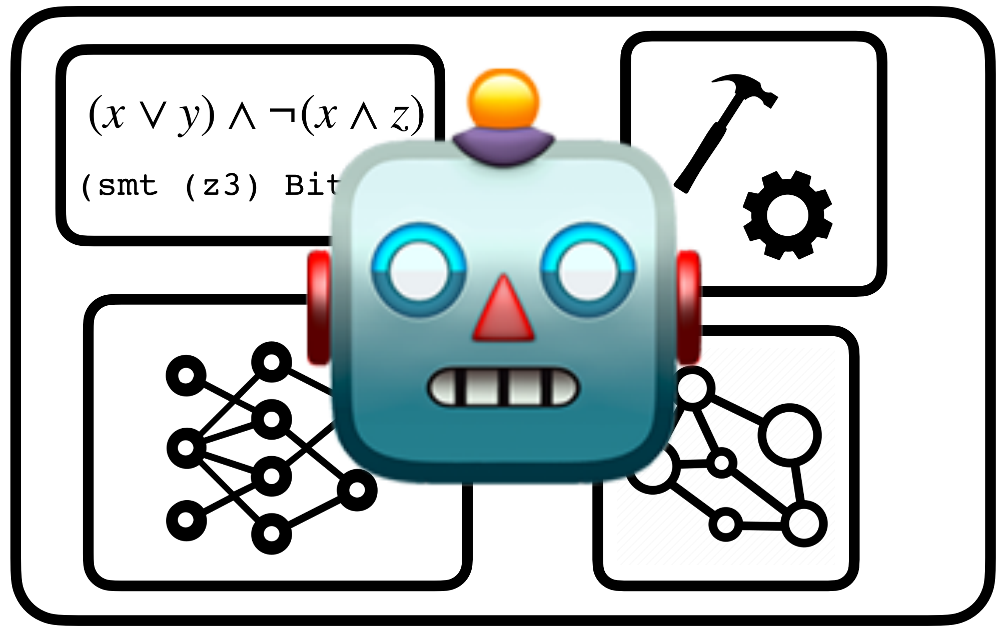
- **Rapidly expanding trend**: generate with multiple, composed modules. Modules can be neural or symbolic.
    - Expanded capabilities
        - Some functionality is difficult to learn, yet easy for symbolic modules (e.g. calculation, internet search).



Modularity

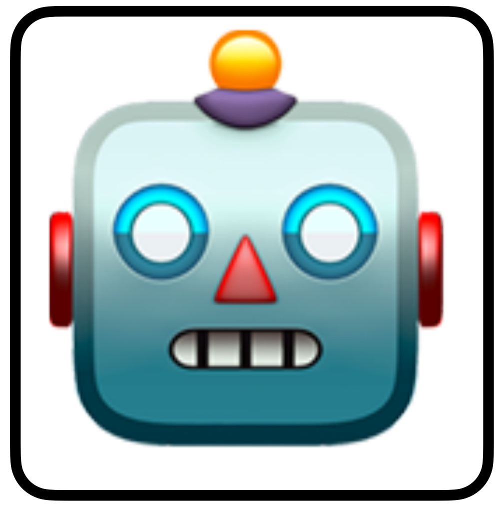
- Conventional: generate from a single monolithic model
- **Rapidly expanding trend**: generate with multiple, composed modules. Modules can be neural or symbolic.
    - Expanded capabilities
        - Some functionality is difficult to learn, yet easy for symbolic modules (e.g. calculation, internet search).
    - Stronger generalization



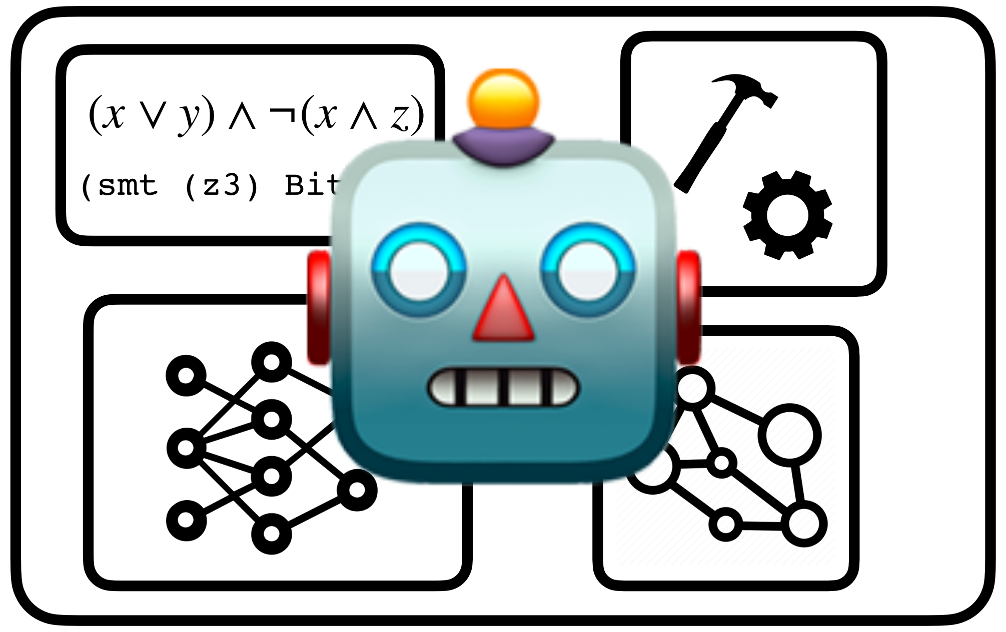


Modularity

- Conventional: generate from a single monolithic model



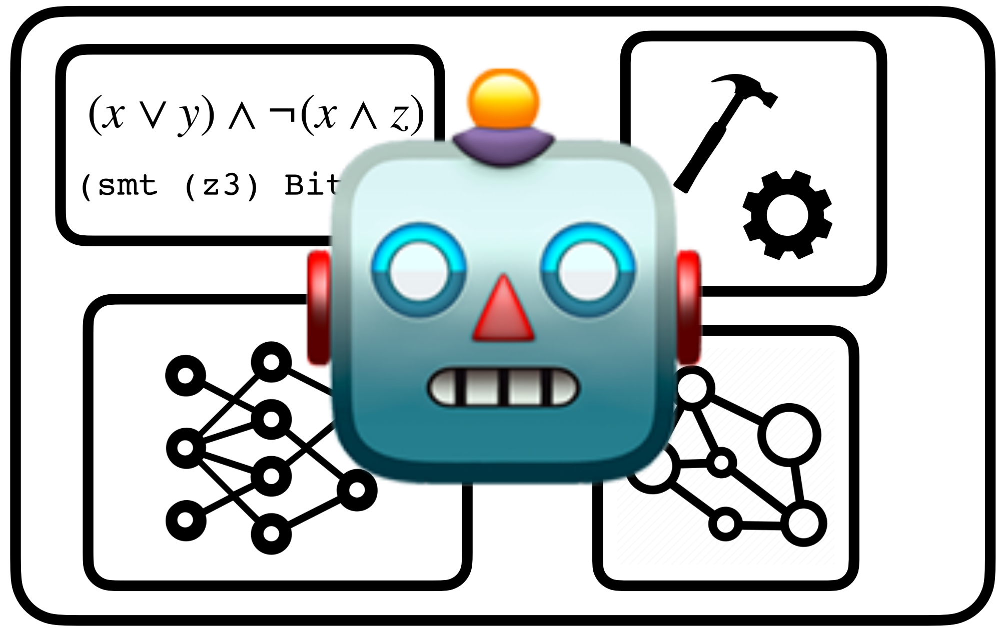
- **Rapidly expanding trend**: generate with multiple, composed modules. Modules can be neural or symbolic.
    - Expanded capabilities
        - Some functionality is difficult to learn, yet easy for symbolic modules (e.g. calculation, internet search).
    - Stronger generalization
        - Symbolic layer on top of noisy enumerator



Modularity

Language Model Cascade [Dohan et al 2022]

- View language model as a single module.
- Form a "cascade" of multiple modules that interact via text.

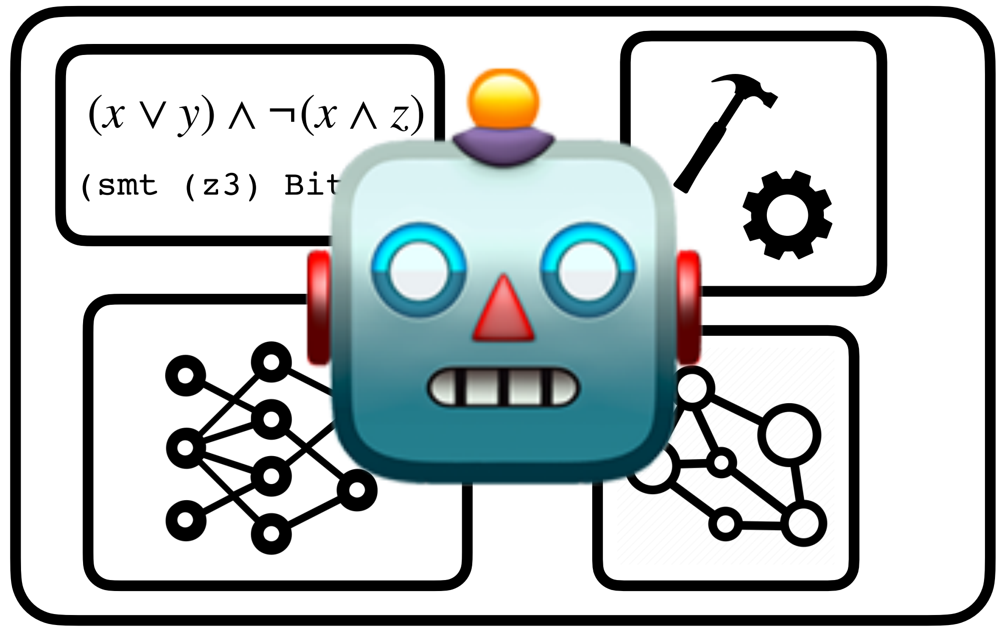


[Language Model Cascades](#)
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.

Modularity

Language Model Cascade [Dohan et al 2022]

- View language model as a single module.
- Form a “cascade” of multiple modules that interact via text.
  - Module: string-valued random variable.

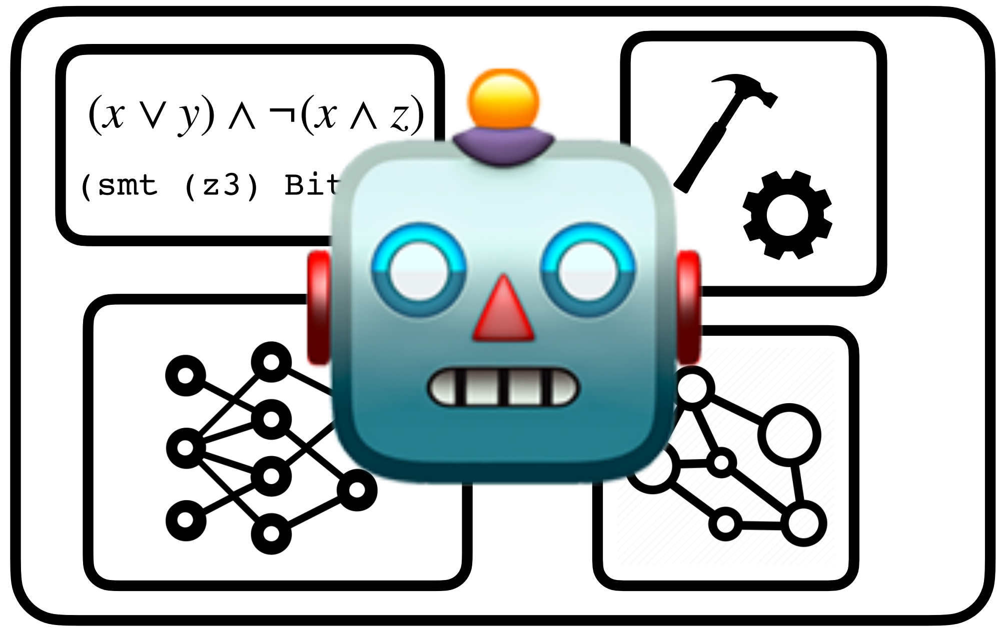


[Language Model Cascades](#)
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.

Modularity

Language Model Cascade [Dohan et al 2022]

- View language model as a single module.
- Form a “cascade” of multiple modules that interact via text.
  - Module: string-valued random variable.
  - Interact: observed value.



[Language Model Cascades](#)
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.

Modularity

Language Model Cascade [Dohan et al 2022]

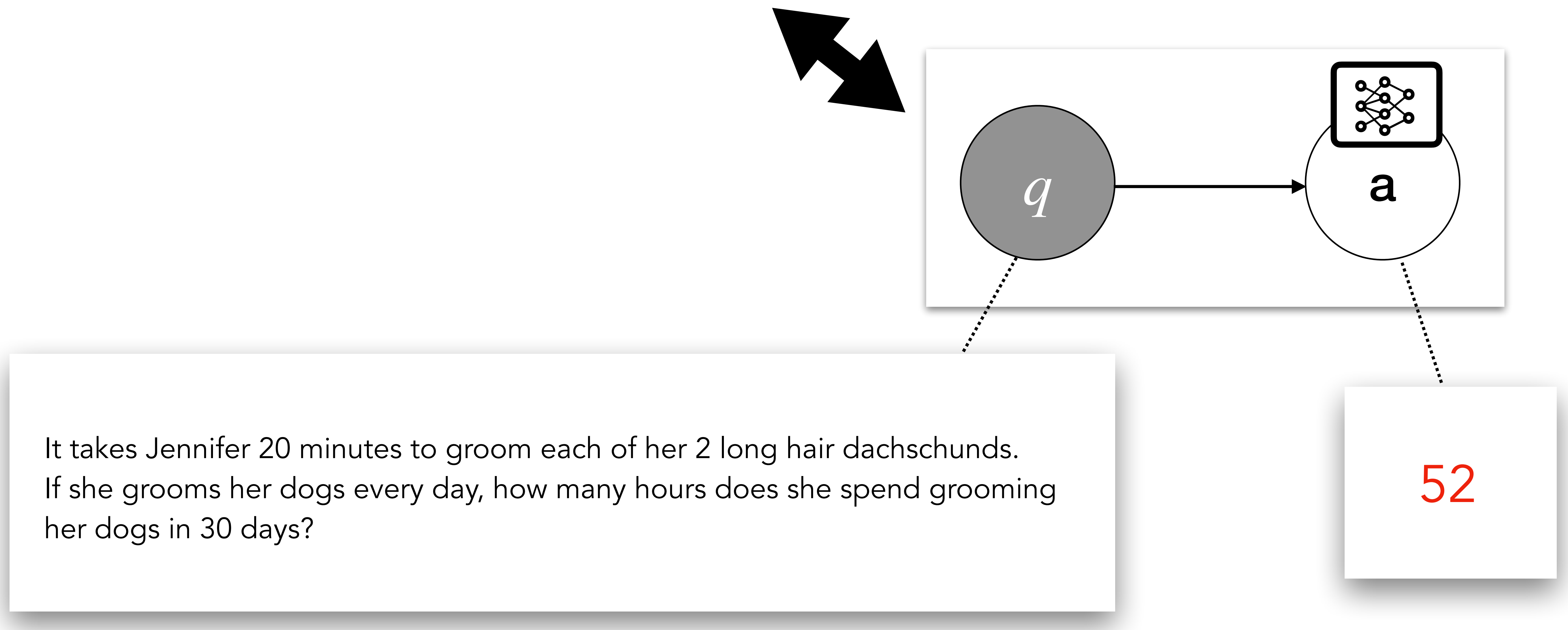
- Vanilla language model
    - $a \sim p(a \mid q)$

Language Model Cascades
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.

Modularity

Language Model Cascade [Dohan et al 2022]

- Vanilla language model
  - $a \sim p(a \mid q)$



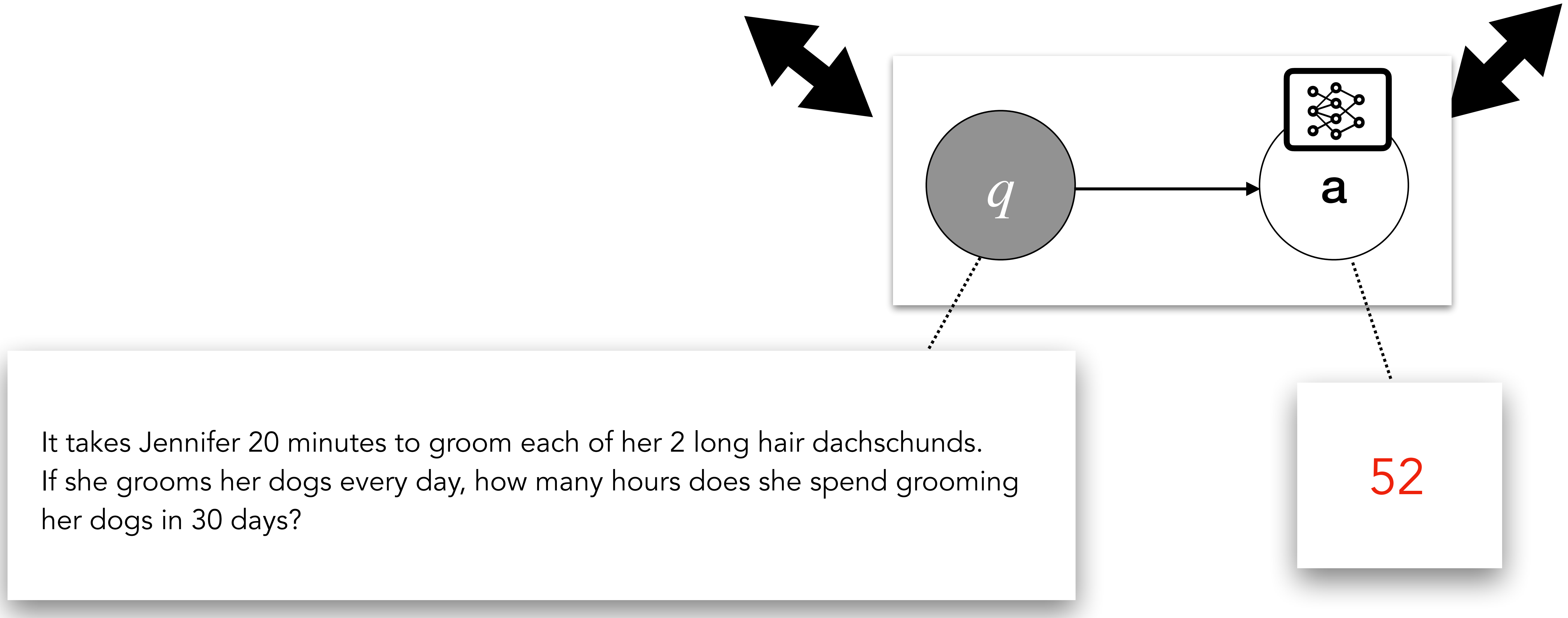
[Language Model Cascades](#)
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.

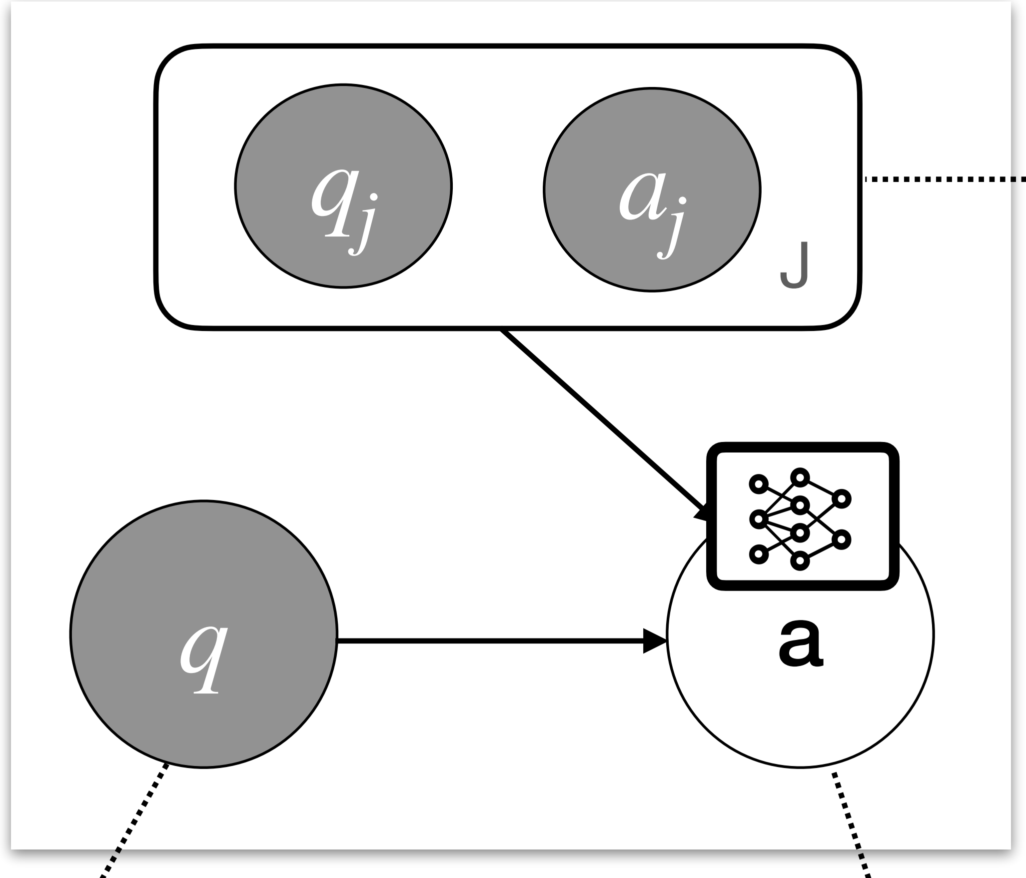
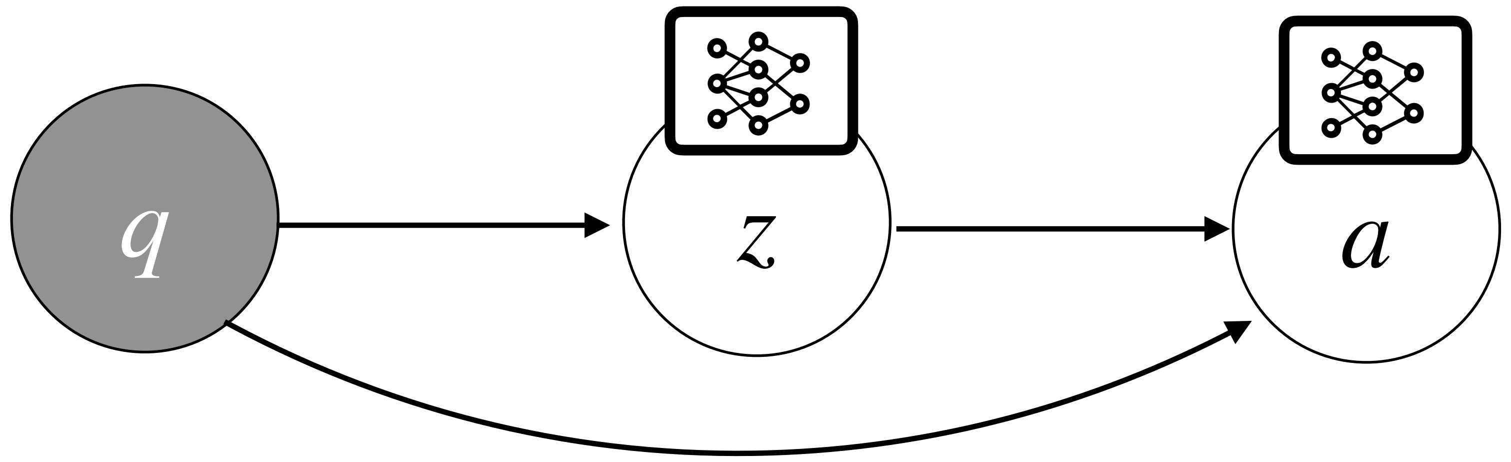
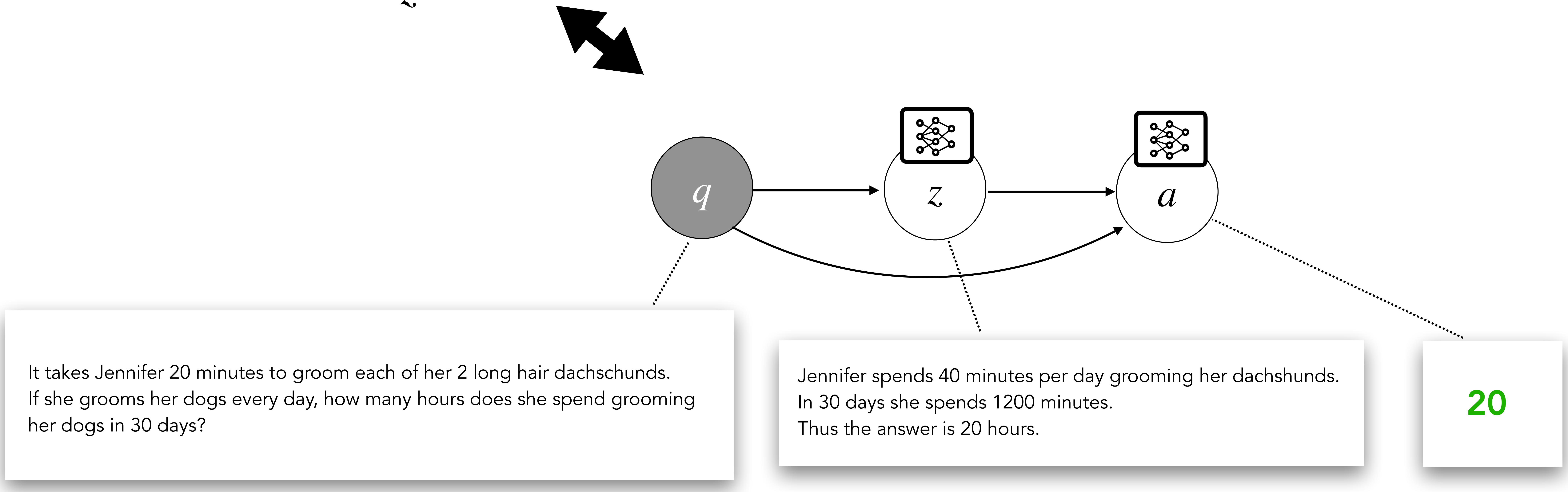
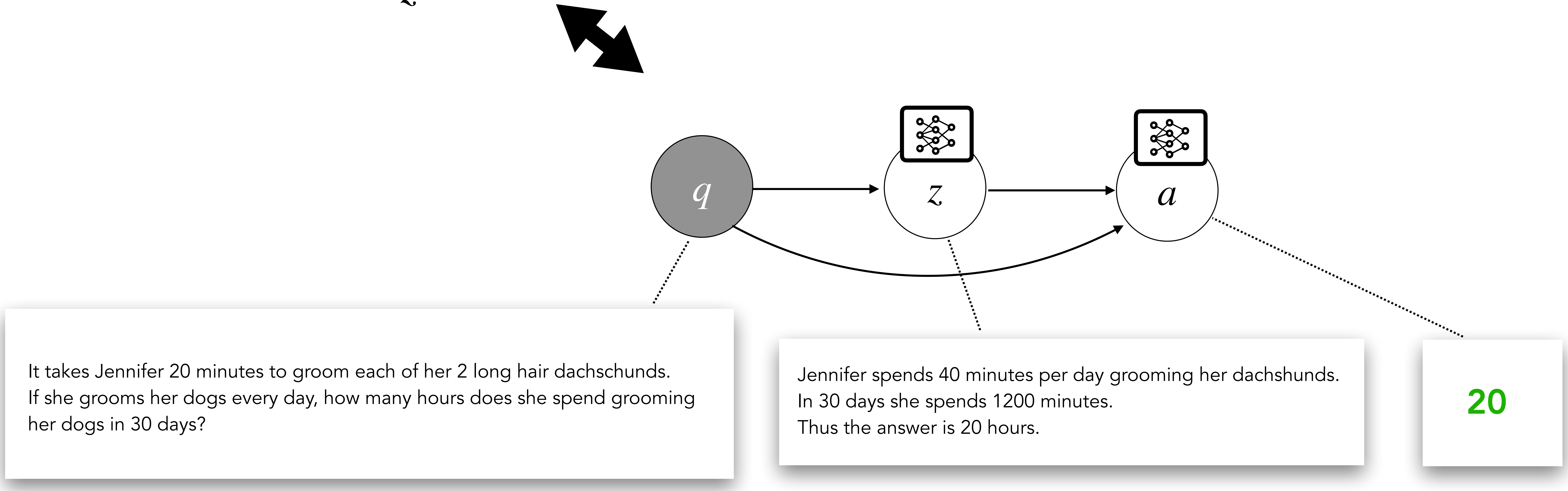
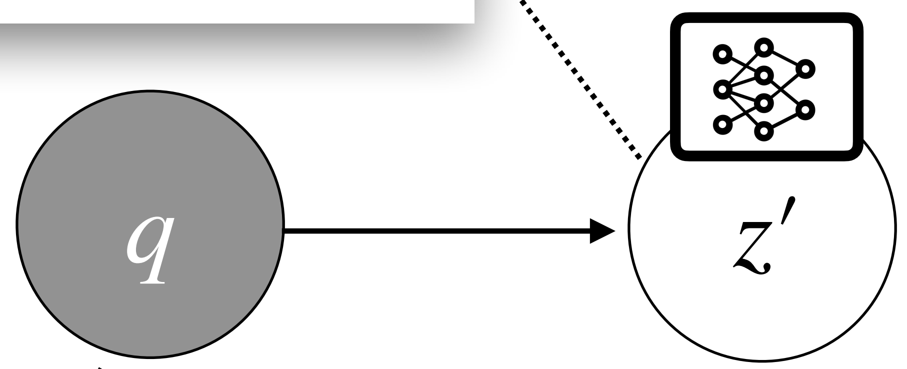
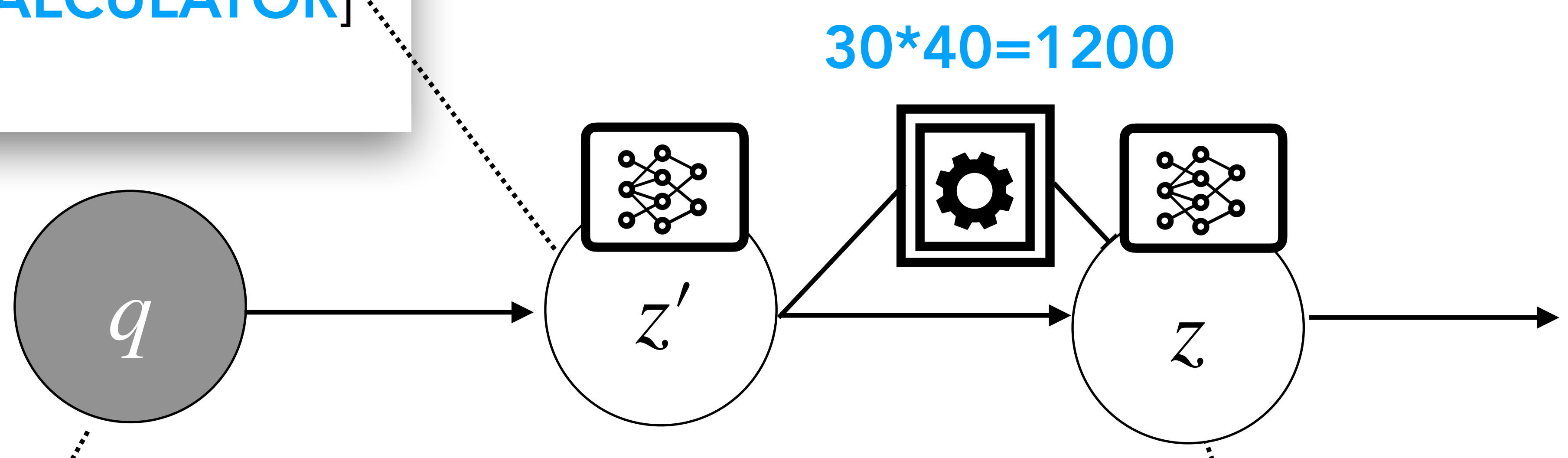
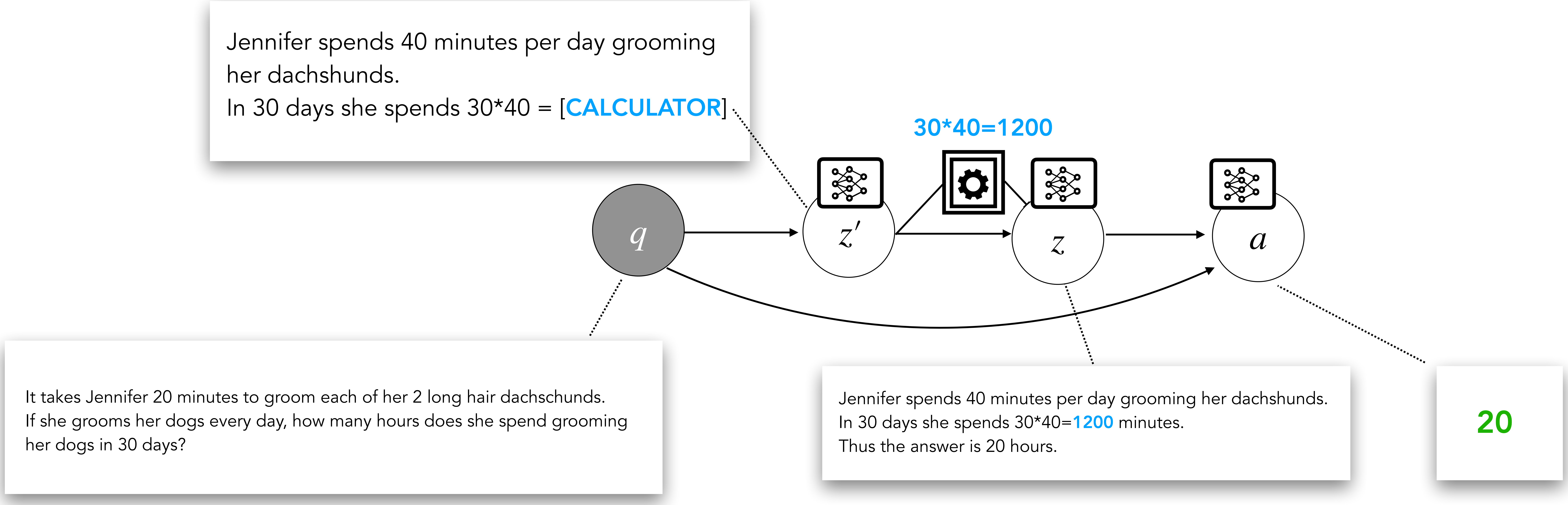
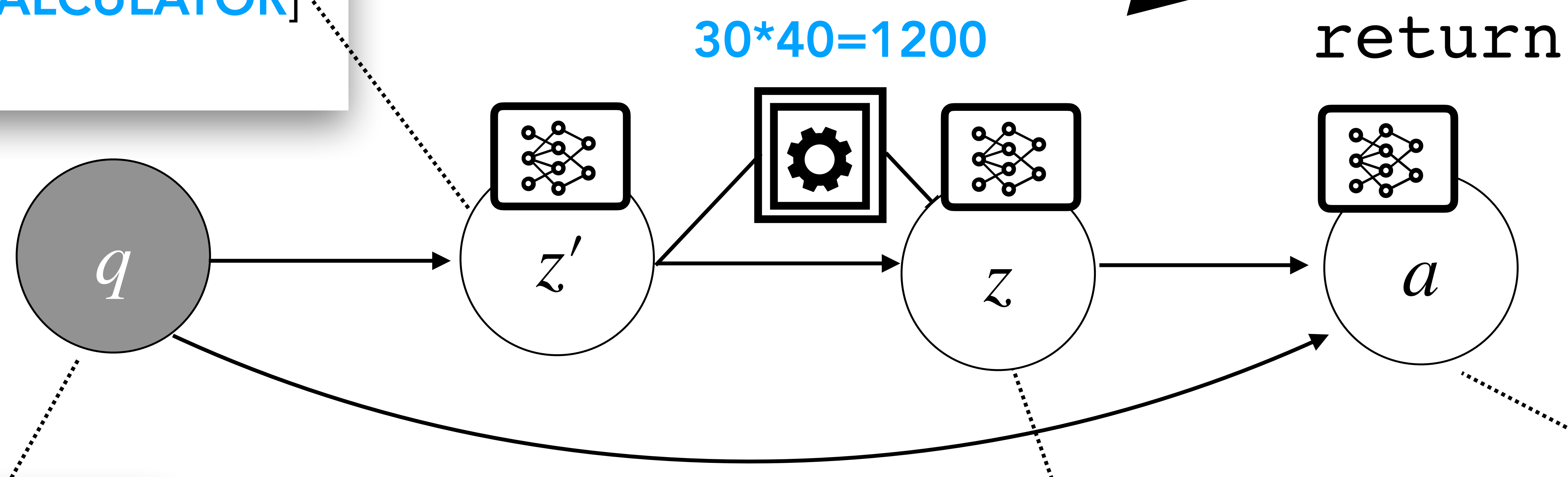
Modularity

Language Model Cascade [Dohan et al 2022]

- Vanilla language model
  - $a \sim p(a \mid q)$

- ```
  def qa():
     q = yield s(‘question’)
     a = yield s(‘answer’,
                 question=q)
     return a
  ```



Language Model Cascades
Dohan et al (Google)
ICML 2022, Beyond Bayes Workshop.
  ```

Modularity

Language Model Cascade [Dohan et al 2022]

- Prompted language model
  - $a \sim p(a \mid q; D)$



Sam memorized six more digits of pi than Carlos memorized… 10

…

Carly collected 7 starfish with 5 arms each and one seastar with 14 arms. How … 49

It takes Jennifer 20 minutes to groom each of her 2 long hair dachschunds. If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

52

Modularity

- Intermediate rationale z
- $$p(a \mid q) = \sum_z p(a \mid q, z) p(z \mid q)$$



[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity

- Intermediate rationale z
- $$p(a \mid q) = \sum_{z} p(a \mid q, z)\, p(z \mid q)$$



[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity

- Intermediate rationale z
  - $p(a \mid q) = \sum_z p(a \mid q, z) p(z \mid q)$

- Approximation:
  - $\hat{z} \sim p(z \mid q)$
    $\hat{a} \sim p(a \mid q, \hat{z})$



[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]()
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity | symbolic tools

- $p(a \mid q) = \sum p(a \mid q, z) p(z \mid \text{exec}(z'), q) p(z' \mid q)$

[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity | symbolic tools

- $p(a \mid q) = \sum p(a \mid q, z) p(z \mid \text{exec}(z'), q) p(z' \mid q)$

Jennifer spends 40 minutes per day grooming her dachshunds.
In 30 days she spends 30*40 = [CALCULATOR]



It takes Jennifer 20 minutes to groom each of her 2 long hair dachshunds.
If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems](#)
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity | symbolic tools

- $p(a \mid q) = \sum p(a \mid q, z) p(z \mid \text{exec}(z'), q) p(z' \mid q)$

Jennifer spends 40 minutes per day grooming her dachshunds.
In 30 days she spends 30*40 = [CALCULATOR]



It takes Jennifer 20 minutes to groom each of her 2 long hair dachschunds.
If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

Jennifer spends 40 minutes per day grooming her dachshunds.
In 30 days she spends 30*40=1200 minutes.
Thus the answer is 20 hours.

[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems](#)
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity | symbolic tools

- $p(a \mid q) = \sum p(a \mid q, z) p(z \mid \text{exec}(z'), q) p(z' \mid q)$



[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.

Modularity | symbolic tools

- $p(a \mid q) = \sum p(a \mid q, z) p(z \mid \text{exec}(z'), q) p(z' \mid q)$

- ```
  def qza():
      q = yield s('question')
      z = yield s('rationale',
                  question=q)
      z = execute(z)
      a = yield s('answer',
                  question=q,
                  rationale=z)
      return a
  ```

Jennifer spends 40 minutes per day grooming her dachshunds.
In 30 days she spends 30*40 = [CALCULATOR]



It takes Jennifer 20 minutes to groom each of her 2 long hair dachschunds.
If she grooms her dogs every day, how many hours does she spend grooming her dogs in 30 days?

Jennifer spends 40 minutes per day grooming her dachshunds.
In 30 days she spends 30*40=1200 minutes.
Thus the answer is 20 hours.

20

[Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems]
W. Ling, D. Yogatama, C. Dyer, P. Blunson
ACL 2017.
  ```

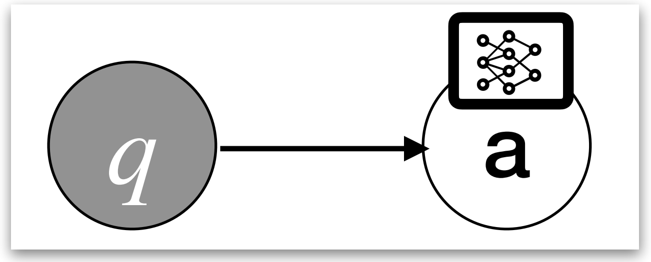
Modularity

- [Cobbe et al 2021]: GPT-3 + supervised rationales + calculator

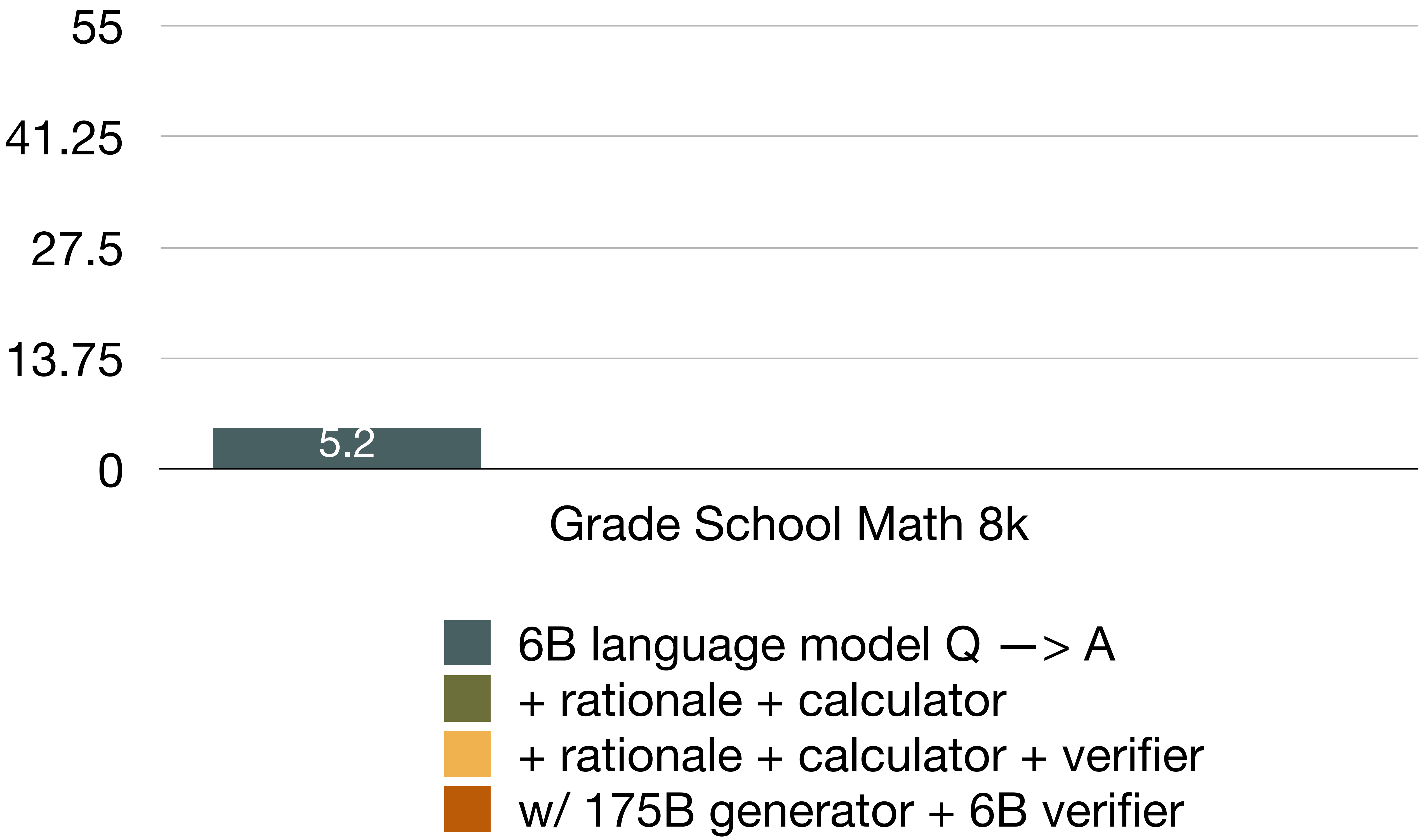
Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?
Solution: Tina buys 3 12-packs of soda, for 3*12= <<3*12=36>>36 sodas
6 people attend the party, so half of them is 6/2= <<6/2=3>>3 people
Each of those people drinks 3 sodas, so they drink 3*3=<<3*3=9>>9 sodas
Two people drink 4 sodas, which means they drink 2*4=<<4*2=8>>8 sodas
With one person drinking 5, that brings the total drank to 5+9+8+3= <<5+9+8+3=25>>25 sodas
As Tina started off with 36 sodas, that means there are 36-25=<<36-25=11>>11 sodas left
Final Answer: 11

Training Verifiers to Solve Math Word Problems
Cobbe et al (OpenAI).
arXiv 2021.

Modularity

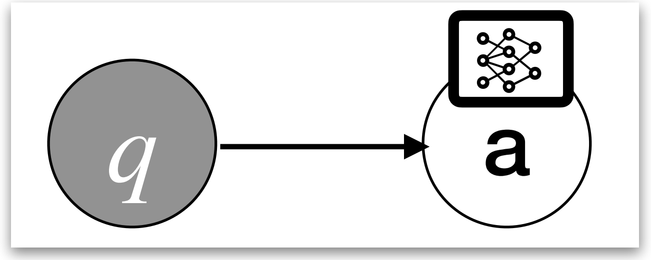


q
a

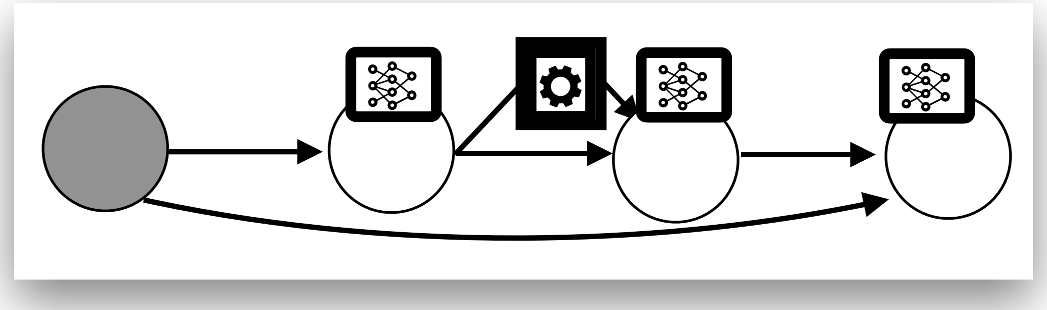


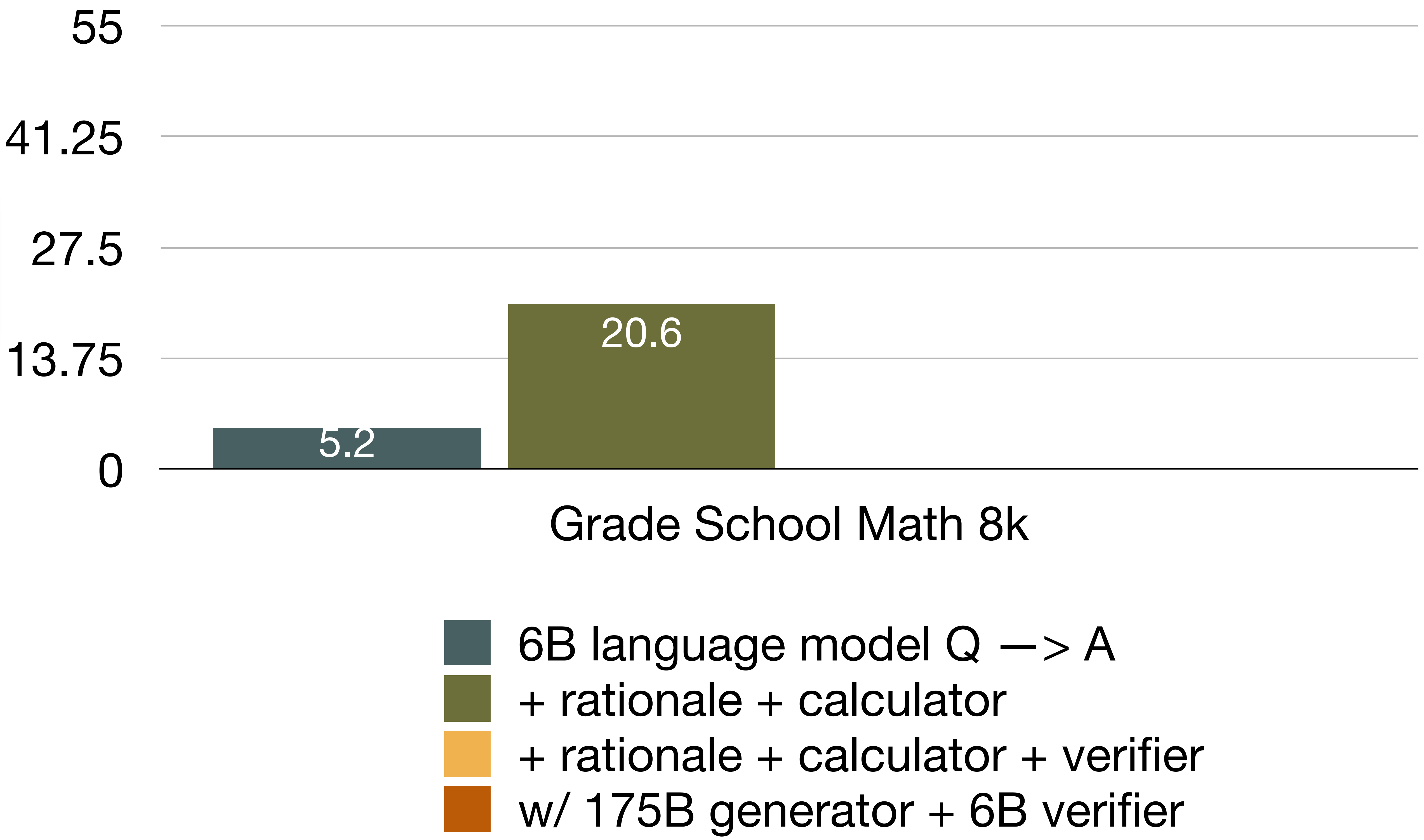
55
41.25
27.5
13.75
0
5.2
Grade School Math 8k
6B language model Q —> A
+ rationale + calculator
+ rationale + calculator + verifier
w/ 175B generator + 6B verifier

Modularity



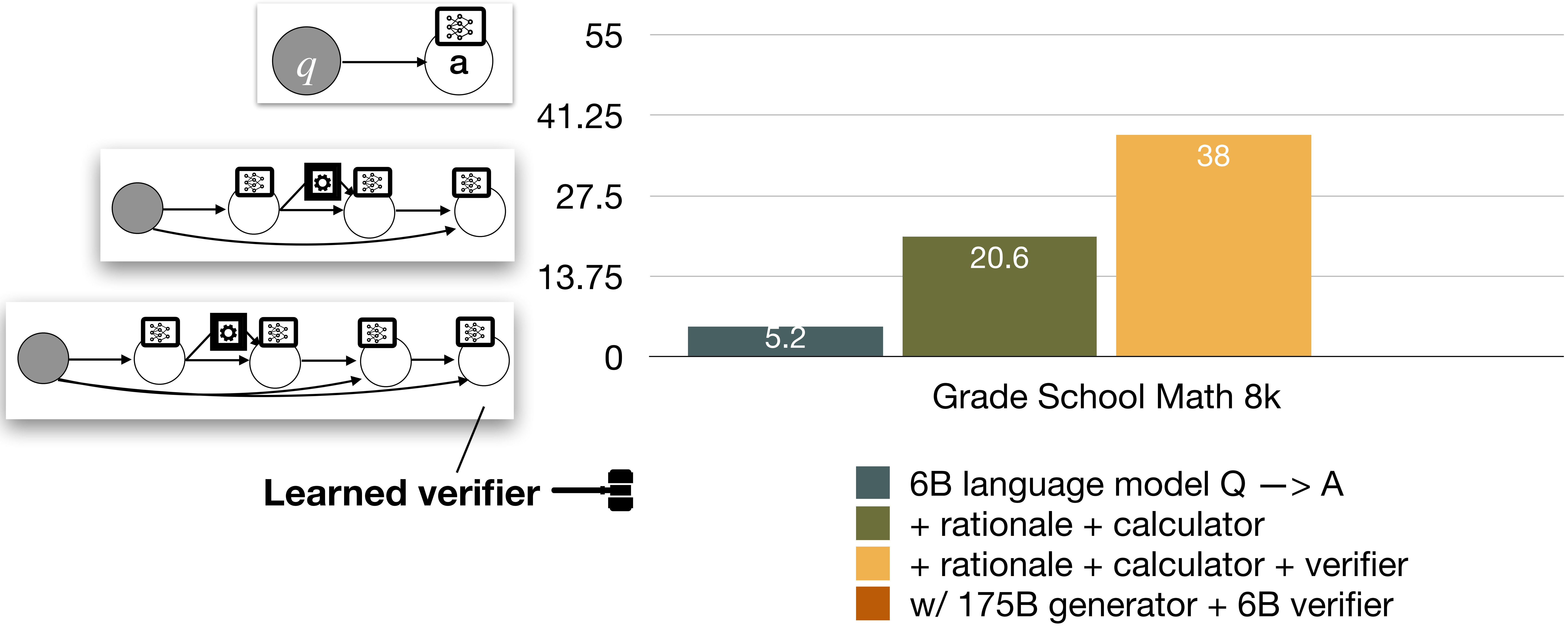
q
a





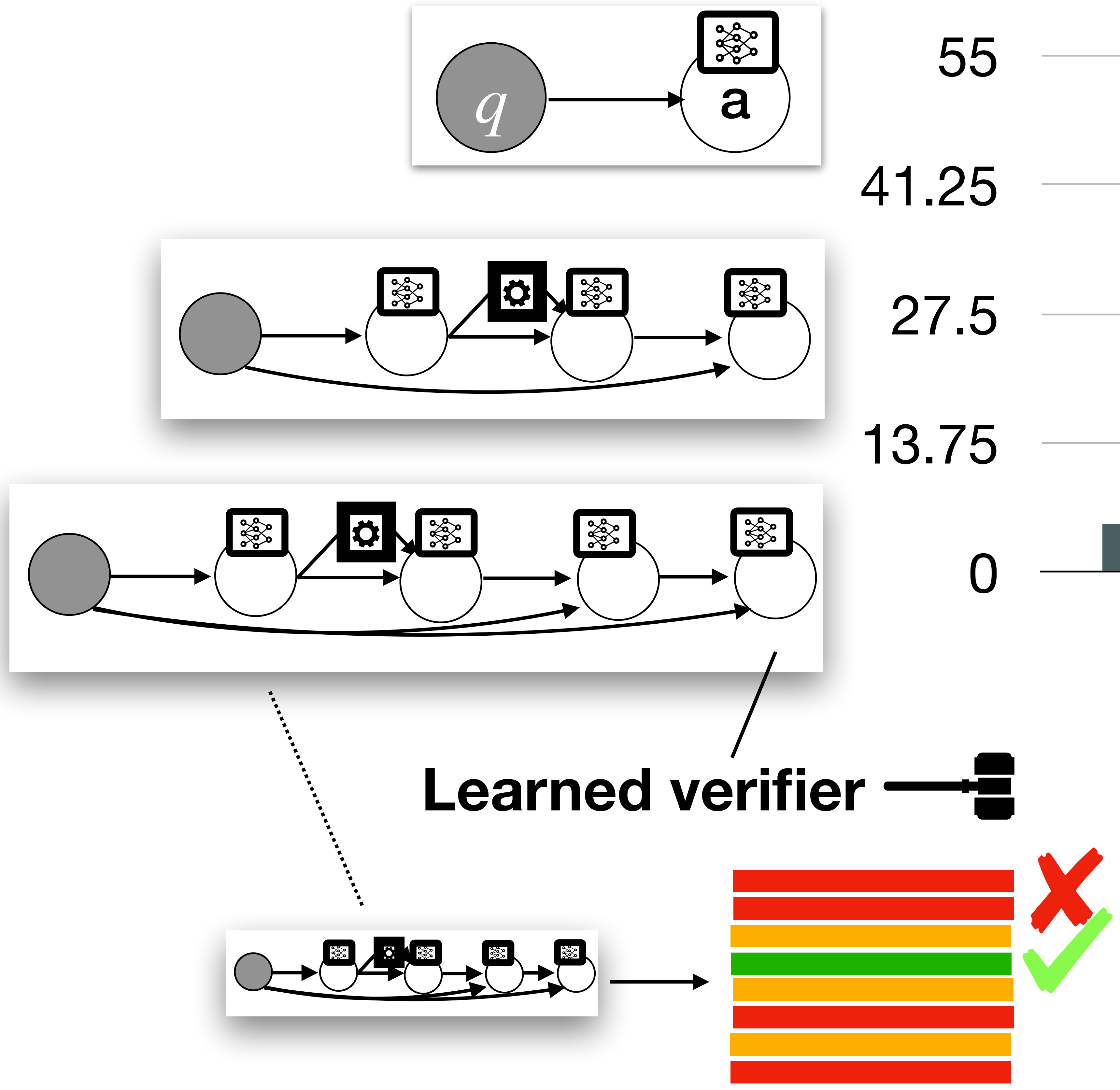
55
41.25
27.5
13.75
0
5.2
20.6
Grade School Math 8k
6B language model Q —> A
+ rationale + calculator
+ rationale + calculator + verifier
w/ 175B generator + 6B verifier

Modularity

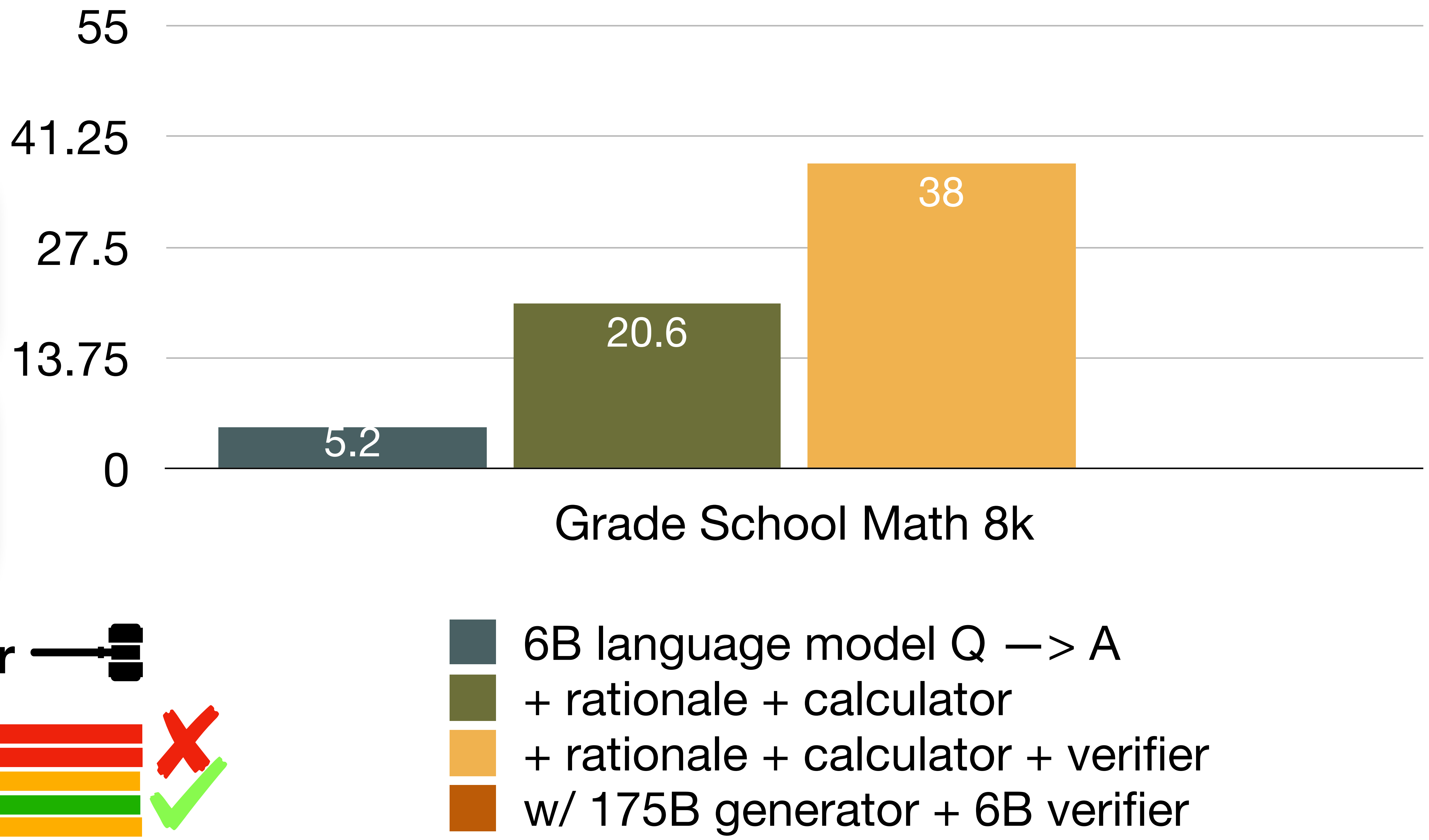


q
a
Learned verifier
55
41.25
27.5
13.75
0
5.2
20.6
38
Grade School Math 8k
6B language model Q —> A
+ rationale + calculator
+ rationale + calculator + verifier
w/ 175B generator + 6B verifier

Modularity

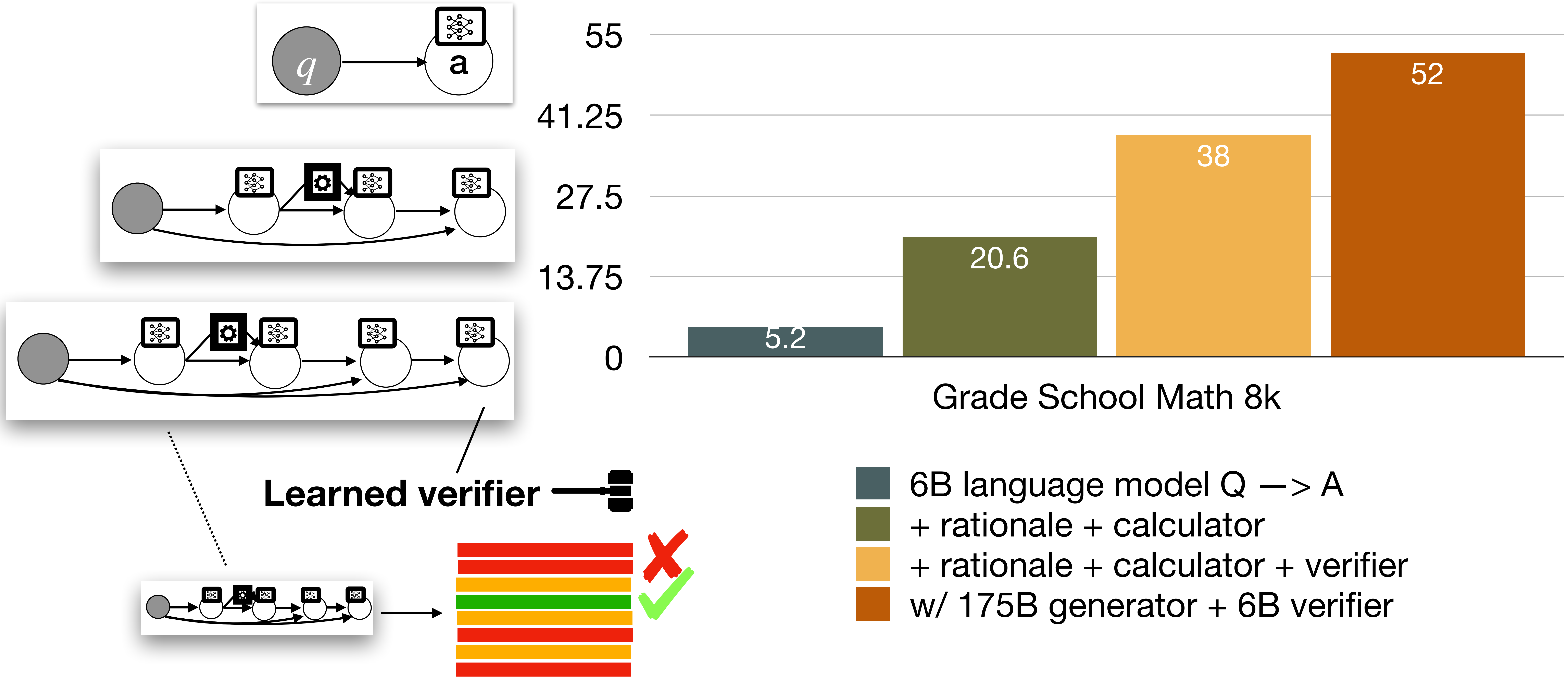


q
a
Learned verifier



55
41.25
27.5
13.75
0
5.2
20.6
38
Grade School Math 8k
6B language model Q —> A
+ rationale + calculator
+ rationale + calculator + verifier
w/ 175B generator + 6B verifier

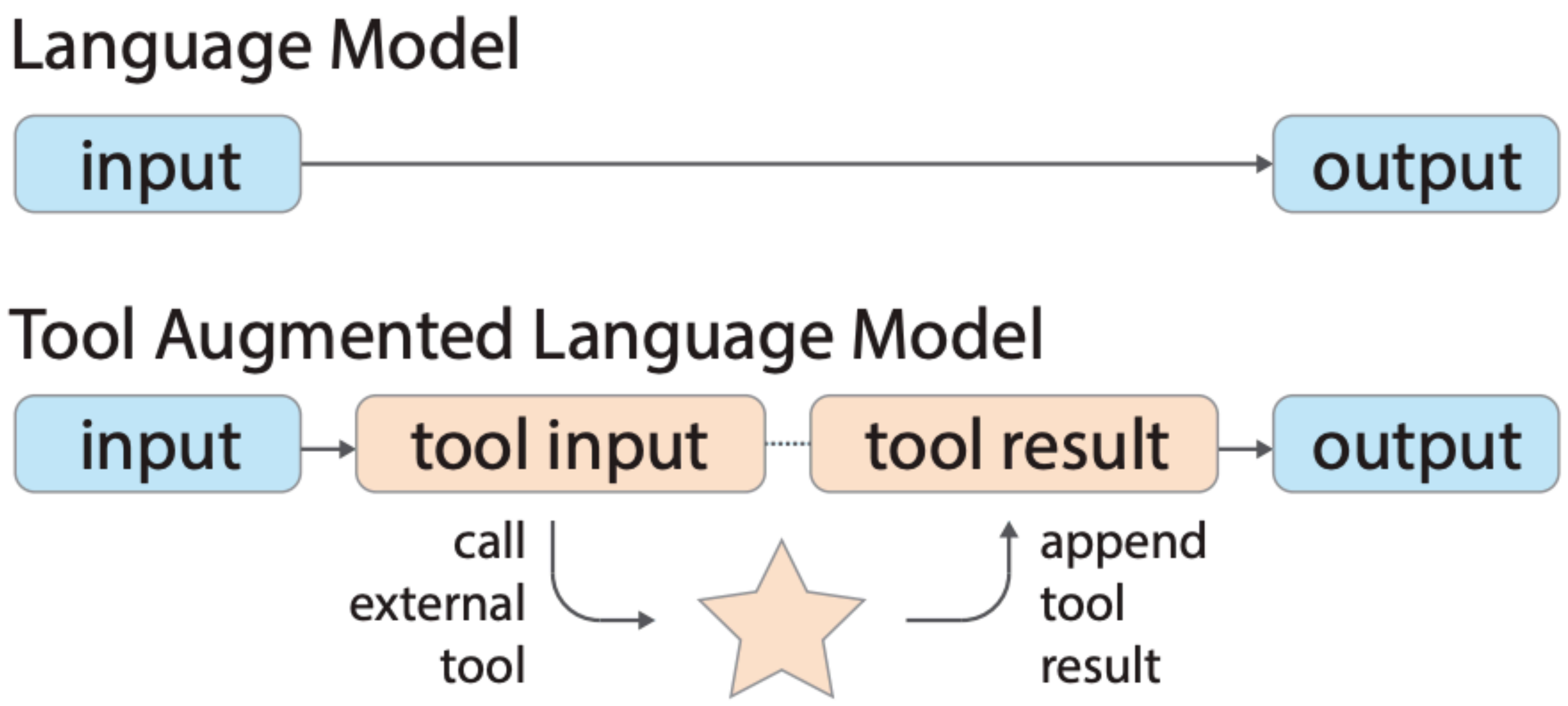
Modularity



q
a
Learned verifier
55
41.25
27.5
13.75
0
5.2
20.6
38
52
Grade School Math 8k
6B language model Q —> A
+ rationale + calculator
+ rationale + calculator + verifier
w/ 175B generator + 6B verifier

Modularity | other tools

- Tool Augmented Language Models [Parisi et al 2022]



Modularity | other tools

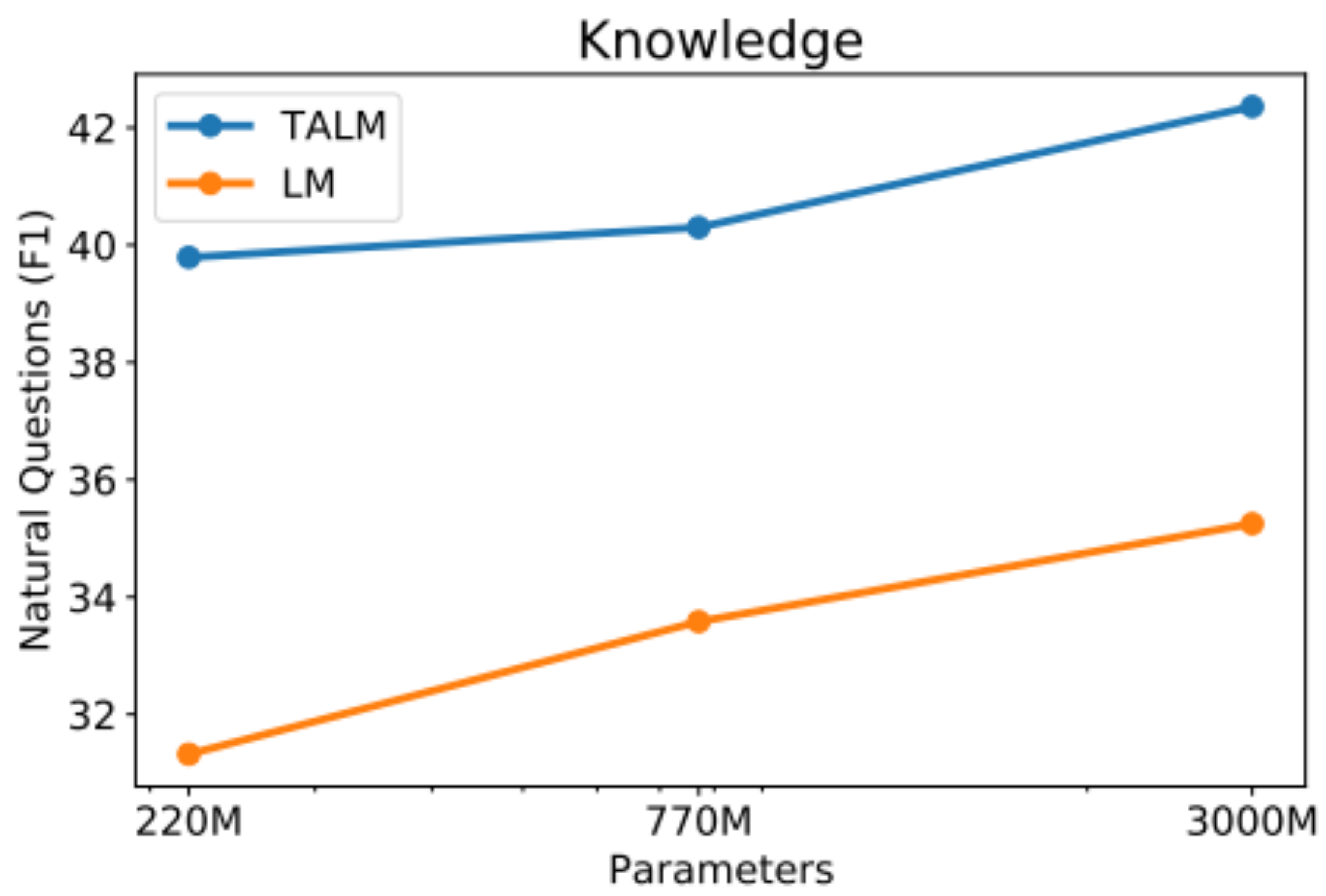
- Tool Augmented Language Models [Parisi et al 2022]
  - **Tool:** Retrieval/web-search

Question: when are hops added in brewing process?
Short Answer: The boiling process.

|question when are hops added in brewing process? |search brewing process |result The boiling process is where chemical reactions take place...including |output The boiling process.

Modularity | other tools

- Tool Augmented Language Models [Parisi et al 2022]
  - **Tool:** Retrieval/web-search



Modularity | other tools

- Lila Benchmark [Mishra et al 2022]
  Unifies 20 math datasets:
  - ‘**Rationale**’: python program
  - **Tools**: libraries (numpy, …), standard Python (variables, …)

Problem:
The pirates plan to explore 4 islands. Two islands require walking 20 miles per day while the other two islands require 25 miles per day. How many miles will they have to walk if it takes 1.5 days to explore each island?

Program:

```
a=20*2
b=25*2
c=a+b
d=c*1.5
answer=d
print(answer)
# ==> 135.0
```

Grade School Math (GSM) 8k

Problem:

Compute the nullity of $\begin{pmatrix} -9 \\ -2 \\ 3 \\ -\frac{1}{2} \end{pmatrix}$.

Program:

```
import numpy as np
a = np.array([[-9], [-2], [3],
             [-(1/2)]])
r = np.linalg.matrix_rank(a)
print(len(a[0]) - r)
# ==> 0.0
```

Linear Algebra

[Lila: A Unified Benchmark for Mathematical Reasoning]
S. Mishra, M. Finlayson, P. Lu, L. Tang, S. Welleck, C. Baral, T. Rajpurohit, O. Tafjord, A. Sabharwal, P. Clark, A. Kalyan
EMNLP 2022.

Modularity | other tools

- Lila Benchmark
  - Program + execution > answer

Dimension	Neo-A		Neo-P	
	IID	OOD	IID	OOD
Math ability	0.191	0.129	**0.445**	**0.188**
Language	0.189	0.147	**0.429**	**0.246**
Format	0.246	0.382	**0.372**	**0.404**
Knowledge	0.206	0.143	**0.331**	**0.213**
Average	0.208	0.200	**0.394**	**0.263**

Modularity | other tools

- Lila Benchmark
  - Program + execution > answer
    - In-domain & OOD generalization

Dimension	Neo-A		Neo-P	
	IID	OOD	IID	OOD
Math ability	0.191	0.129	**0.445**	**0.188**
Language	0.189	0.147	**0.429**	**0.246**
Format	0.246	0.382	**0.372**	**0.404**
Knowledge	0.206	0.143	**0.331**	**0.213**
Average	0.208	0.200	**0.394**	**0.263**

Modularity | bridging informal+formal reasoning

Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, where 1 is positive and 1 is negative. The absolute values of these two solutions are also both less than 2. This proves the base case.

Now assume that for some positive integer k, $P_k(x)$ has 2^k distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and 2^{k-1} are negative.

Choose a root r of $P_{k+1}(x)$. Let $P_1(r) = s$, where s is a real root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. Therefore all of the roots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

MATHEMATICAL ASSOCIATION OF AMERICA

Modularity | bridging informal+formal reasoning

Natural language mathematics

Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, where 1 is positive and 1 is negative. The absolute values of these two solutions are also both less than 2. This proves the base case.

Now assume that for some positive integer k, $P_k(x)$ has 2^k distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and 2^{k-1} are negative.

Choose a root r of $P_{k+1}(x)$. Let $P_1(r) = s$, where s is a real root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. Therefore all of the roots of P_{k+1} are real and have absolute values less than 2.

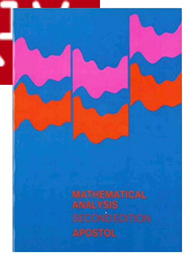
Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

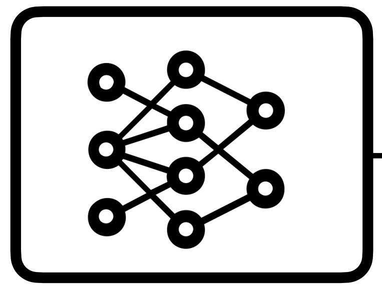
Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

Modularity | bridging informal+formal reasoning

Natural language mathematics





Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, where 1 is positive and 1 is negative. The absolute values of these two solutions are also both less than 2. This proves the base case.

Now assume that for some positive integer k, $P_k(x)$ has 2^k distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and 2^{k-1} are negative.

Choose a root r of $P_{k+1}(x)$. Let $P_1(r) = s$, where s is a real root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. Therefore all of the roots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

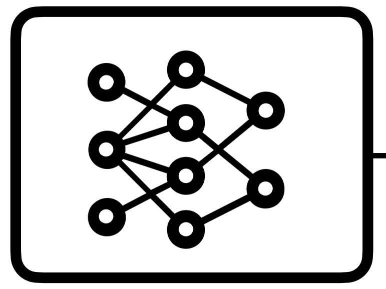
Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

Modularity | bridging informal+formal reasoning

Natural language mathematics

Flexibility
Data



Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, where 1 is positive and 1 is negative. The absolute values of these two solutions are also both less than 2. This proves the base case.

Now assume that for some positive integer k, $P_k(x)$ has 2^k distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and 2^{k-1} are negative.

Choose a root r of $P_{k+1}(x)$. Let $P_1(r) = s$, where s is a real root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. Therefore all of the roots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

Modularity | bridging informal+formal reasoning

Natural language mathematics

Flexibility
Data

Verifiability
Grounding

Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher … s negative. The absolute values of these two solutions are also both less than 2. This … e.

Now assume that for some positive inte… distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and …

Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ … root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. … oots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

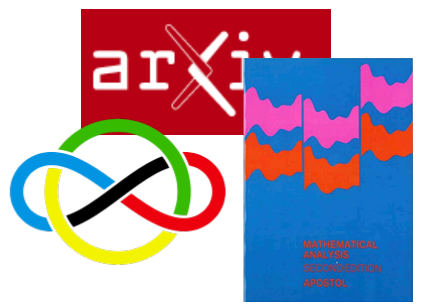
Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

Modularity | bridging informal+formal reasoning

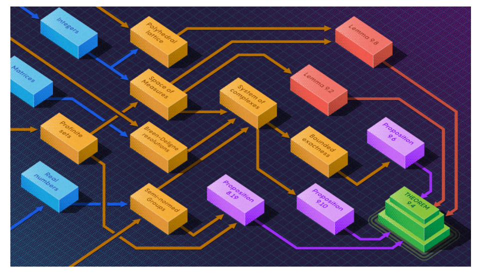
Natural language mathematics

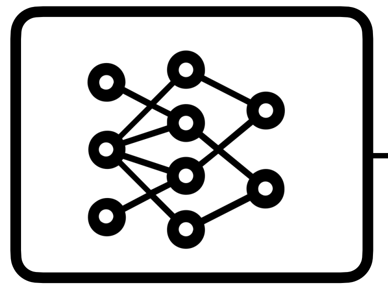
Flexibility
Data



Verifiability
Grounding

Formalized mathematics





Problem

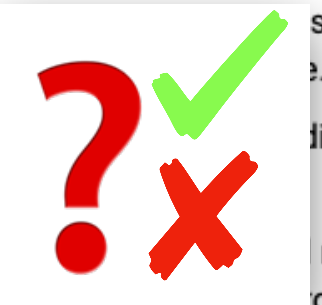
Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher s negative. The absolute values of these two solutions are also both less than 2. This e.

Now assume that for some positive inte istinct real solutions with absolute values less than 2, where 2^{k-1} are positive and



Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. oots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

```
theorem aime_1984_p1
  (u : ℕ → ℚ)
  (h₀ : ∀ n, u (n + 1) = u n + 1)
  (h₁ : ∑ k in finset.range 98, u k.succ = 137) :
  ∑ k in finset.range 49, u (2 * k.succ) = 93 :=
begin
  rw finset.sum_eq_multiset_sum,
  dsimp [finset.range] at h₁,
  simp [h₀],
  ring,
  norm_num at h₁,
  norm_num,
  apply eq_of_sub_eq_zero,
  { simp only [*, abs_of_pos, add_zero] at *, linarith },
end
```

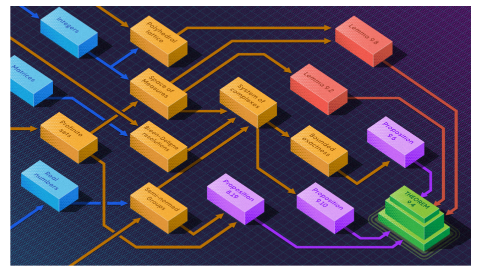
Modularity | bridging informal+formal reasoning

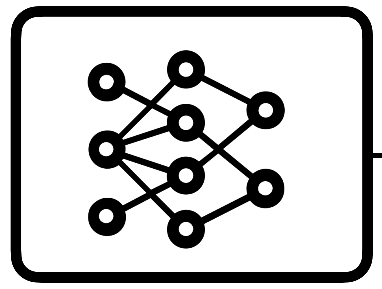
Natural language mathematics

Flexibility
Data

Verifiability
Grounding

Formalized mathematics





Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher … s negative. The absolute values of these two solutions are also both less than 2. This … e.

Now assume that for some positive inte … distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and …

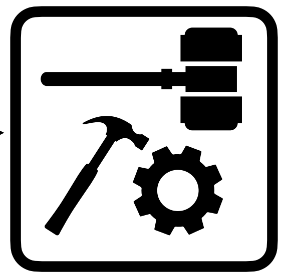
Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ … root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. … oots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

```
theorem aime_1984_p1
  (u : ℕ → ℚ)
  (h₀ : ∀ n, u (n + 1) = u n + 1)
  (h₁ : ∑ k in finset.range 98, u k.succ = 137) :
  ∑ k in finset.range 49, u (2 * k.succ) = 93 :=
begin
  rw finset.sum_eq_multiset_sum,
  dsimp [finset.range] at h₁,
  simp [h₀],
  ring,
  norm_num at h₁,
  norm_num,
  apply eq_of_sub_eq_zero,
  { simp only [*, abs_of_pos, add_zero] at *, linarith },
end
```



LEAN THEOREM PROVER
Microsoft Research

Isabelle

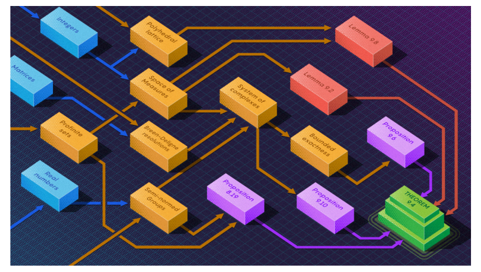
Modularity | bridging informal+formal reasoning

Natural language mathematics

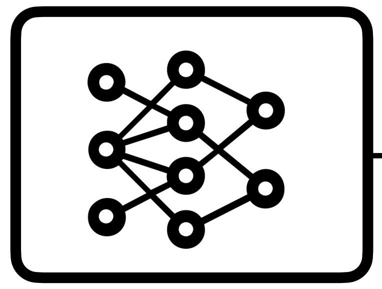
Flexibility
Data

Verifiability
Grounding

Formalized mathematics



Verifiability
Grounding



Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher s negative. The absolute values of these two solutions are also both less than 2. This e.

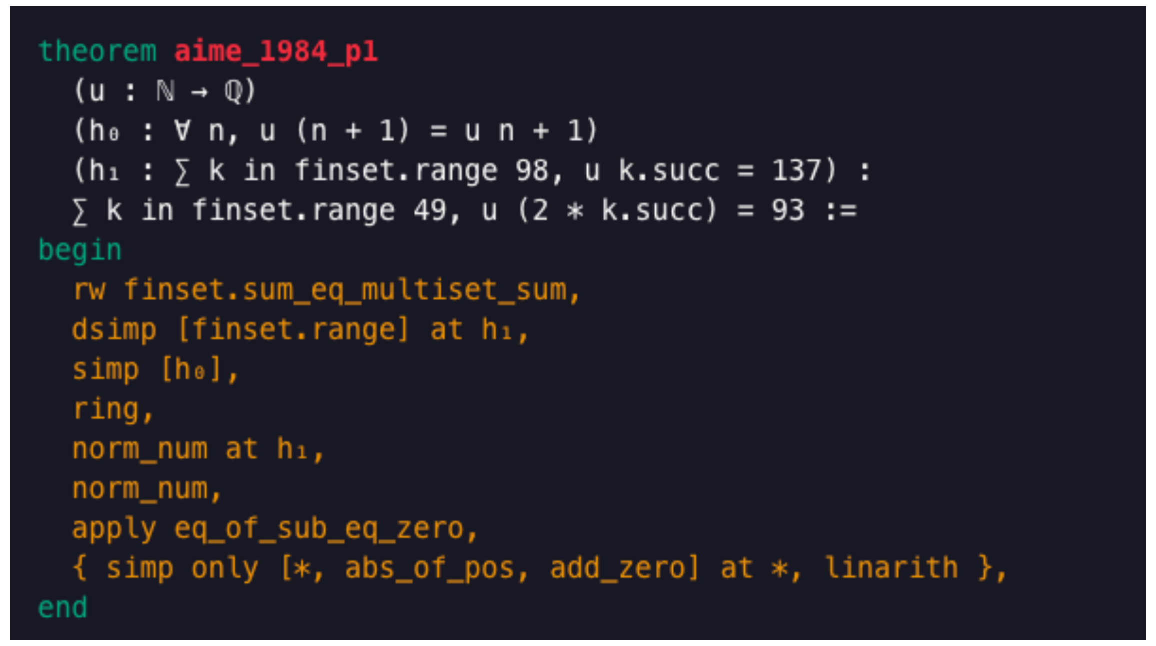
Now assume that for some positive inte distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and

Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. oots of P_{k+1} are real and have absolute values less than 2.

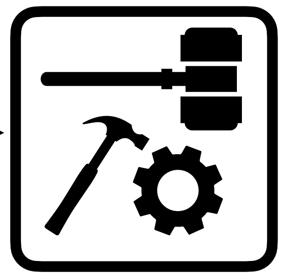
Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

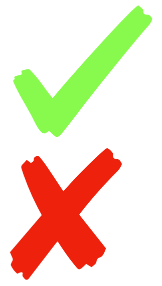
Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.



```
theorem aime_1984_p1
  (u : ℕ → ℚ)
  (h₀ : ∀ n, u (n + 1) = u n + 1)
  (h₁ : ∑ k in finset.range 98, u k.succ = 137) :
  ∑ k in finset.range 49, u (2 * k.succ) = 93 :=
begin
  rw finset.sum_eq_multiset_sum,
  dsimp [finset.range] at h₁,
  simp [h₀],
  ring,
  norm_num at h₁,
  norm_num,
  apply eq_of_sub_eq_zero,
  { simp only [*, abs_of_pos, add_zero] at *, linarith },
end
```





LEAN THEOREM PROVER
Microsoft Research

Isabelle

Modularity | bridging informal+formal reasoning

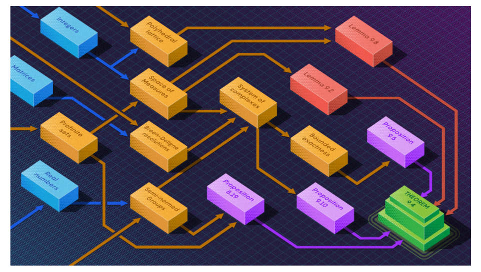
Natural language mathematics

Flexibility
Data

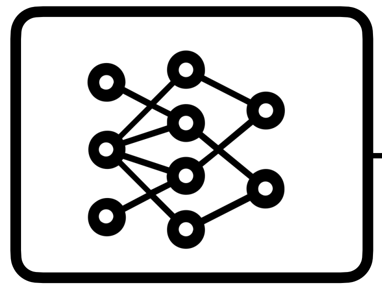
Verifiability
Grounding

Formalized mathematics

Flexibility
Data



Verifiability
Grounding



Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher s negative. The absolute values of these two solutions are also both less than 2. This e.

Now assume that for some positive inte distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and

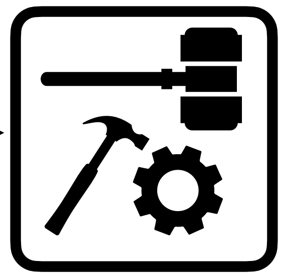
Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. oots of P_{k+1} are real and have absolute values less than 2.

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

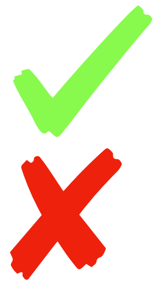
Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

```
theorem aime_1984_p1
  (u : ℕ → ℚ)
  (h₀ : ∀ n, u (n + 1) = u n + 1)
  (h₁ : ∑ k in finset.range 98, u k.succ = 137) :
  ∑ k in finset.range 49, u (2 * k.succ) = 93 :=
begin
  rw finset.sum_eq_multiset_sum,
  dsimp [finset.range] at h₁,
  simp [h₀],
  ring,
  norm_num at h₁,
  norm_num,
  apply eq_of_sub_eq_zero,
  { simp only [*, abs_of_pos, add_zero] at *, linarith },
end
```



LEAN THEOREM PROVER
Microsoft Research
Isabelle



Modularity | bridging informal+formal reasoning

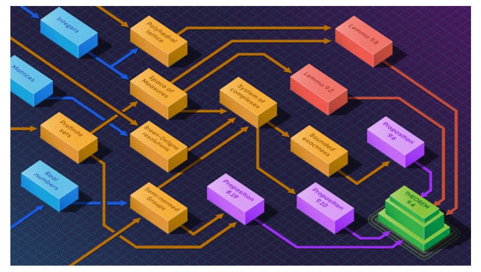
Natural language mathematics

Flexibility
Data

Verifiability
Grounding

Formalized mathematics

Flexibility
Data



Verifiability
Grounding

Best of both worlds?

Problem

Let $P_1(x) = x^2 - 2$ and $P_j(x) = P_1(P_{j-1}(x))$ for $j = 2, \ldots$ Prove that for any positive integer n the roots of the equation $P_n(x) = x$ are all real and distinct.

Solution

I shall prove by induction that $P_n(x)$ has 2^n distinct real solutions, where 2^{n-1} are positive and 2^{n-1} are negative. Also, for ever root r, $|r| < 2$.

Clearly, $P_1(x)$ has 2 real solutions, wher … s negative. The absolute values of these two solutions are also both less than 2. This … .

Now assume that for some positive inte… … distinct real solutions with absolute values less than 2, where 2^{k-1} are positive and …

Choose a root r of $P_{k+1}(x)$. Let $P_1(r)$ … root of $P_k(x)$. We have that $-2 < s < 2$, so $0 < r^2 < 4$, so r is real and $|r| < 2$. … oots of P_{k+1} are real and have absolute values less than 2.

?

Note that the function $P_{k+1}(x)$ is an even function, since $P_1(x)$ is an even function. Therefore half of the roots of P_{k+1} are positive, and half are negative.

Now assume for the sake of contradiction that $P_{k+1}(x)$ has a double root r. Let $P_1(r) = s$. Then there exists exactly one real number r such that $r^2 - 2 = s$. The only way that this could happen is when $s + 2 = 0$, or $s = -2$. However, $|s| < 2$ from our inductive hypothesis, so this is a contradiction. Therefore $P_{k+1}(x)$ has no double roots. This proves that that the roots of $P_{k+1}(x)$ are distinct.

This completes the inductive step, which completes the inductive proof.

```
theorem aime_1984_p1
  (u : ℕ → ℚ)
  (h₀ : ∀ n, u (n + 1) = u n + 1)
  (h₁ : ∑ k in finset.range 98, u k.succ = 137) :
  ∑ k in finset.range 49, u (2 * k.succ) = 93 :=
begin
  rw finset.sum_eq_multiset_sum,
  dsimp [finset.range] at h₁,
  simp [h₀],
  ring,
  norm_num at h₁,
  norm_num,
  apply eq_of_sub_eq_zero,
  { simp only [*, abs_of_pos, add_zero] at *, linarith },
end
```

LEAN THEOREM PROVER
Microsoft Research

Isabelle

Modularity | sketching

- Draft-Sketch-Prove [Jiang et al 2022]

Statement

If $\gcd(n, 4) = 1$ and
$\operatorname{lcm}(n, 4) = 28$,
show that n is 7.

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching

- Draft-Sketch-Prove [Jiang et al 2022]

Statement

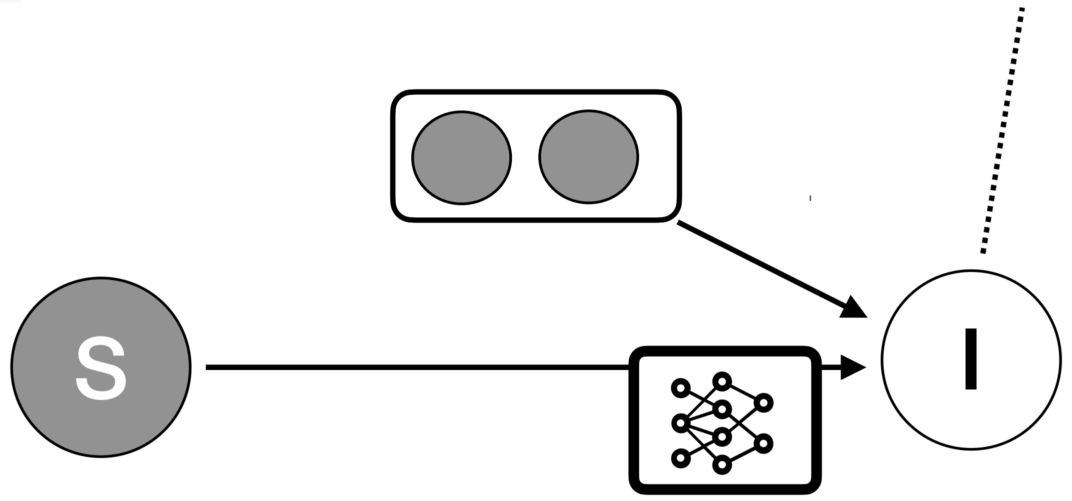
If $\gcd(n, 4) = 1$ and $\mathrm{lcm}(n, 4) = 28$, show that n is 7.

Informal proof

We know that $\gcd(a, b) \cdot \mathrm{lcm}(a, b) = ab$, hence $1 \cdot 28 = n \cdot 4$.

Then $n = 1 \cdot 28/4 = 7$,

completing the proof. ■

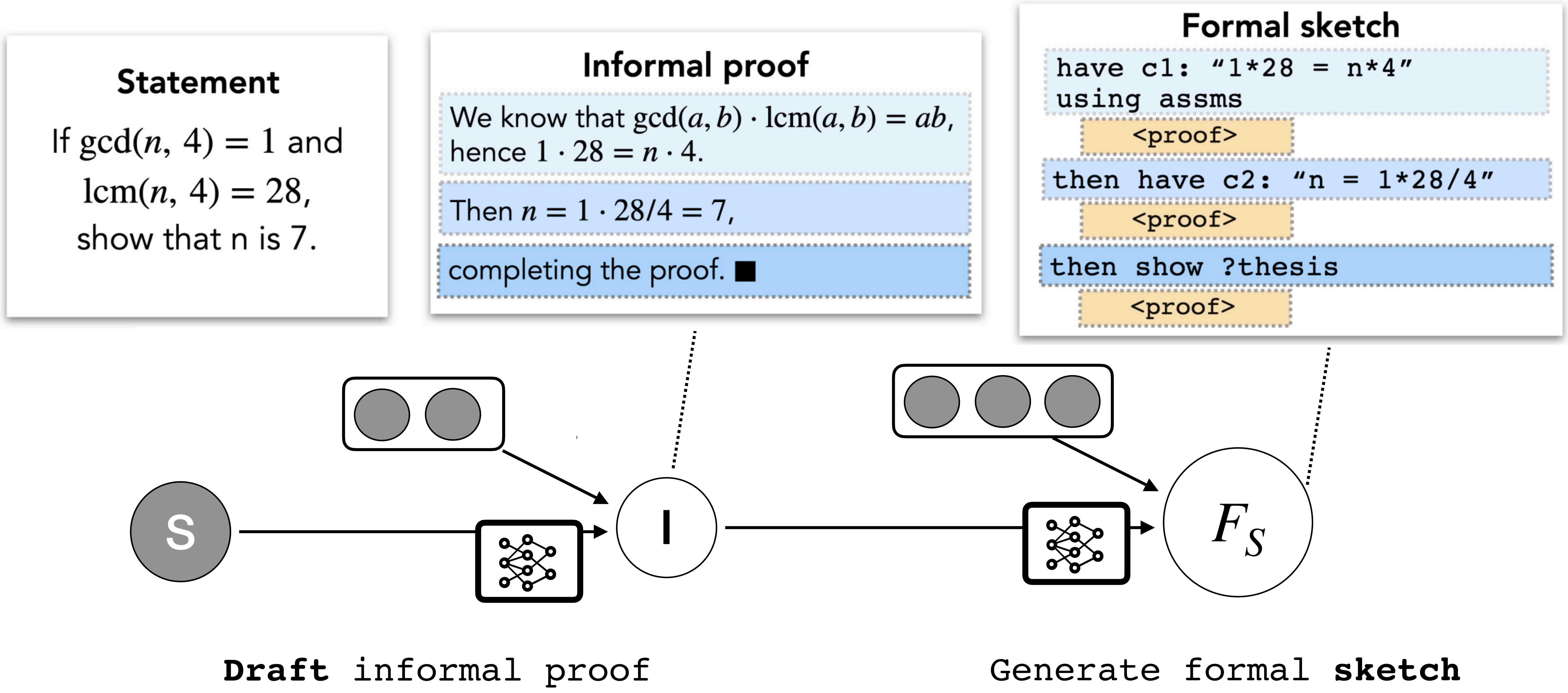


Draft informal proof

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching

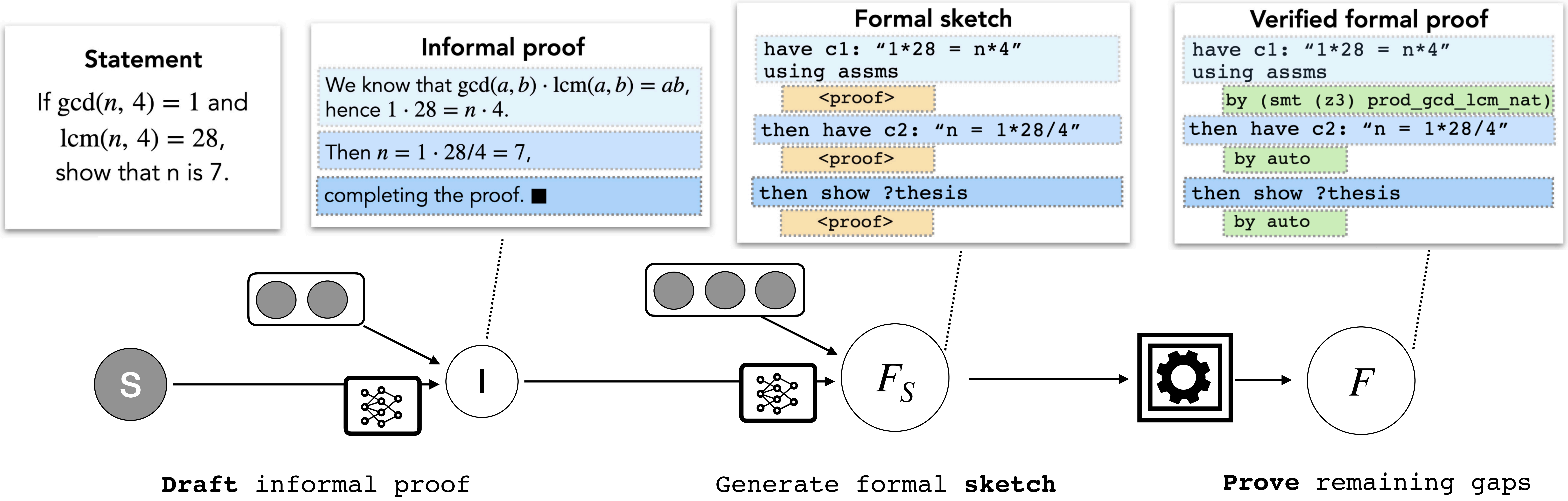
- Draft-Sketch-Prove [Jiang et al 2022]



Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching

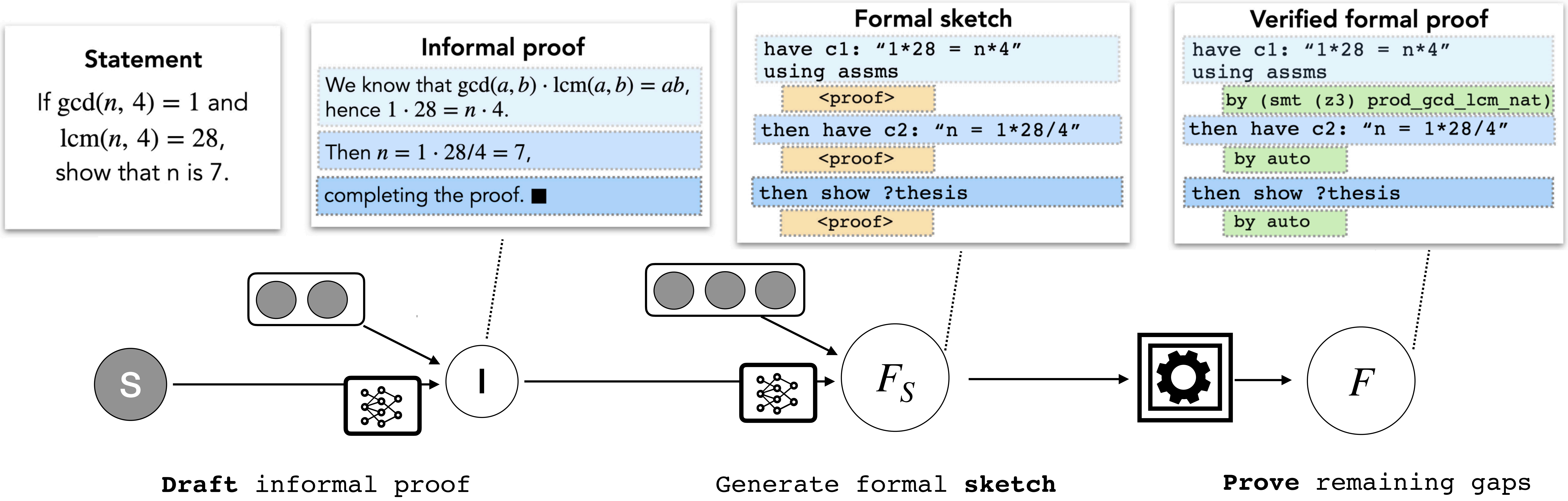
- Draft-Sketch-Prove [Jiang et al 2022]

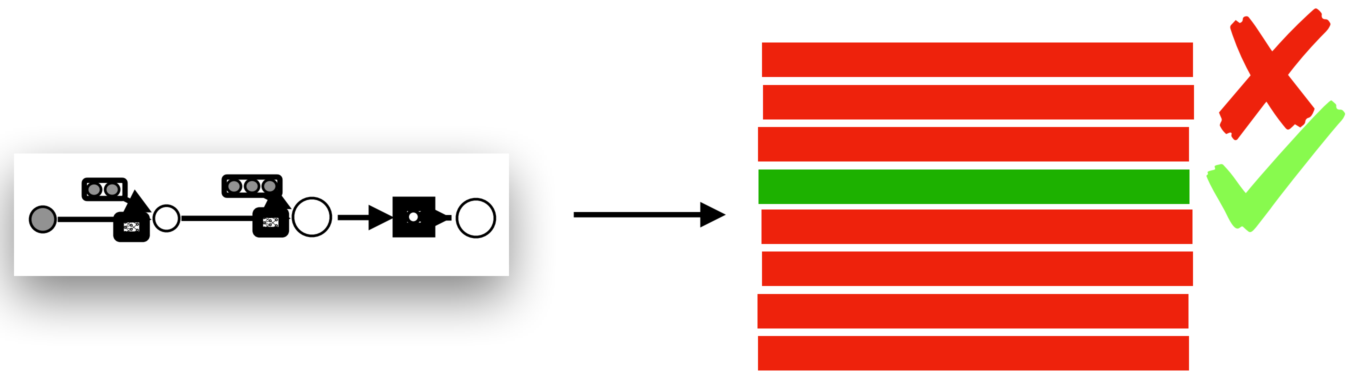


Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching

- Draft-Sketch-Prove [Jiang et al 2022]

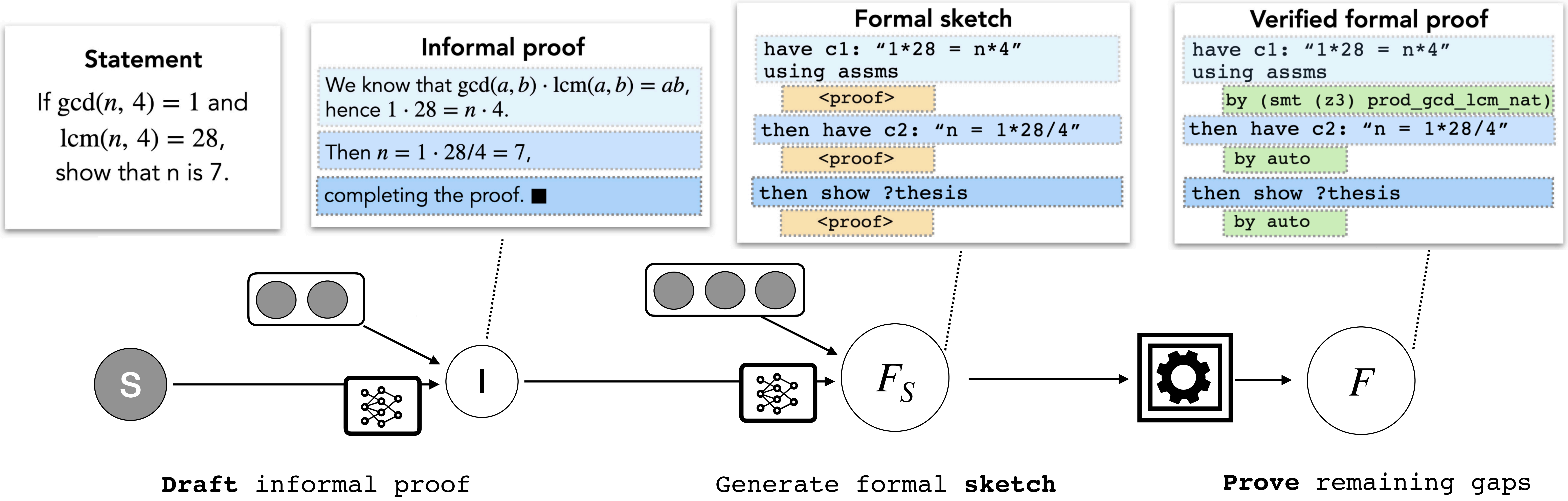




Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

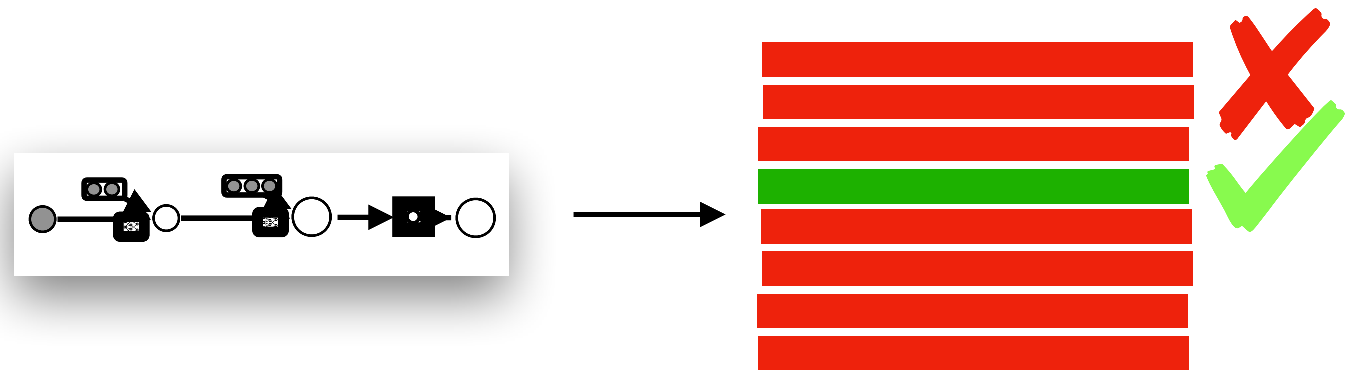
Modularity | sketching

- Draft-Sketch-Prove [Jiang et al 2022]



- **Neural**: informal proof + high-level *formal sketch*

- **Symbolic tools**: formal prover & verifier



Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):
We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$21n+4 = 1 \cdot (14n+3) + 7n + 1$$
$$14n+3 = 2 \cdot (7n+1) + 1.$$

Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

Formal Proof:

```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto  [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto  [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto  [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))  [ATP]
  then show ?thesis
    using c1
    by blast  [ATP]
qed
```

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):
We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$
$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

Formal Proof:

```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto  [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto  [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto  [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))  [ATP]
  then show ?thesis
    using c1
    by blast  [ATP]
qed
```

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Neural natural prover

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):

We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$
$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

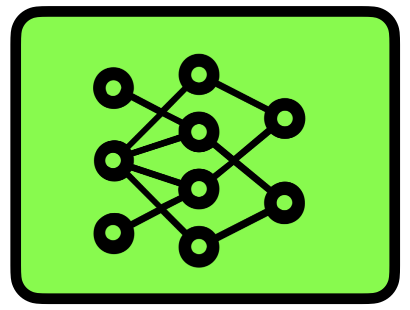
Formal Proof:

```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto  [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto  [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto  [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))  [ATP]
  then show ?thesis
    using c1
    by blast  [ATP]
qed
```

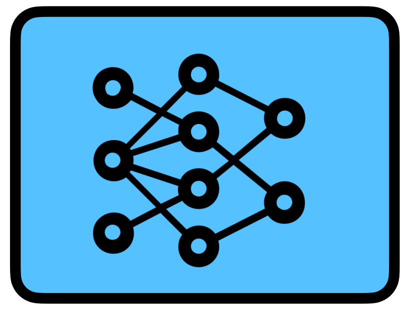
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Neural natural prover



Neural sketcher



Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):

We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$
$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

Since $\gcd(7n + 1, 1) = 1$, we have $\gcd(21n + 4, 14n + 3) = 1$.

Formal Proof:

```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto   [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto   [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto   [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))   [ATP]
  then show ?thesis
    using c1
    by blast   [ATP]
qed
```

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Neural natural prover

Neural sketcher

Symbolic prover

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):

We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$21n+4 = 1 \cdot (14n+3) + 7n + 1$$
$$14n+3 = 2 \cdot (7n+1) + 1.$$

Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

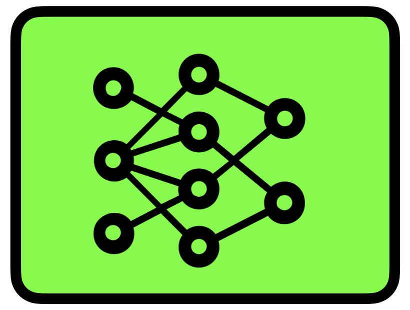
Formal Proof:

```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto   [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto   [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto   [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))   [ATP]
  then show ?thesis
    using c1
    by blast   [ATP]
qed
```

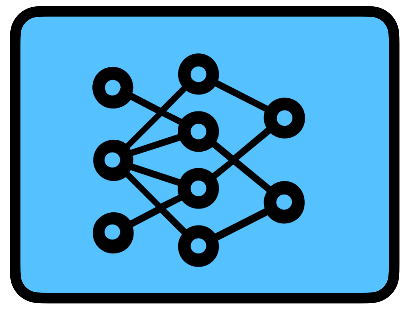
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

International Math Olympiad 1959 P.1

Neural natural prover



Neural sketcher



Symbolic prover

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):

We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

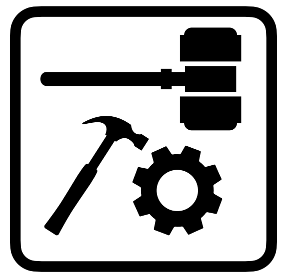
$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$
$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

Formal Proof:

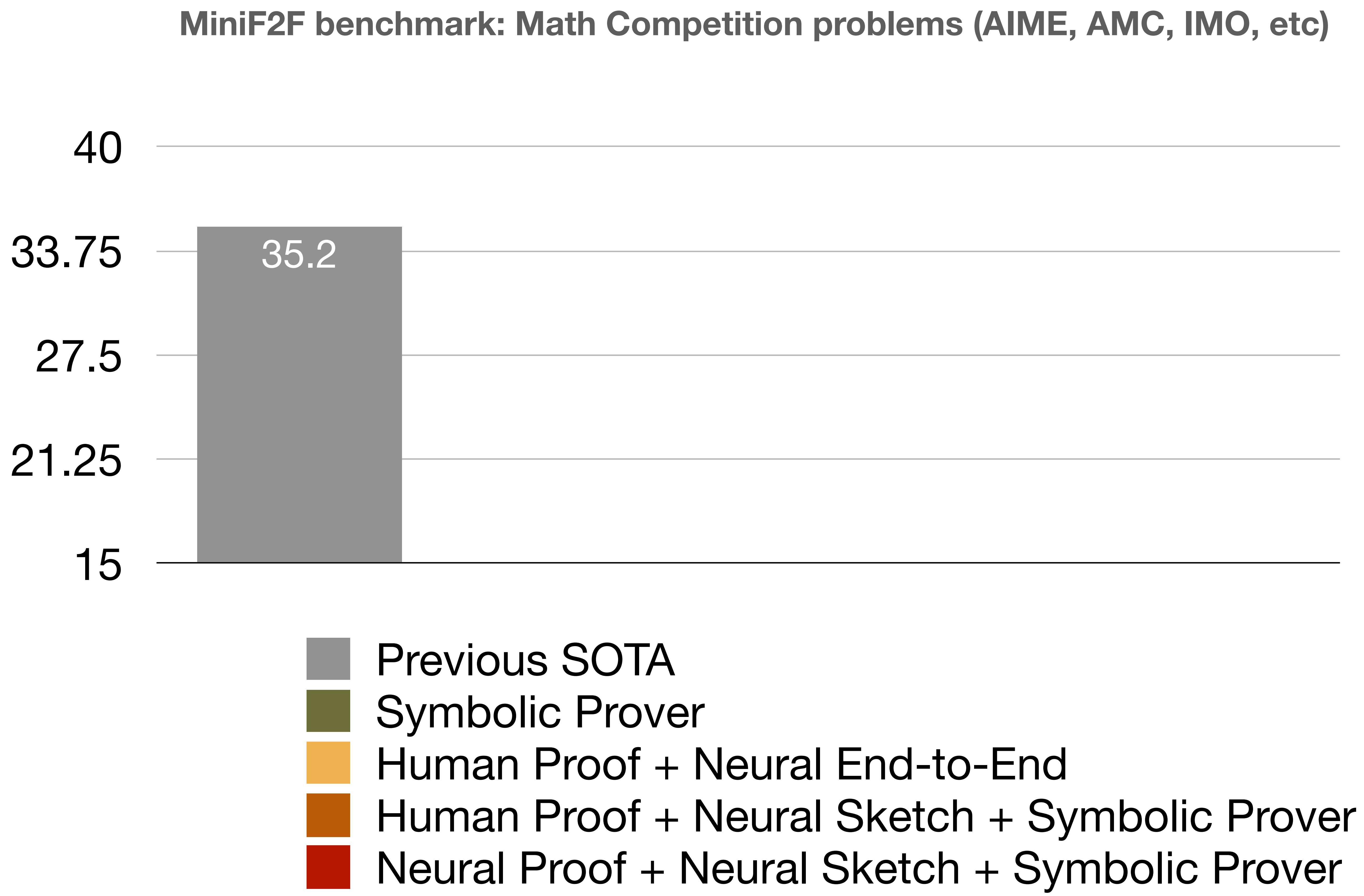
```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto   [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto   [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto   [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))   [ATP]
  then show ?thesis
    using c1
    by blast   [ATP]
qed
```

Symbolic kernel



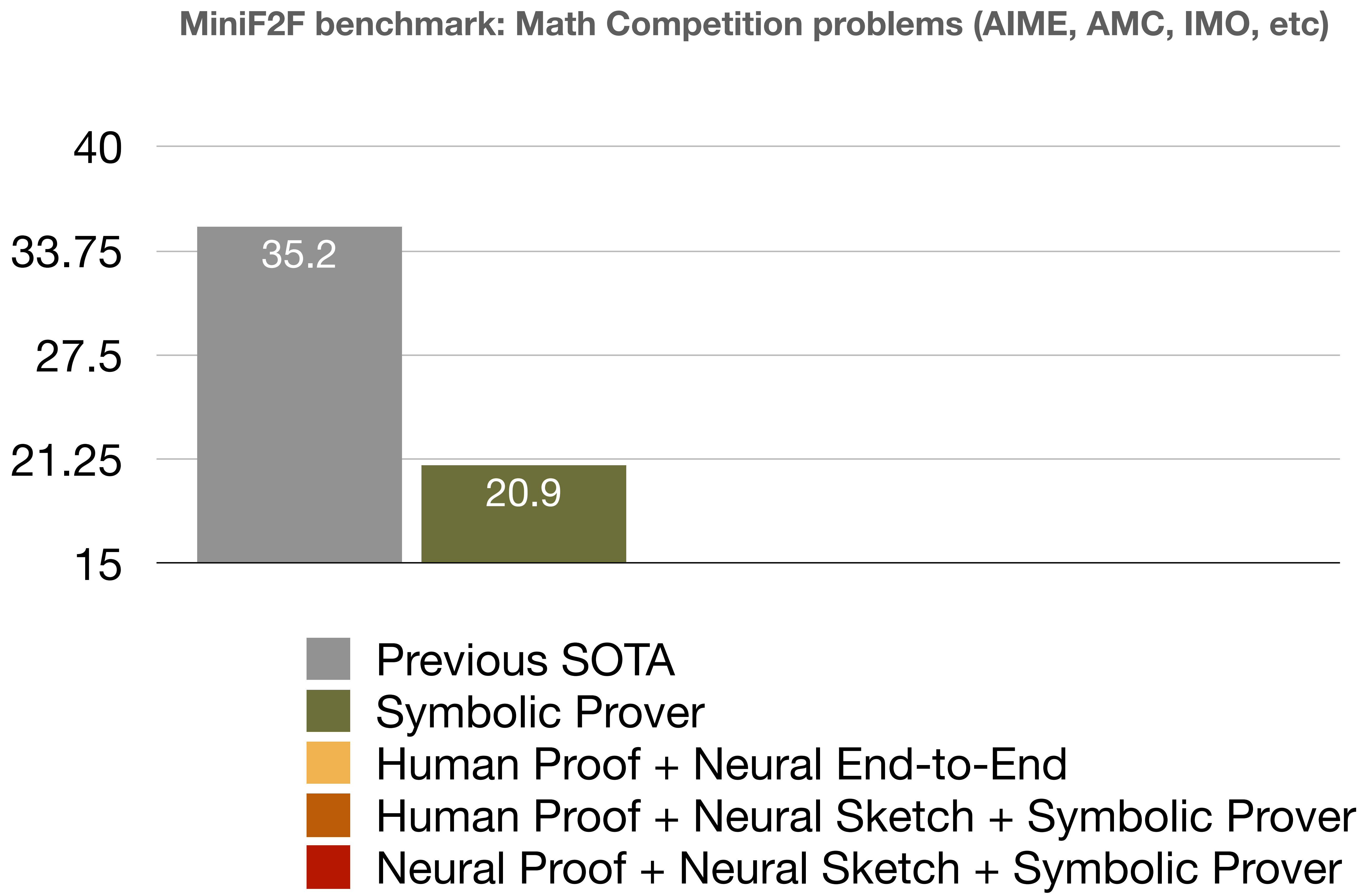
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



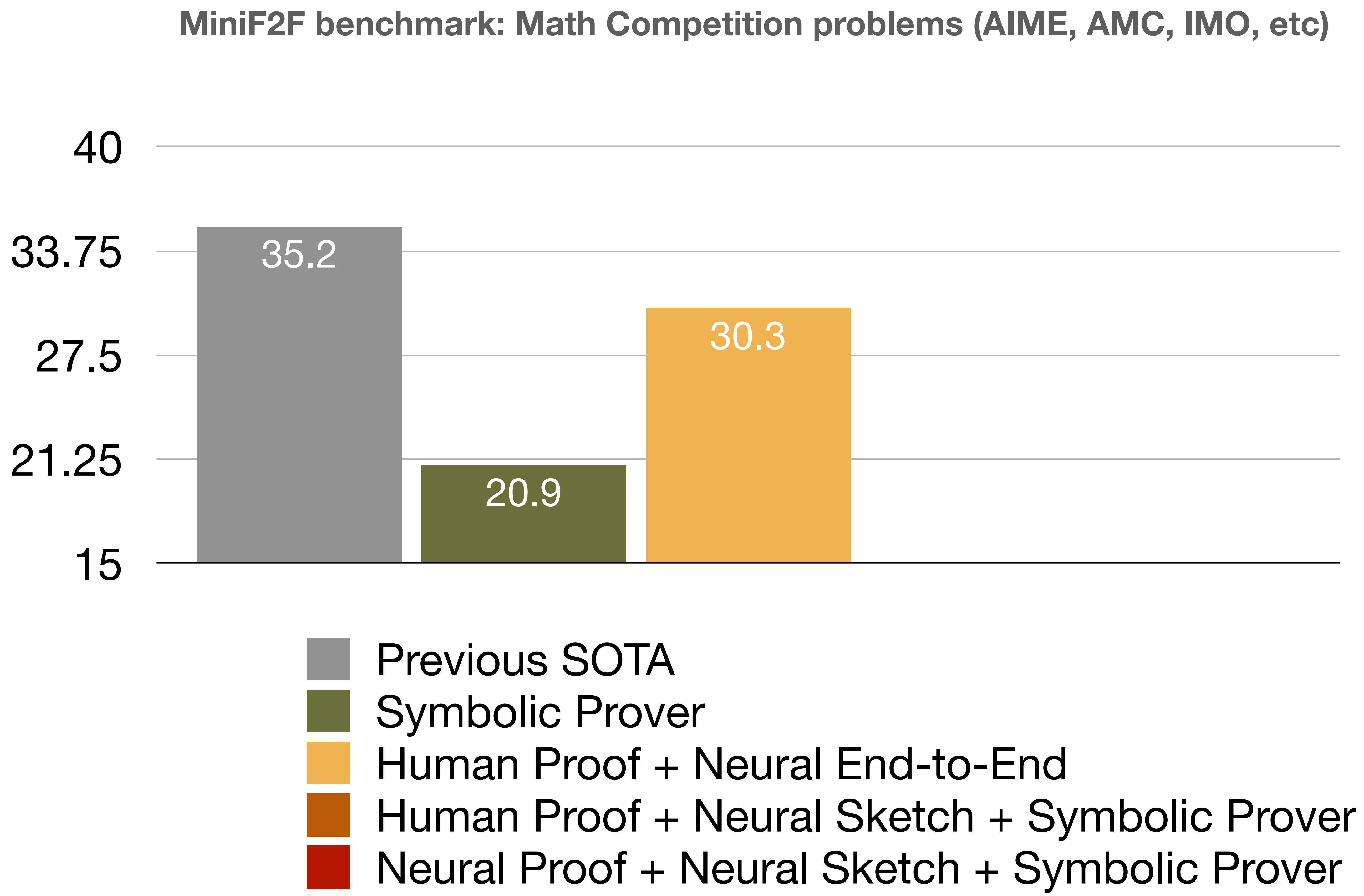
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



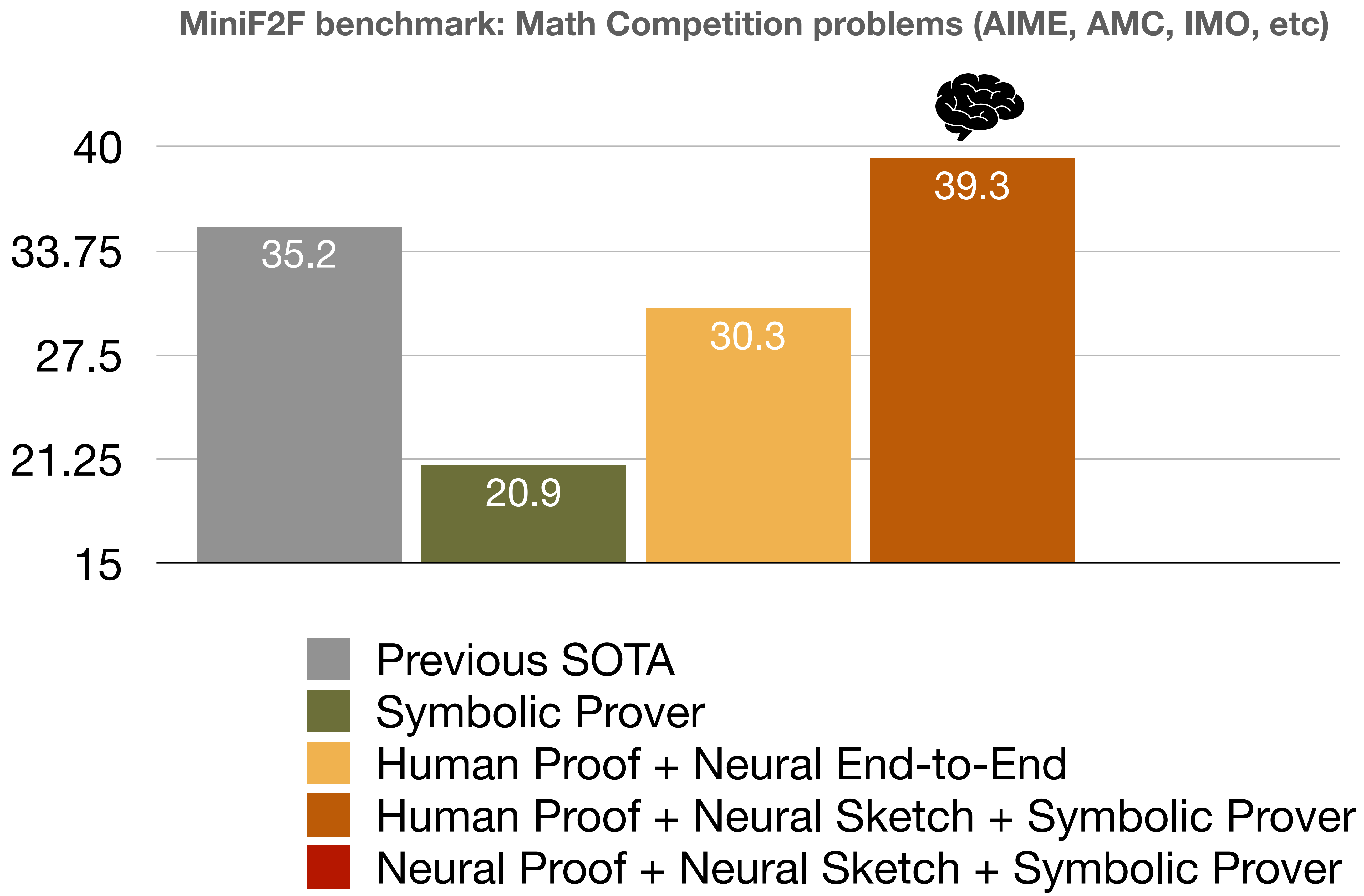
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



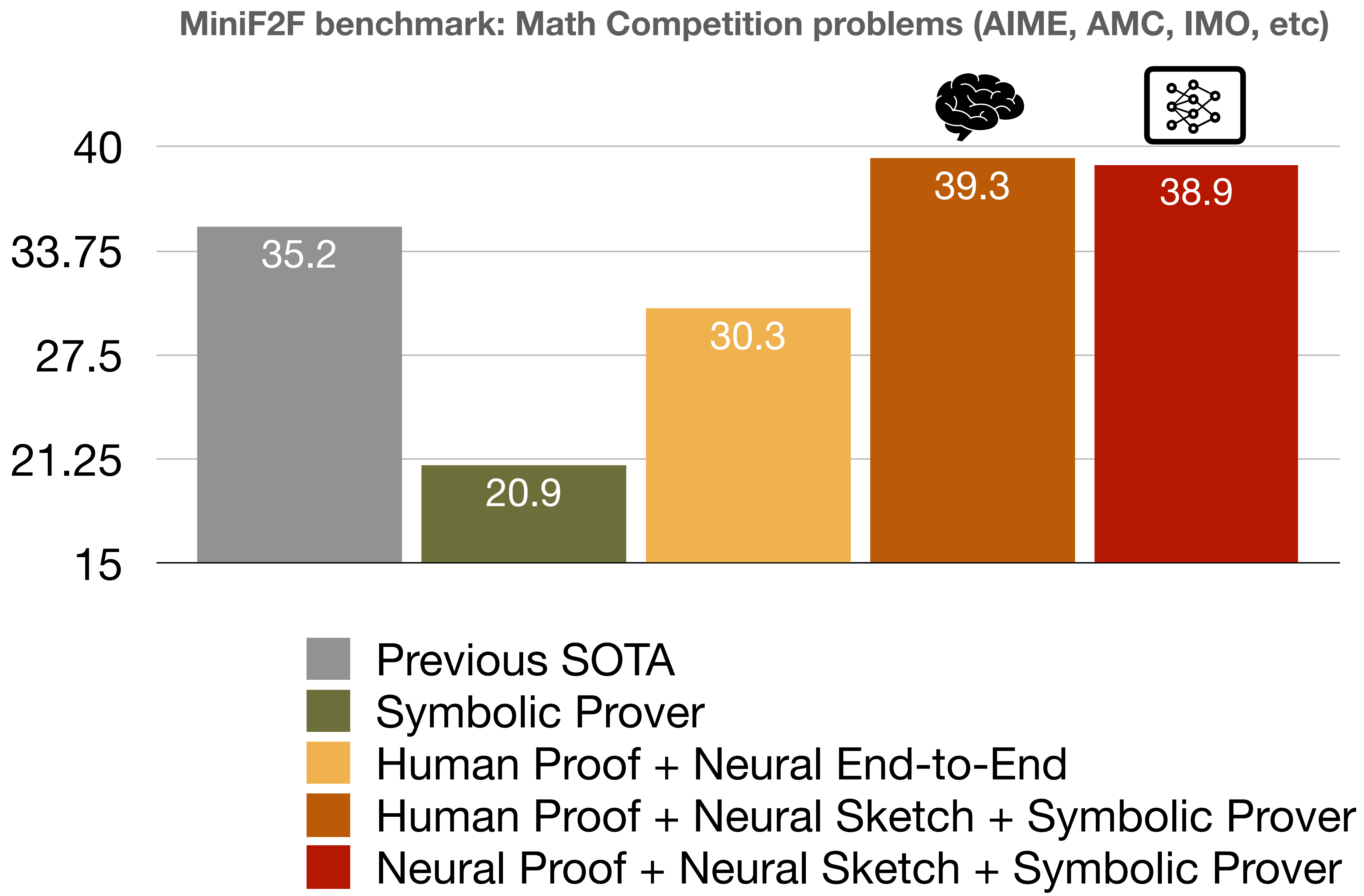
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



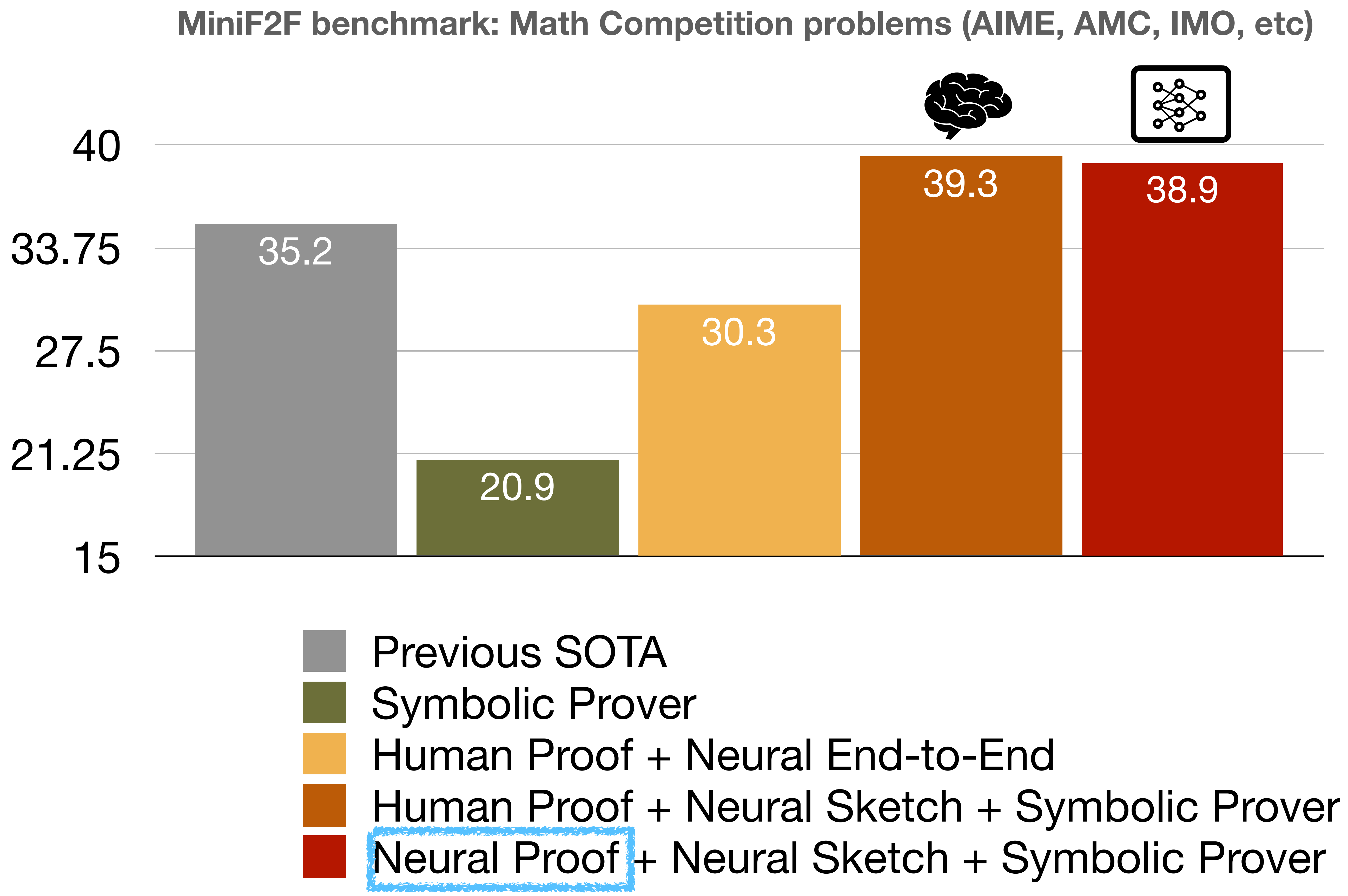
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



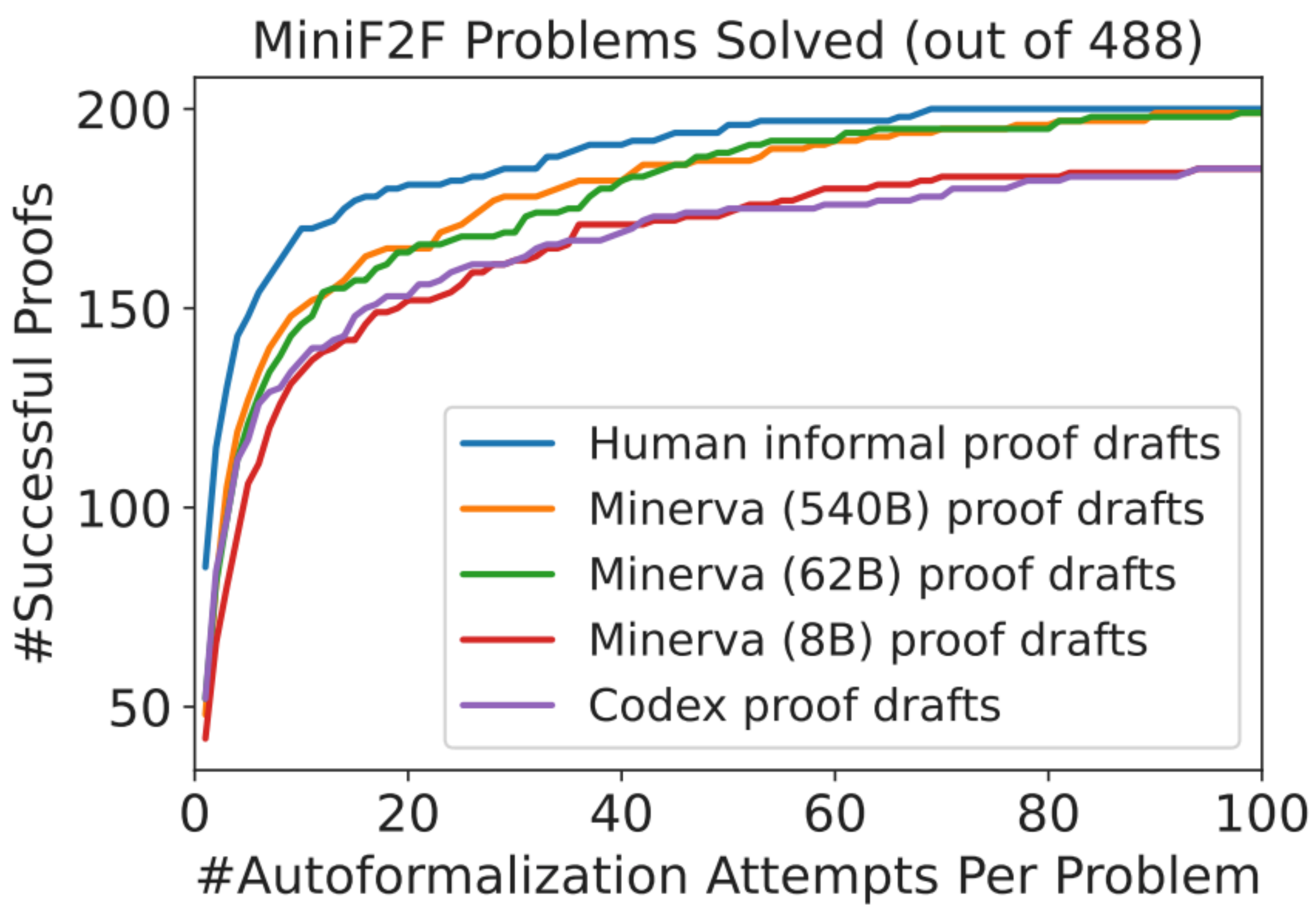
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



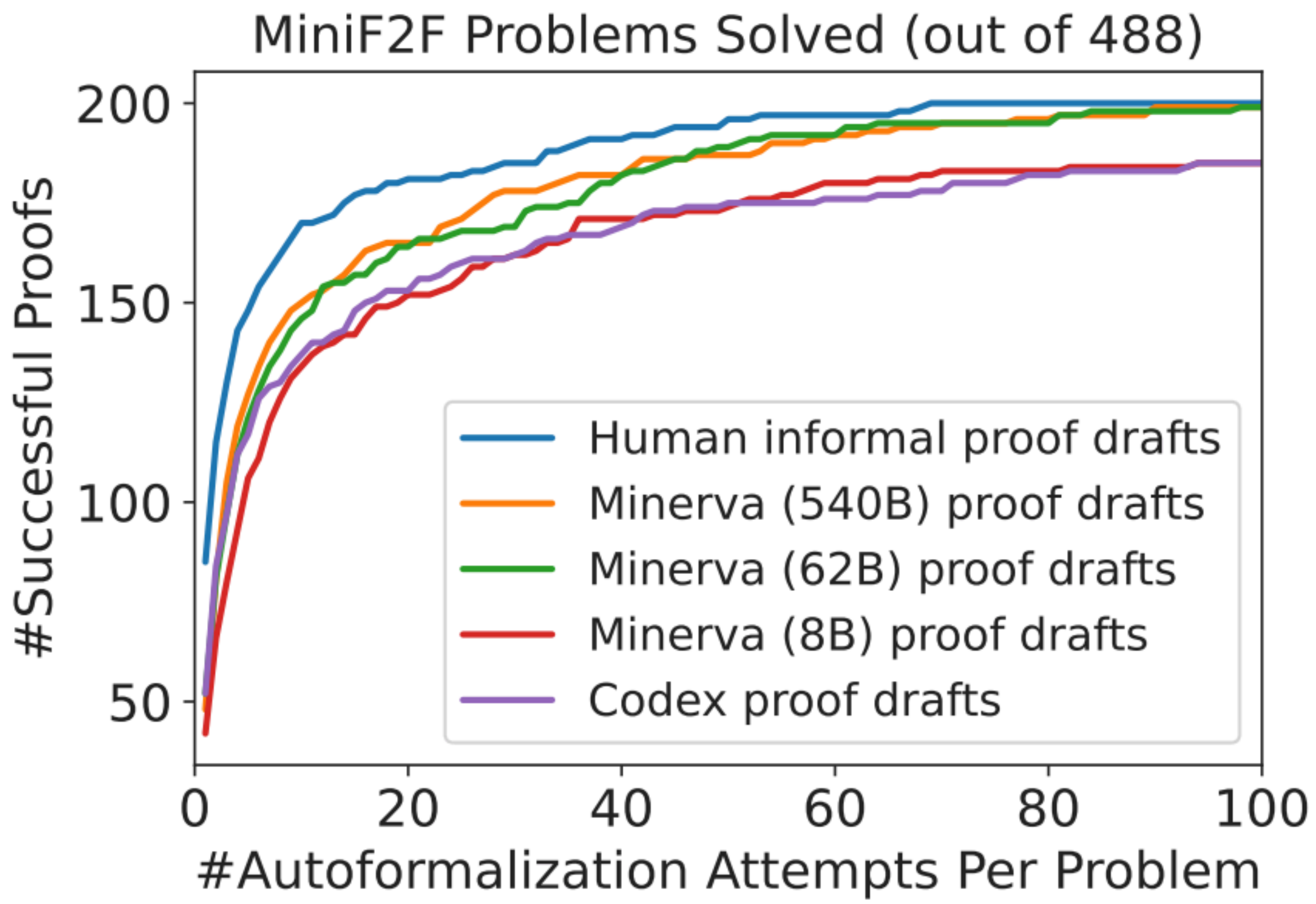
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

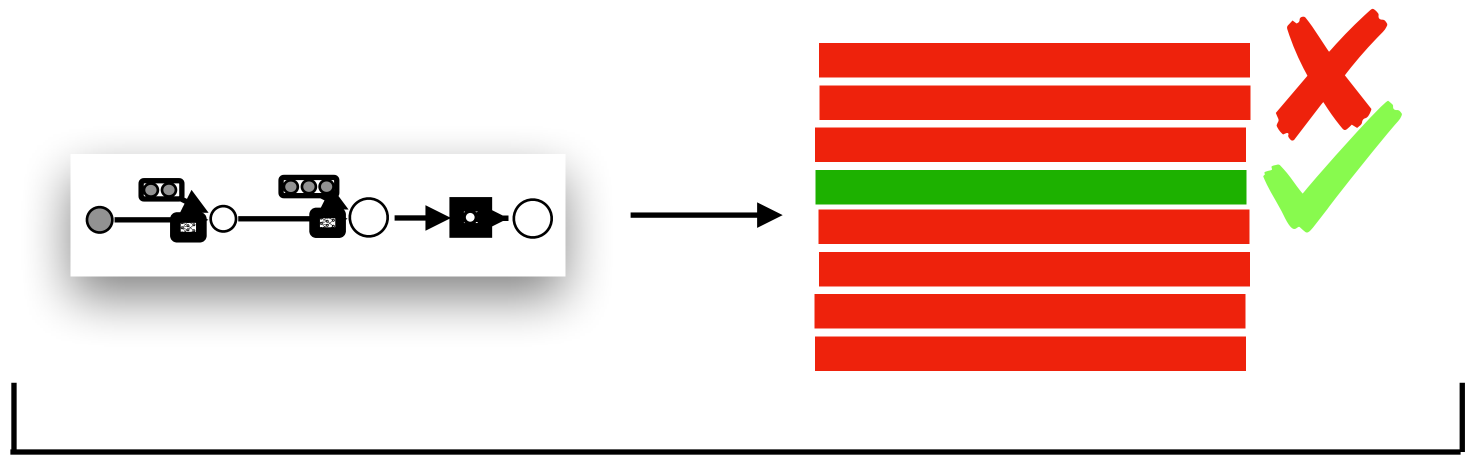
Modularity | sketching



Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | sketching



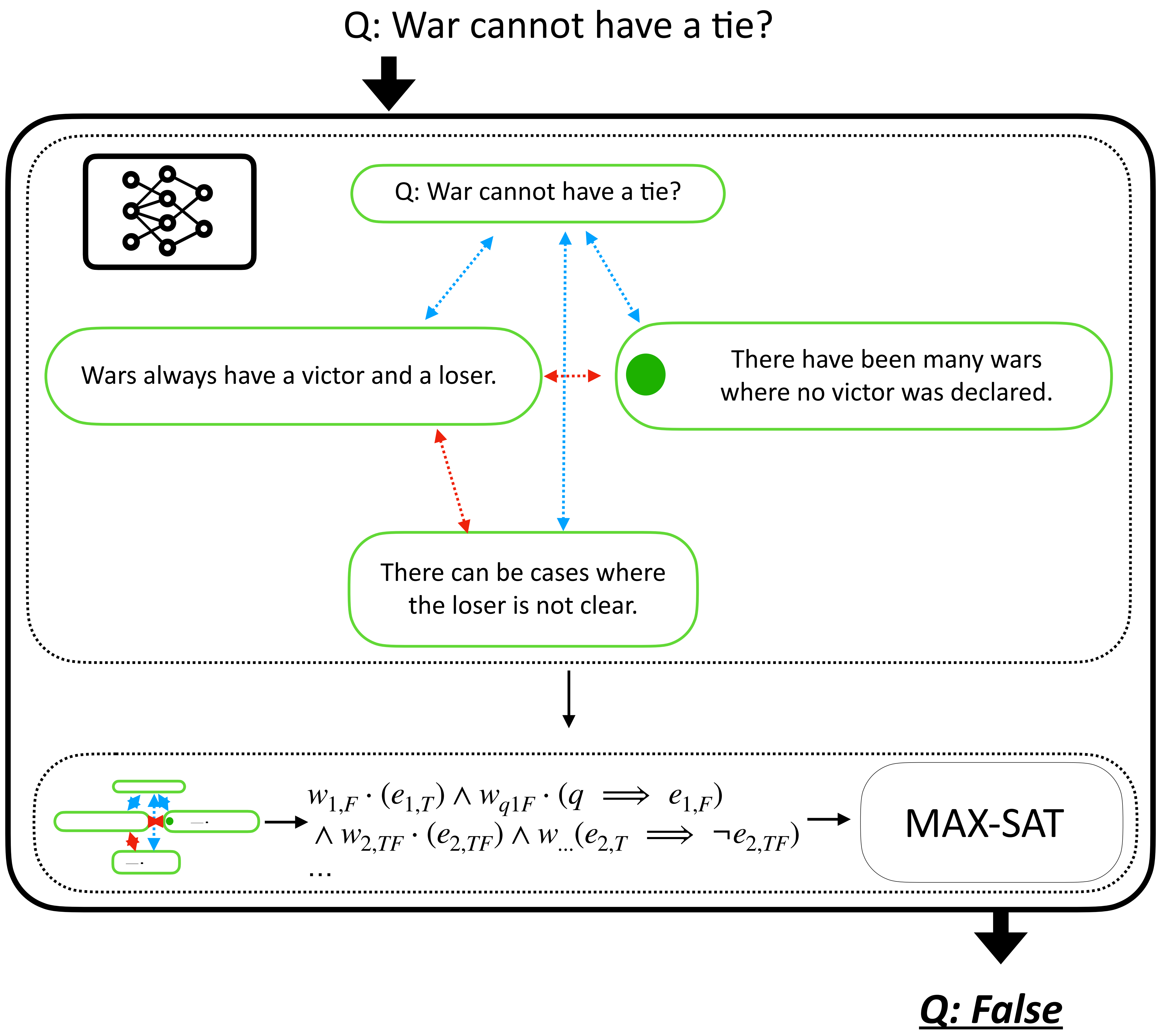


"Inference-time algorithm"

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, Y. Wu

Modularity | inference

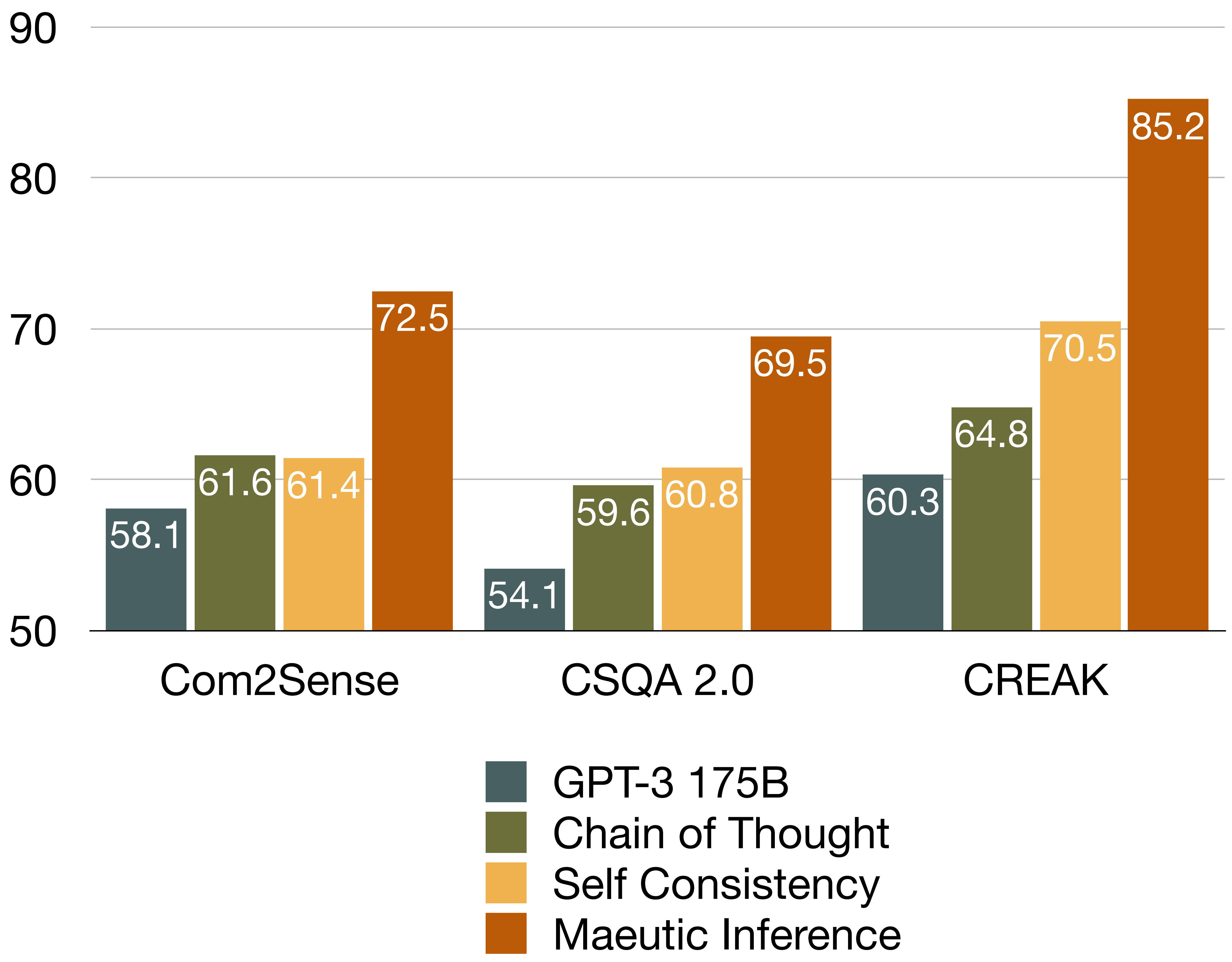
- Maeutic Inference [Jung et al 2022]:
  - Enumerate & score tree of rationales
  - Infer answer with MAX-Satisfiability
- **Modules & tools**: language model, scorer, verifier, MAX-SAT solver



Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations
J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. Le Bras, Y. Choi.
EMNLP 2022.

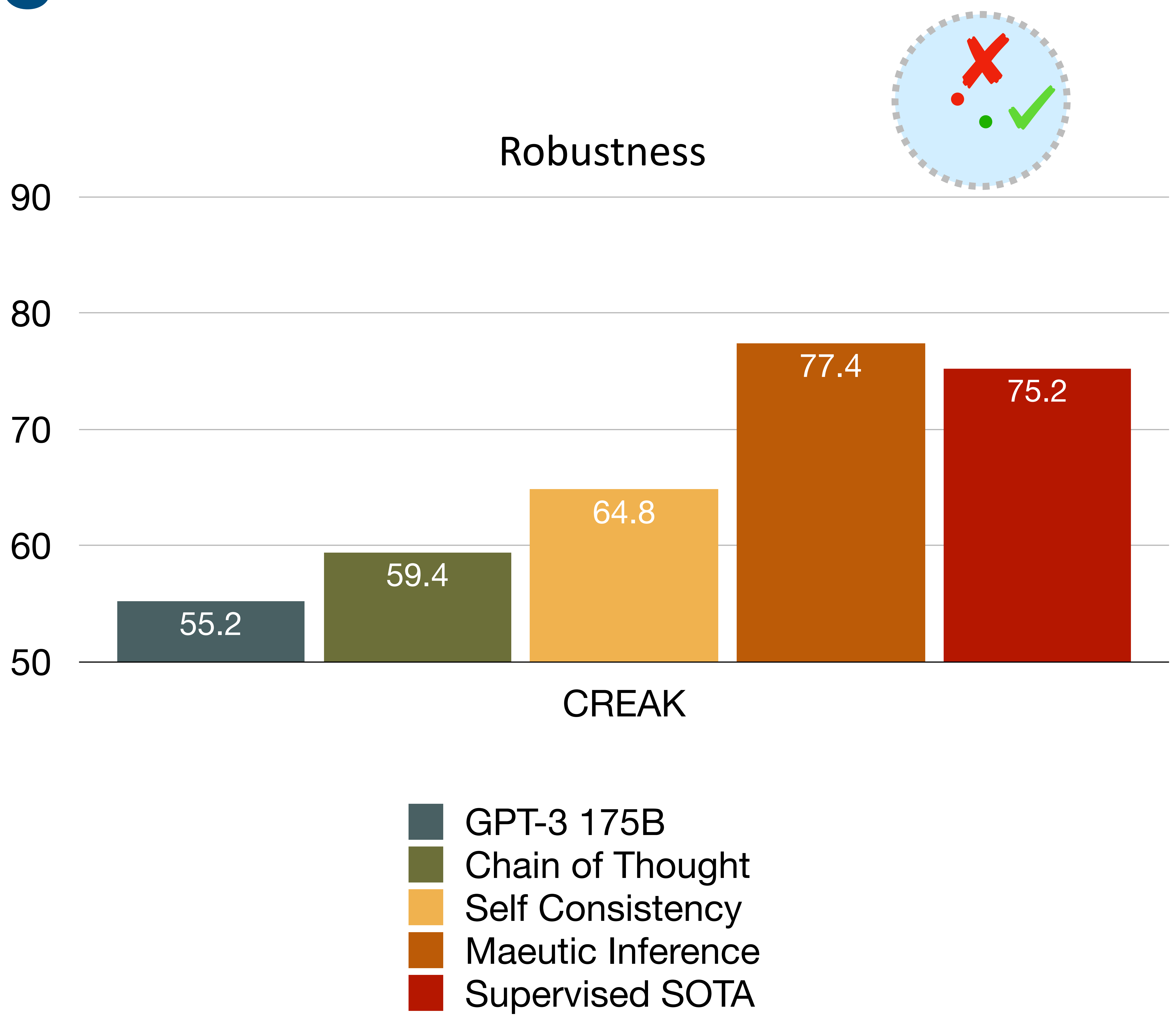
Modularity | inference

- Maeutic Inference [Jung et al 2022]:
  - Performance
    (commonsense QA & fact verification)
  - Robustness



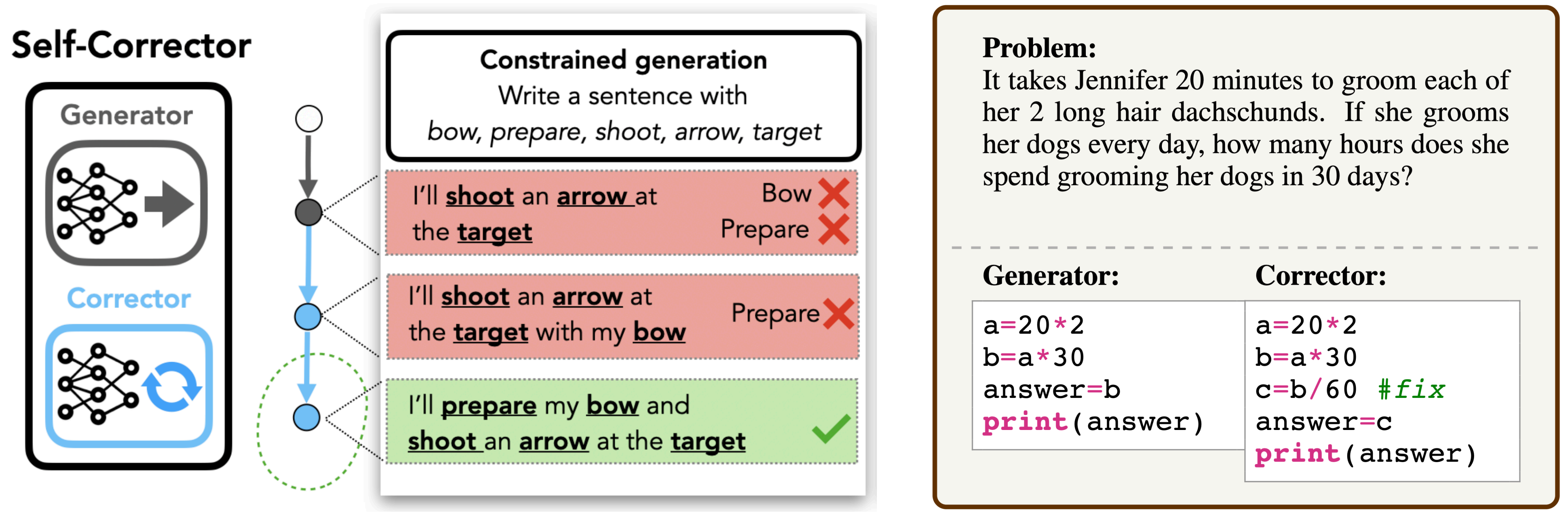
Modularity | inference

- Maeutic Inference [Jung et al 2022]:
  - Performance (commonsense QA & fact verification)
  - Robustness

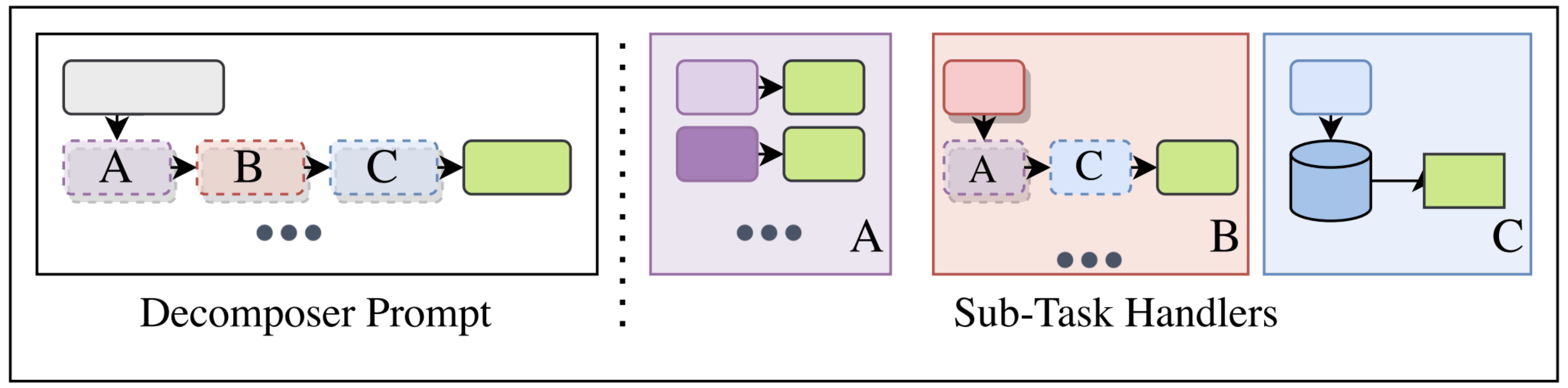


Modularity | other examples

- Recursion & correction
  - e.g. Self-correction [Welleck et al 2022]



- General decompositions
  - e.g. Decomposed prompting [Khot et al 2022]



- Text as "protocol" for multiple modalities
  - e.g. Socratic models [Zeng et al 2022]

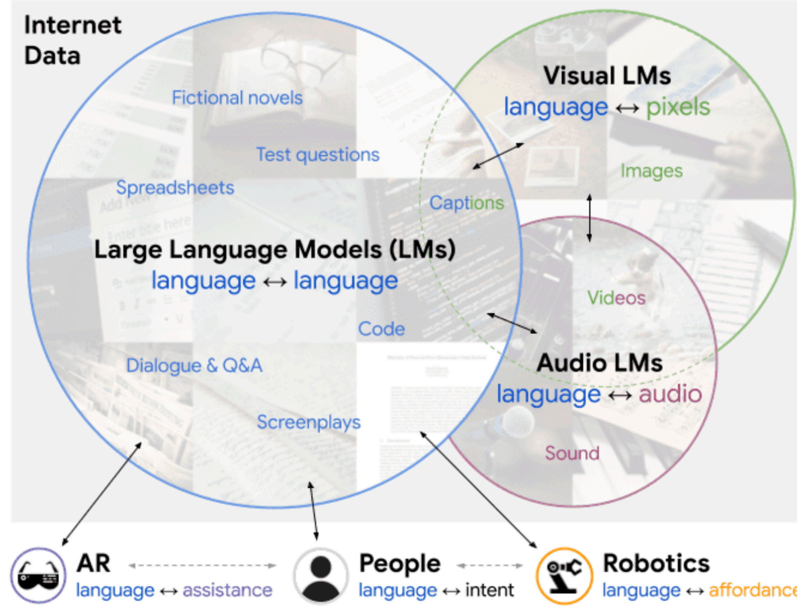
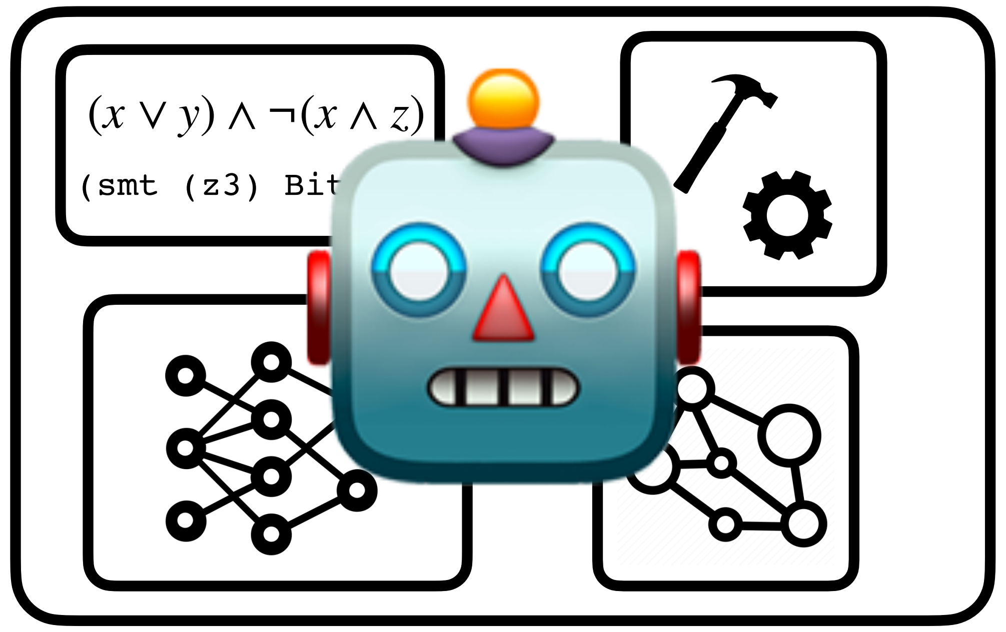


Figure 4: SMs with VLM, LM, and ALM can be prompted to generate a captions for key moments in videos, which can be assembled into a language-based world-state history (e.g., in the form of an event log) that the LM can answer free-form questions about.

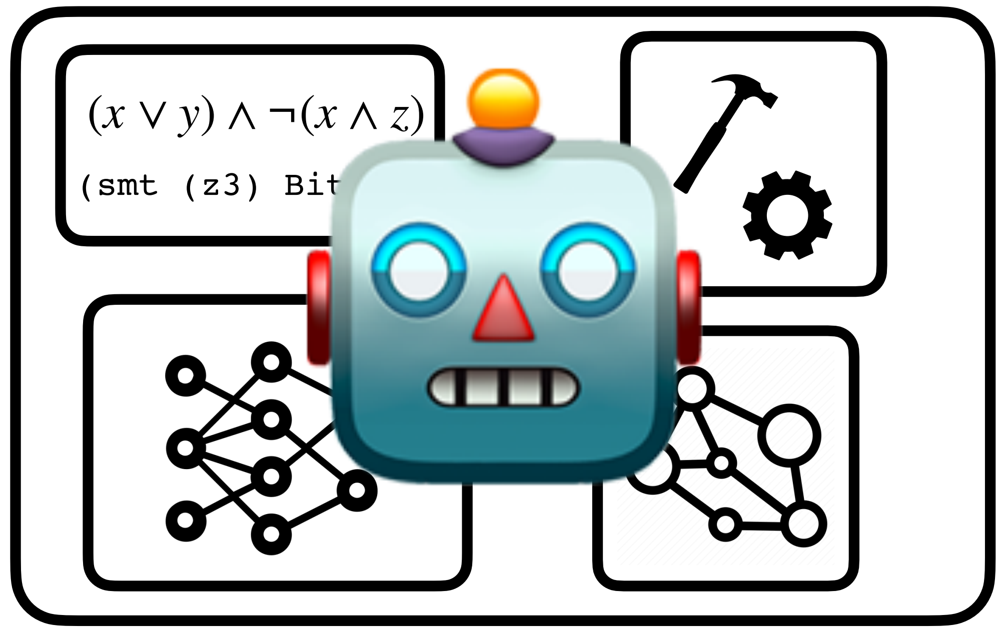
- … many more! An exciting & expanding area

Modularity | Takeaways



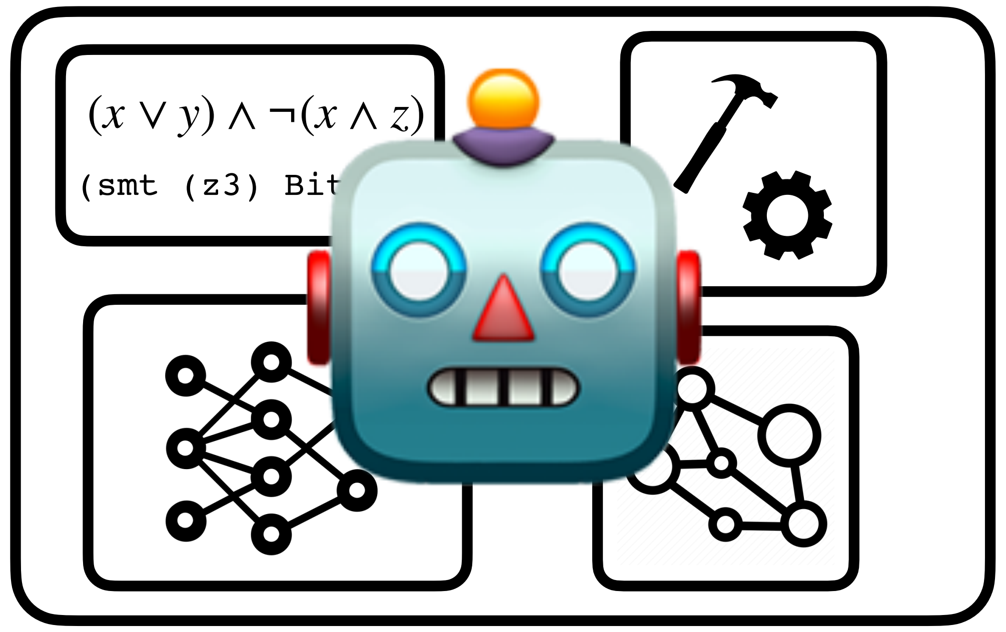
Modularity | Takeaways

- Multiple modules interacting through text
    - Formalism: graphical model / probabilistic program



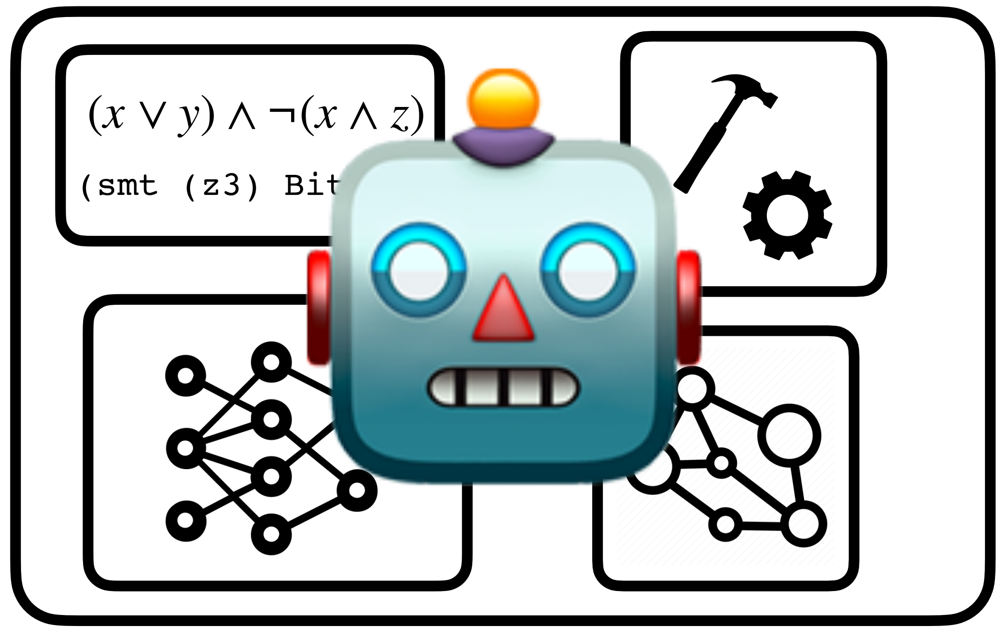
Modularity | Takeaways

- Multiple modules interacting through text
  - Formalism: graphical model / probabilistic program
- Intuition 1: Separation of concerns
  - High-level reasoning vs. low-level computation
  - Generation vs. retrieval & verification

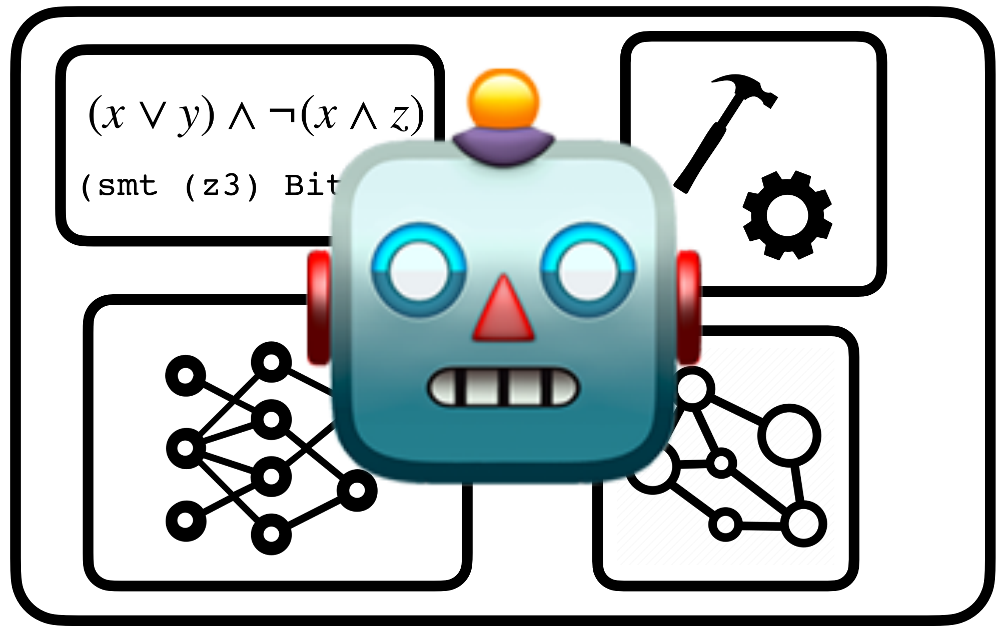


Modularity | Takeaways

- Multiple modules interacting through text
    - Formalism: graphical model / probabilistic program
- Intuition 1: Separation of concerns
    - High-level reasoning vs. low-level computation
    - Generation vs. retrieval & verification
- Intuition 2: Robust layer on top of a noisy enumerator
    - Neural: enumerate many solution candidates
    - Symbolic: verify, fill in gaps, resolve globally



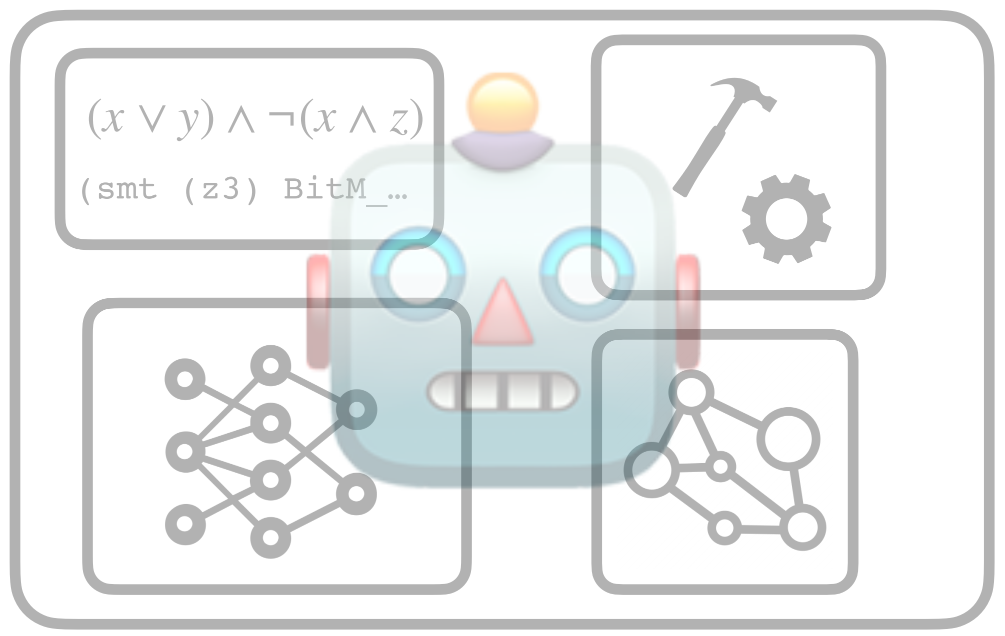
Modularity | Takeaways



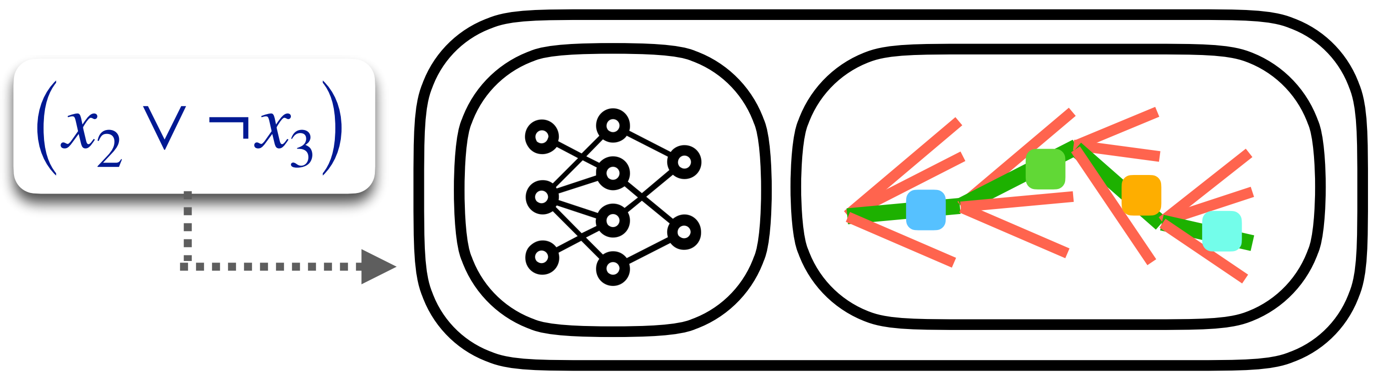
- Multiple modules interacting through text
  - Formalism: graphical model / probabilistic program
- Intuition 1: Separation of concerns
  - High-level reasoning vs. low-level computation
  - Generation vs. retrieval & verification
- Intuition 2: Robust layer on top of a noisy enumerator
  - Neural: enumerate many solution candidates
  - Symbolic: verify, fill in gaps, resolve globally
- Many more ideas to explore here!

Overview

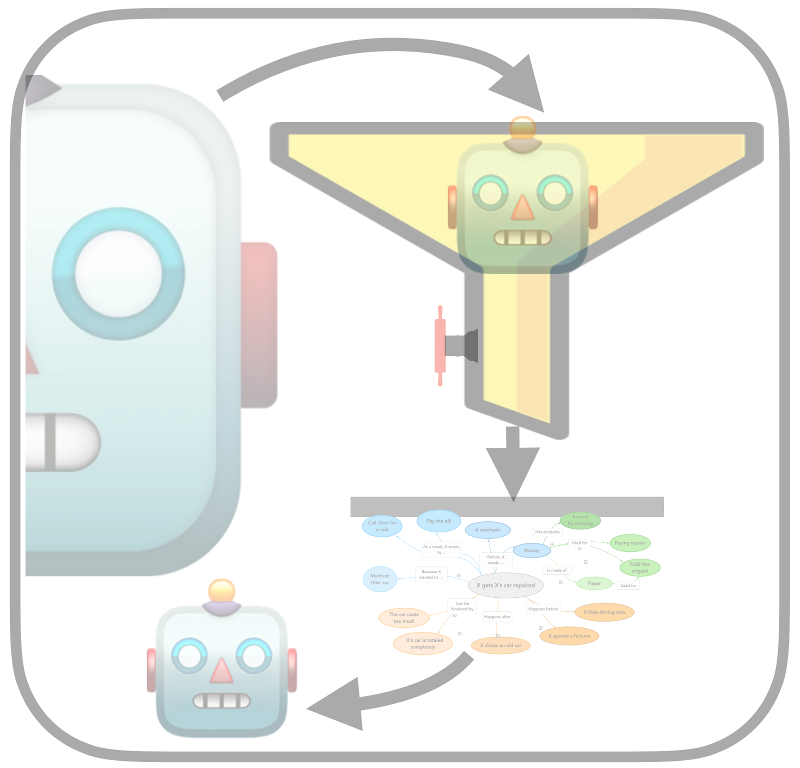
- **Modularity**
  - Single monolithic system → decomposed neural & symbolic modules



- **Constraints**
  - Discrete logical constraints

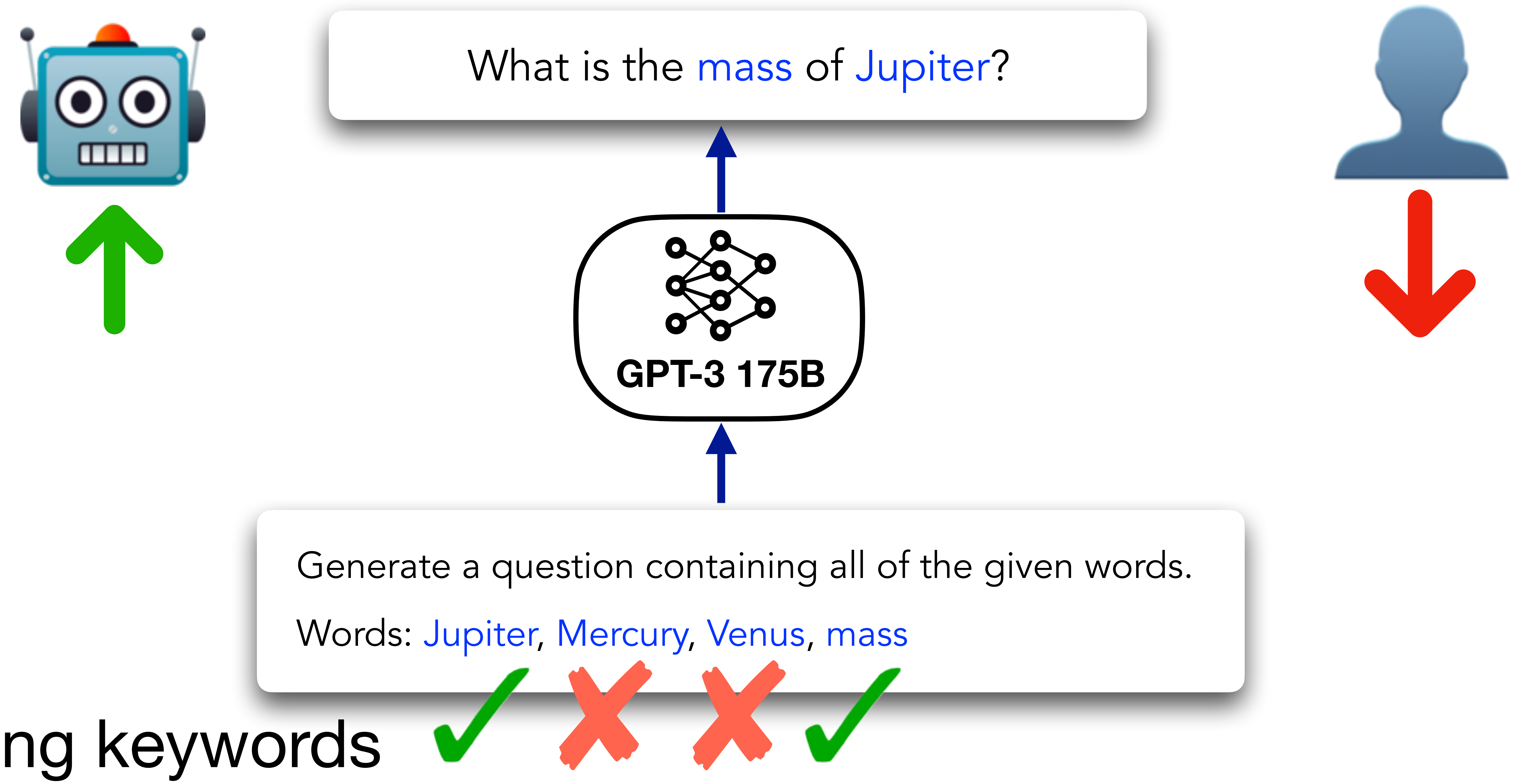


- **Knowledge**
  - Hand-crafted → *generated* and *distilled*



Constraints

- Language models are difficult to *control*



missing keywords

Example from: https://beta.openai.com/playground

Constraints

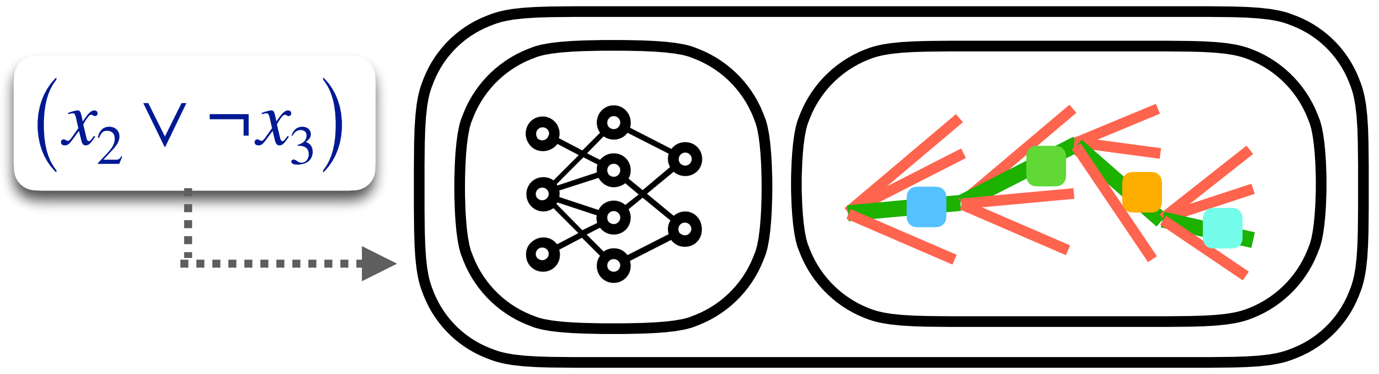
- Language models are difficult to *control*

Constraints

- Language models are difficult to *control*
- Build **decoding algorithm** to enforce constraints.

Constraints

- Language models are difficult to *control*
- Build **decoding algorithm** to enforce constraints.
  - **Lexical constraints**: words should or should not appear in the generation.



Constraints

- Language models are difficult to *control*
- Build **decoding algorithm** to enforce constraints.
  - **Lexical constraints**: words should or should not appear in the generation.

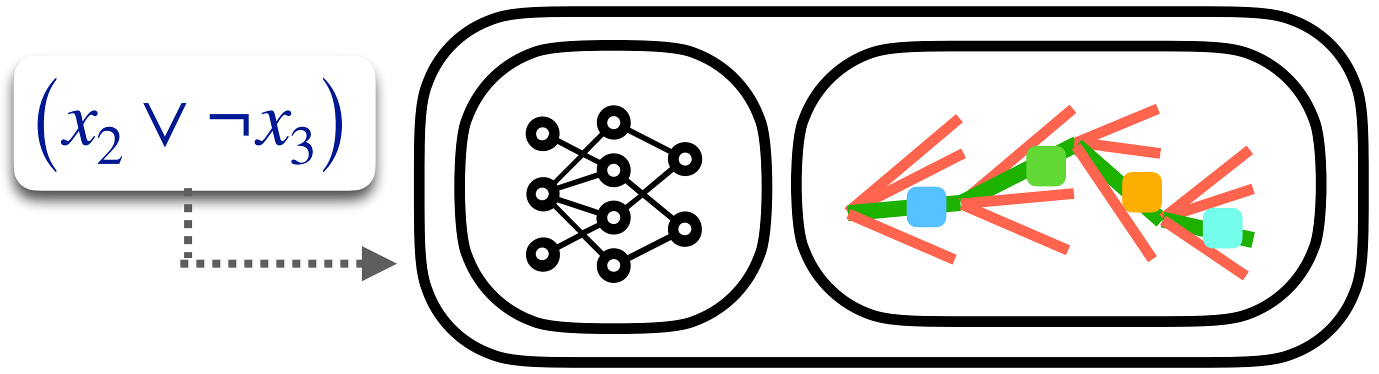


Table to Text

X

type	hotel
count	182
dogs allowed	don't care

Y There are 182 hotels if you do not care whether dogs are allowed .

Theorem Proving

X **Theorem:** Let x be an even integer. Then $x + 5$ is odd.

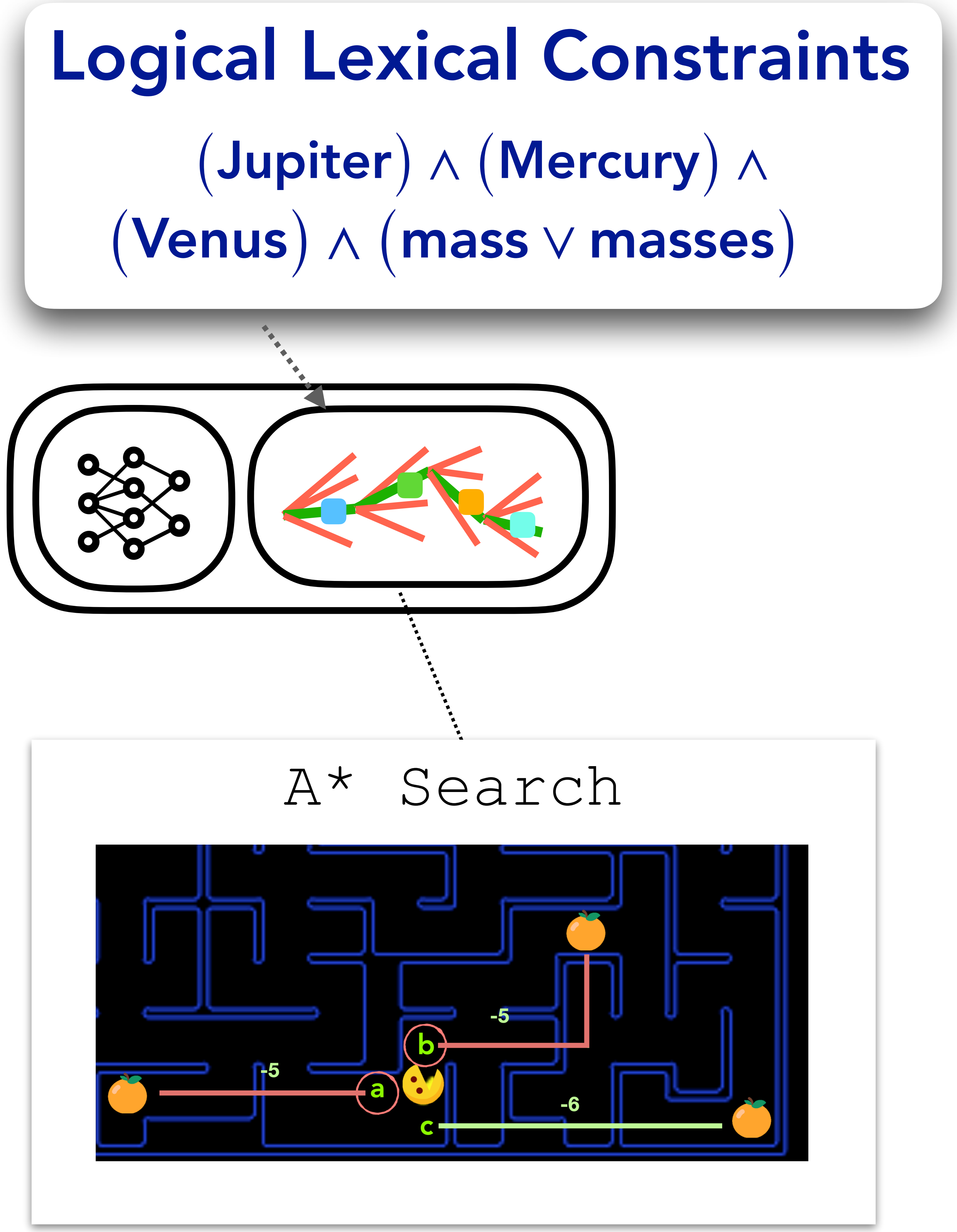
Y **Proof:** Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even. Then there exists an integer k such that $x + 5 = 2k$.

…

[Welleck et al 2022]

Constraints

- **NeuroLogic A*-Esque Decoding**
  [Lu et al 2022]
  - Lexical constraints expressed in Conjuctive Normal Form
  - A*-search-like lookahead



[NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics]
X. Lu, S. Welleck, P. West, L. Jiang, J. Kasai, D. Khashabi, R. Le Bras, L. Qin, Y. Yu, R. Zellers, N. Smith, Y. Choi
NAACL 2022

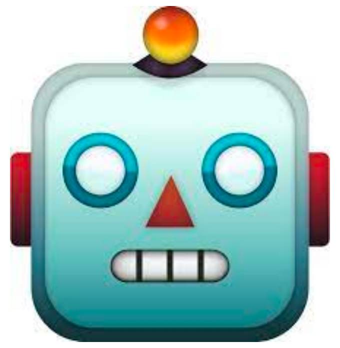
Constraints

NeuroLogic A*-Esque Decoding [Lu et al 2022]

Write a sentence with: car $\wedge$ drive $\wedge$ snow

score $\mathbf{s} = \log P_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t})$

Off-the-Shelf GPT2



Beam Search

[NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics]

X. Lu, S. Welleck, P. West, L. Jiang, J. Kasai, D. Khashabi, R. Le Bras, L. Qin, Y. Yu, R. Zellers, N. Smith, Y. Choi

NAACL 2022

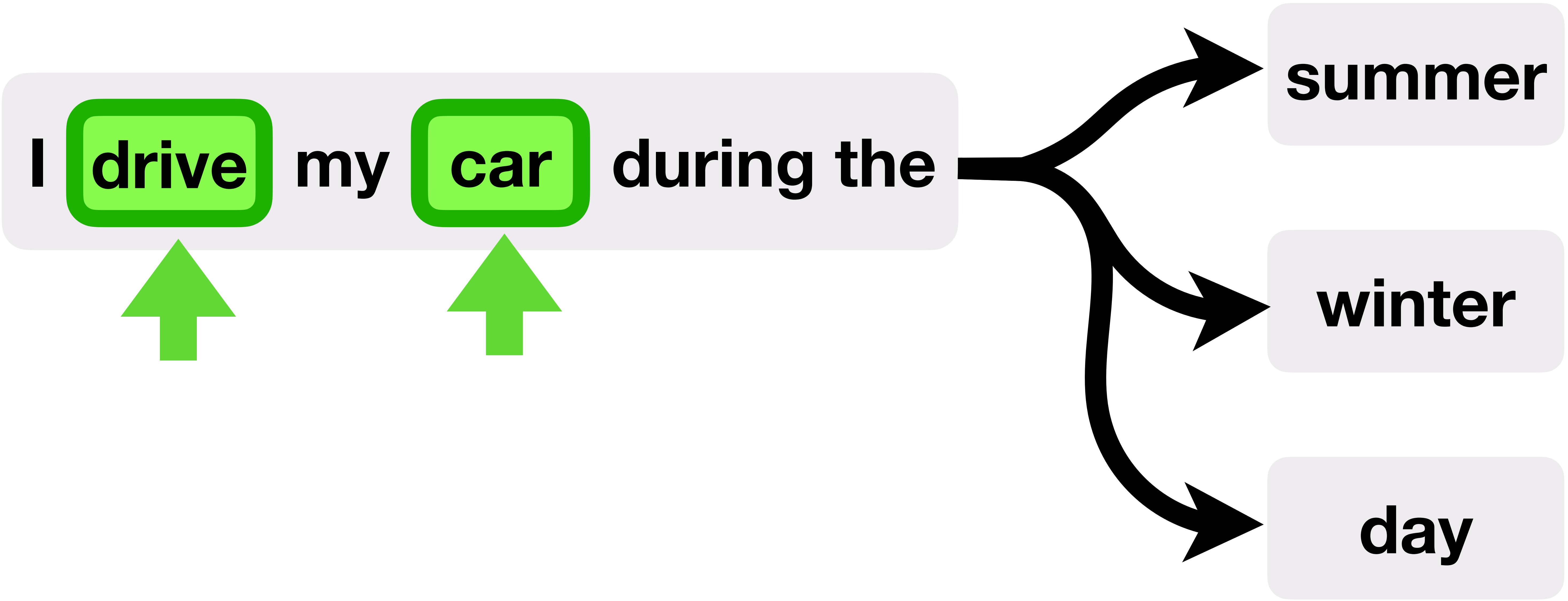
Constraints

NeuroLogic A*-Esque Decoding [Lu et al 2022]

Write a sentence with: car $\wedge$ drive $\wedge$ snow

score $\mathbf{s} = \log P_\theta(\mathbf{y}_t | \mathbf{y}_{<t}) + \alpha' \sum_{i=1}^{m} C_i + \lambda_1 \cdot \max_{\{D_i:\, D_i=0\}} \log P_\theta(D_i | \mathbf{y}_{<t+k})$

Constraints



NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics
X. Lu, S. Welleck, P. West, L. Jiang, J. Kasai, D. Khashabi, R. Le Bras, L. Qin, Y. Yu, R. Zellers, N. Smith, Y. Choi
NAACL 2022

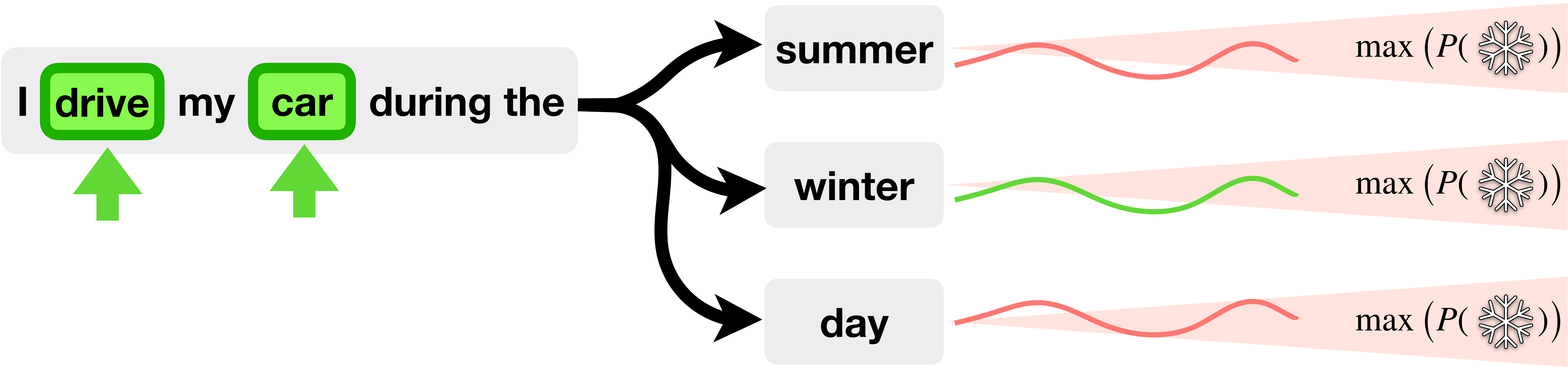
Constraints

NeuroLogic A*-Esque Decoding [Lu et al 2022]

Write a sentence with: car $\wedge$ drive $\wedge$ snow

score $\quad s = \log P_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}) + \alpha' \sum_{i=1}^{m} C_i + \lambda_1 \cdot \max_{\{D_i:\, D_i = 0\}} \log P_\theta(D_i \mid \mathbf{y}_{<t+k})$

Constraints **A* Heuristic**



[NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics](#)
X. Lu, S. Welleck, P. West, L. Jiang, J. Kasai, D. Khashabi, R. Le Bras, L. Qin, Y. Yu, R. Zellers, N. Smith, Y. Choi
NAACL 2022

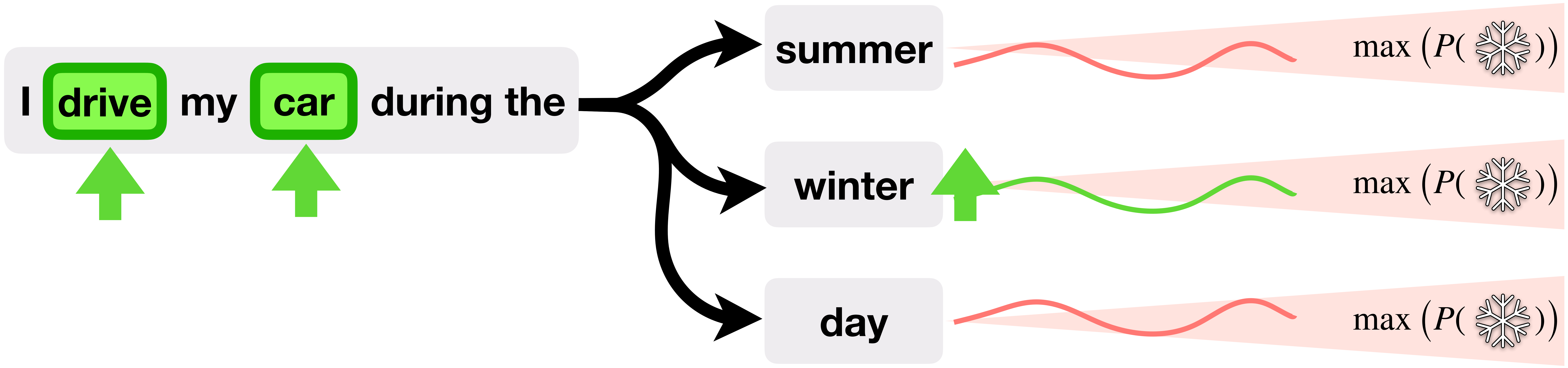
Constraints

NeuroLogic A*-Esque Decoding [Lu et al 2022]

Write a sentence with: car $\wedge$ drive $\wedge$ snow

score $\mathbf{s} = \log P_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}) + \alpha' \sum_{i=1}^{m} C_i + \lambda_1 \cdot \max_{\{D_i:\, D_i=0\}} \log P_\theta(D_i \mid \mathbf{y}_{<t+k})$

Constraints **A* Heuristic**



[NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics]

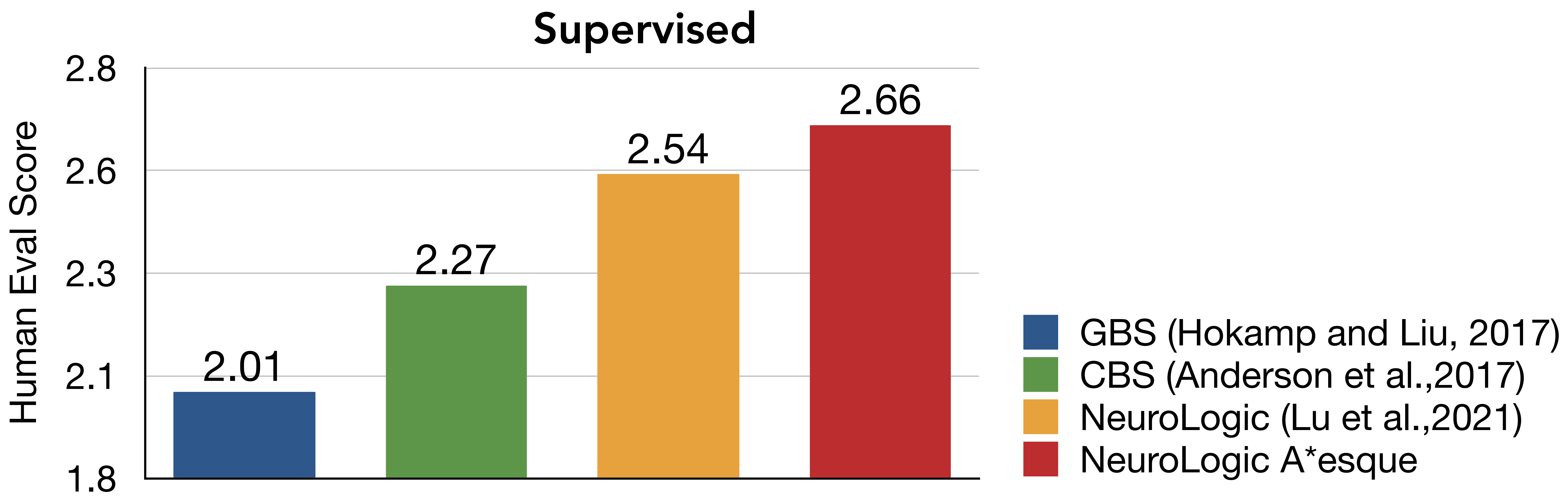
X. Lu, S. Welleck, P. West, L. Jiang, J. Kasai, D. Khashabi, R. Le Bras, L. Qin, Y. Yu, R. Zellers, N. Smith, Y. Choi

NAACL 2022

Human Evaluation Results

CommonGen

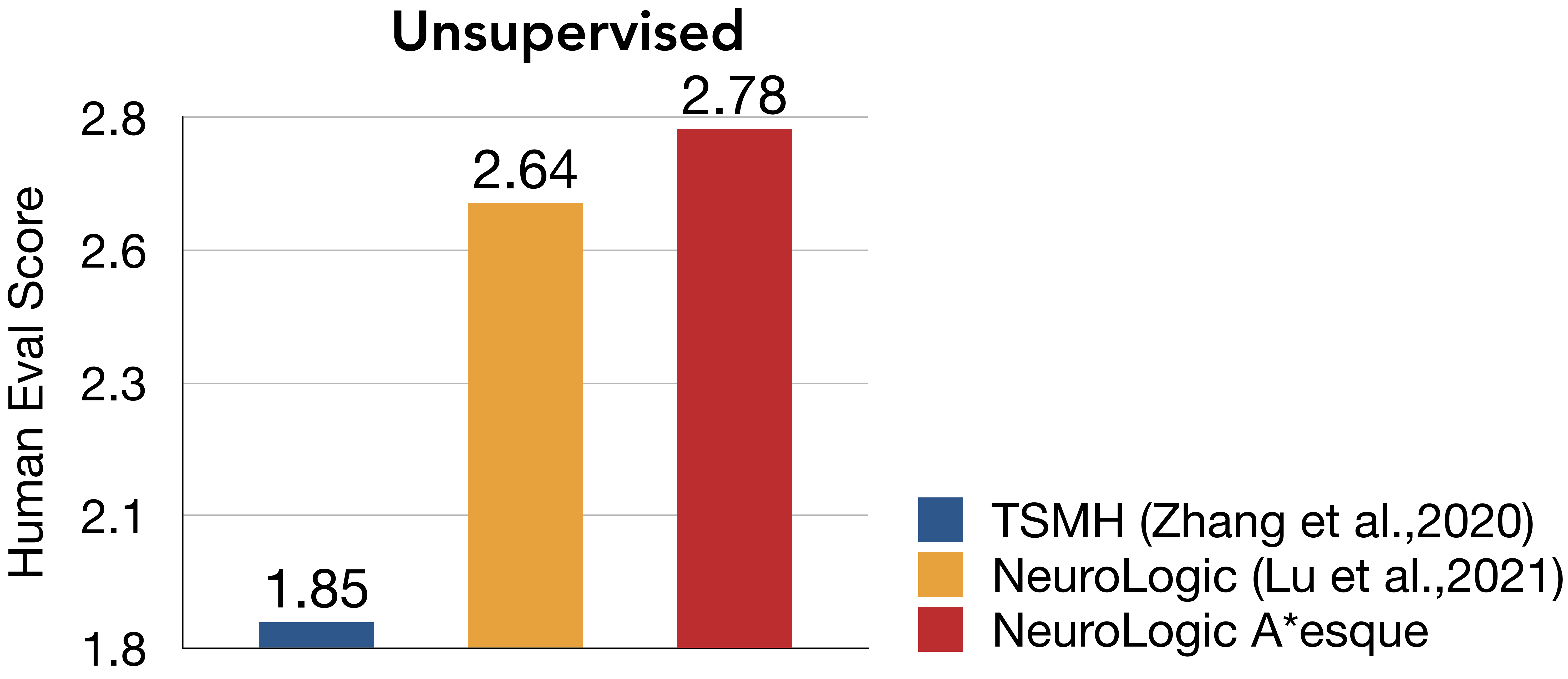
(Lin et al., 2020)



Human Evaluation Results

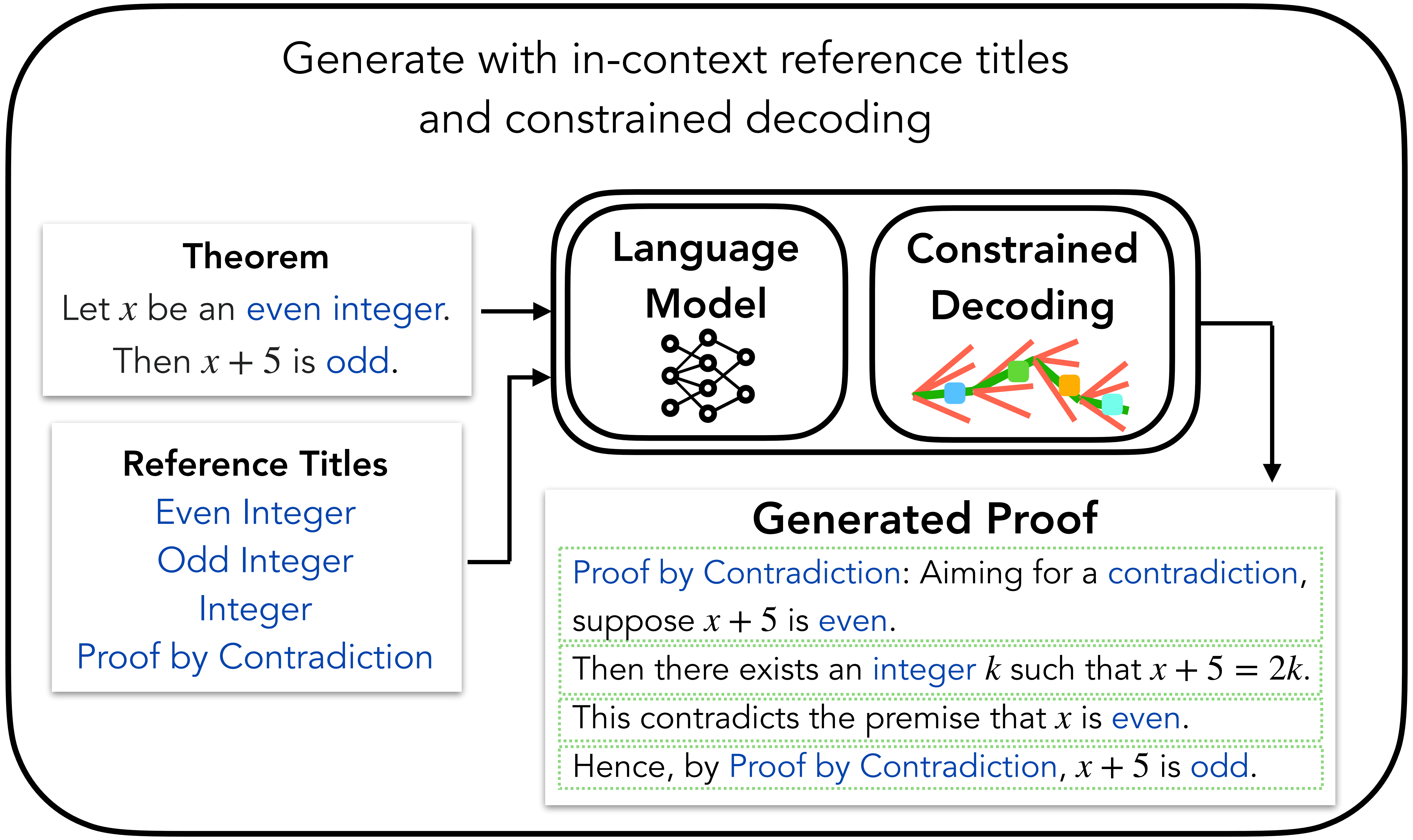
CommonGen

(Lin et al., 2020)



Constraints

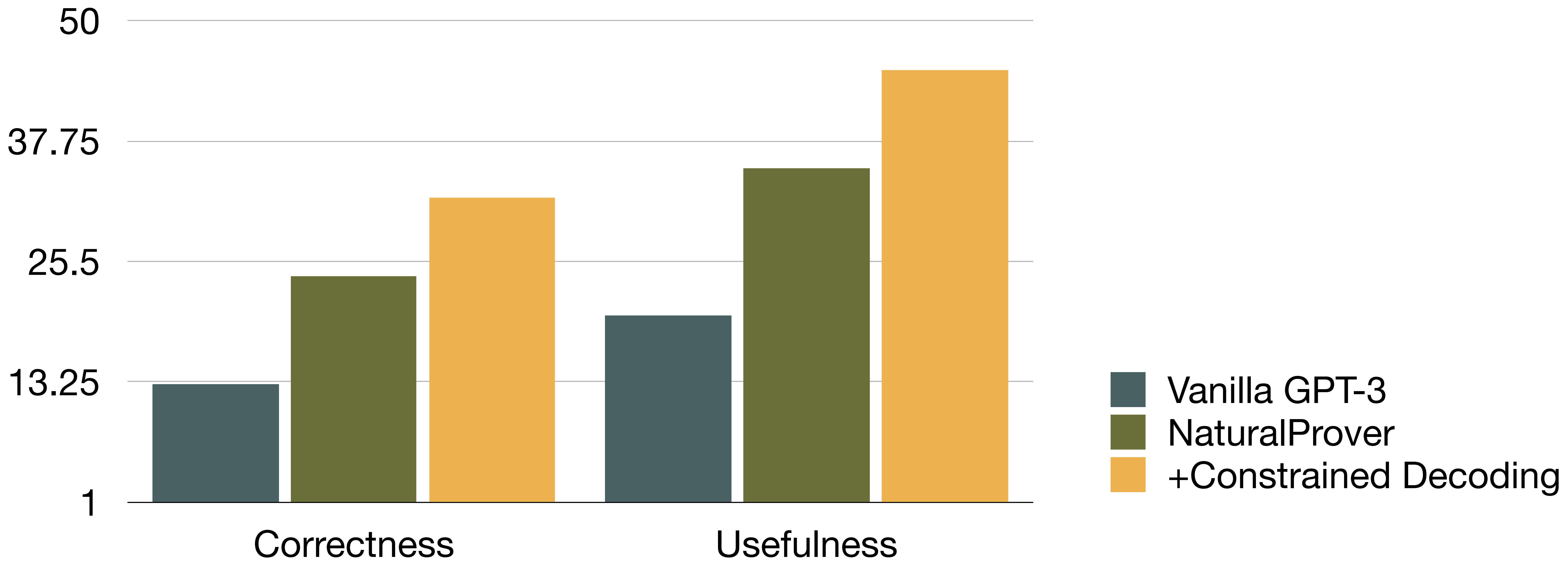
- **Stepwise Stochastic Beam Search** [Welleck et al 2022]
    - Beam-search over arbitrary-length segments with a constraint value function.



NaturalProver: Grounded Mathematical Proof Generation with Language Models
S. Welleck, J. Liu, X. Lu, H. Hajishirzi, Y. Choi
NeurIPS 2022.

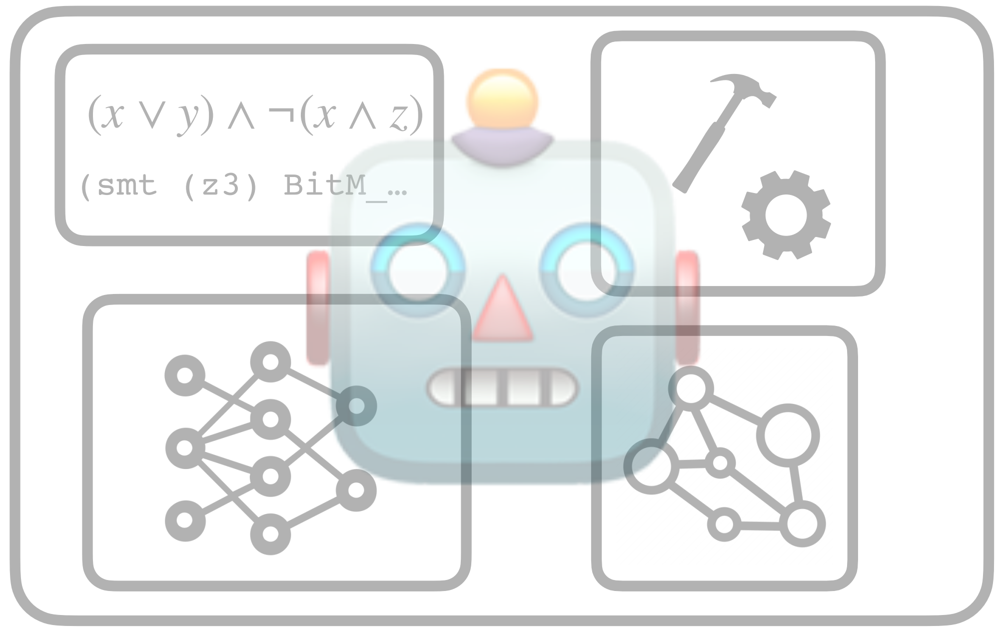
Constraints

- **Stepwise Stochastic Beam Search**
  [Welleck et al 2022]
    - Human evaluation (UW Mathematics students)

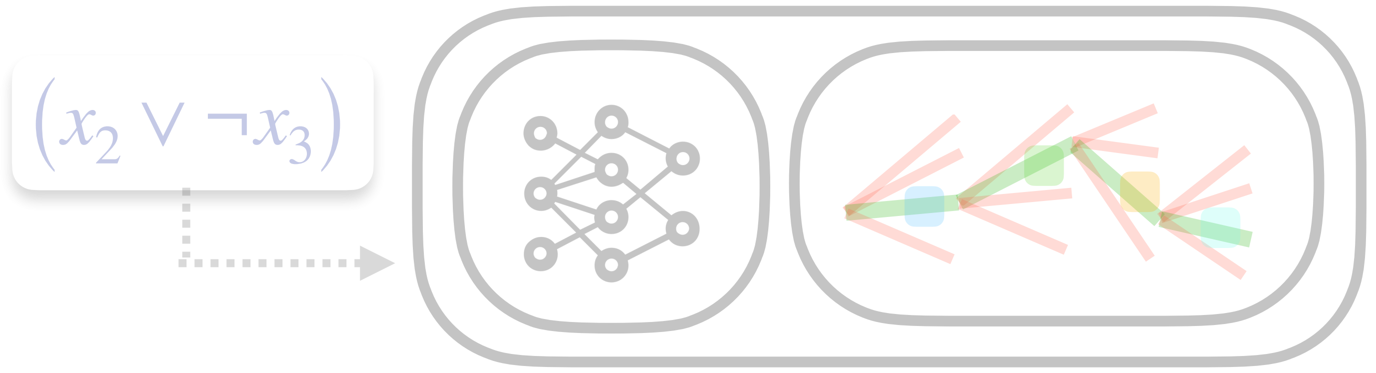


Overview

- **Modularity**
  - Single monolithic system → decomposed neural & symbolic modules



- **Constraints**
  - Discrete logical constraints



- **Knowledge**
  - Hand-crafted → *generated* and *distilled*