

# Prover-Verifier Games Improve Legibility of LLM Outputs

Jan Hendrik Kirchner†\* **Yining Chen**\* Harri Edwards† Jan  
Leike† Nat McAleese Yuri Burda †

\* Equal contribution †Work done while at OpenAI

# Overview

# The legibility problem

can be fully understood to the extent that any possible errors would be spotted

Write me a fast sorting algorithm

```
for (k = 2; k <= n; k *= 2)
  for (j = k/2; j > 0; j /= 2)
    for (i = 0; i < n; i++)
      l = bitwiseXOR (i, j);
      if (l > i)
        if ( (bitwiseAND (i, k) == 0) AND (arr[i] > arr[l])
            OR (bitwiseAND (i, k) != 0) AND (arr[i] < arr[l]) )
          swap the elements arr[i] and arr[l]
```

???



Human



Superhuman AI

# How to make LLM outputs more legible?

- Reinforcement Learning from Human Feedback (RLHF) [Christiano et al. 2017]
  - Human labels
- This work: Optimize LLM chain-of-thoughts (CoTs) to be more verifiable by smaller LLMs  $\Rightarrow$  more legible by humans.
  - CoTs [Wei et al. 2022]: natural justifications for final answers
  - Small models as proxies for humans
    - Scalable oversight
    - Continuously parameterize capability gap

# Verifiability (verifier $V$ , prover model family $F$ )

- **Completeness**
  - Exists helpful prover  $H$  such that for any input in domain, it generates a correct solution AND verifier recognizes it as correct.
  - Empirical:  $H$ : high pass rates; verifier: high recall
- **Soundness**
  - For any input in domain, for any sneaky prover in  $F$ ,  $V$  rejects incorrect solutions from  $S$ .
  - Empirical:  $V$ : high precision on (compute-limited) adversarial attacks

Inspired by Probably-Approximately Correct (PAC) verification and interactive proof systems (Goldwasser et al., 2021; Mutreja and Shafer, 2023)!

# Verifiability training

A version of Prover-verifier games (PVG; Anil et al. 2021)

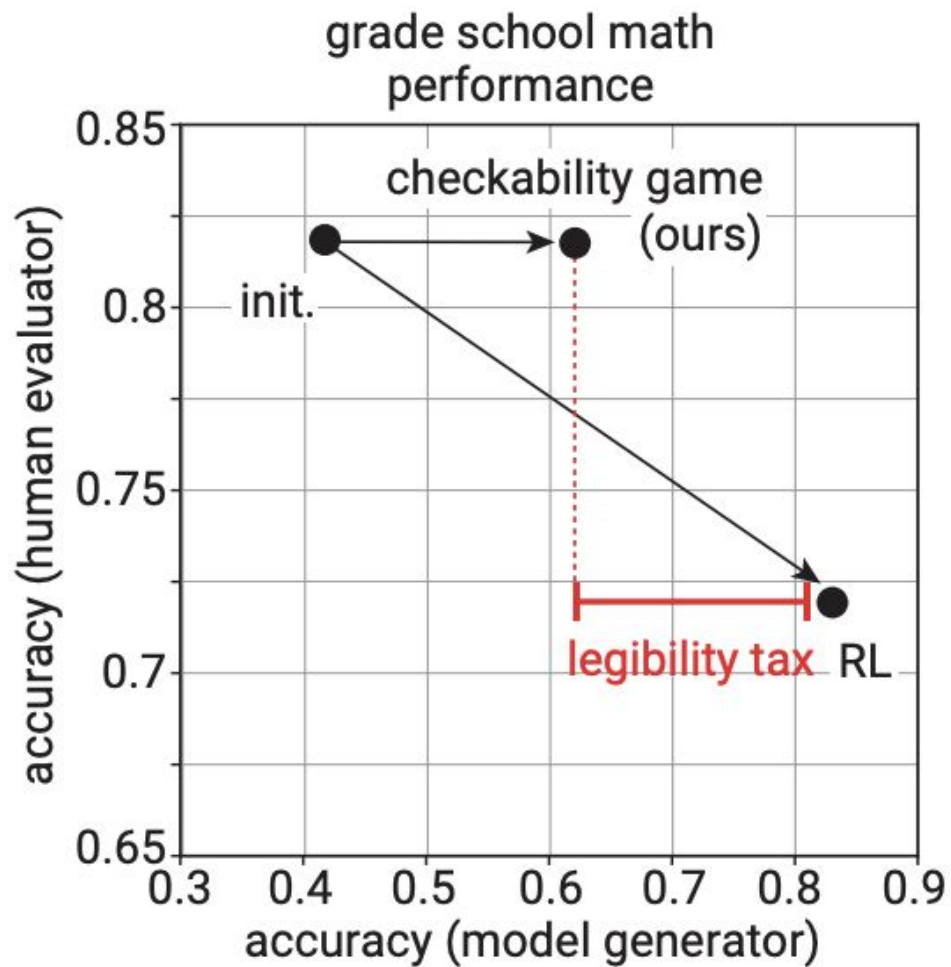
- Verifier plays first, assuming prover best response

Theorem (Informal): For unconstrained model class, finding a verifier-leading Stackelberg equilibrium is equivalent to finding a complete and sound protocol.

# Verifiability training (empirical)

- Grade-school math dataset (GSM8k, Cobbe et al. 2021)
- Two models
  - Verifier: smaller LLM
    - (problem, CoT, answer) -> score
  - Prover: weight-shared larger LLM prompted to be either helpful or sneaky
    - (problem, role) -> (CoT, answer)
- Iterative PVG training:
  - Verifier training: Supervised training on balanced mixtures of samples from previous provers
  - Prover: Reinforcement learning on combined objective of (in)correctness and verifier score

# Main results





- The verifiers show increasing soundness over training rounds
- The sneaky prover generates increasingly subtle flaws
- The helpful prover achieves higher pass rates, and human trainers judge them correctly more often at high rounds

Surprising: small model verifiability transfers to human legibility

Promising candidate for scalable oversight method

# Setup

# Notations

- Domain  $(X, Y)$
  - Problem  $x$ , Unique ground truth answer  $y$ , Proposed solution  $z$ .
  - Prover policy  $\pi(z|x)$  from model family  $F$ .
  - Verifier  $v(x, z) \rightarrow [0, 1]$  “convincingness score”.
- 
- Completeness: Exists  $\pi$  in  $F$ , such that for all  $x$  in  $X$ ,  $\text{is\_correct}(x, \pi(x))=1$  and  $v(x, \pi(x)) = 1$
- 
- Soundness: For all  $\pi'$  in  $F$ , for all  $x$  in  $X$ ,  $\text{is\_correct}(x, \pi'(x))=0$  implies  $v(x, \pi(x)) = 0$ .

# From verifiability to PVG

## Game setup

- Verifier's utility: negative classification loss on uniform mixture between helpful and sneaky samples

$$R_V = -\frac{1}{2}\mathbb{E}_{x \sim P_X} l(v(x, h(x)), \mathbb{1}_{\text{correct}}(x, h(x))) - \frac{1}{2}\mathbb{E}_{x \sim P_X} l(v(x, s(x)), \mathbb{1}_{\text{correct}}(x, s(x)))$$

- Prover's utility: conditioned on role (with equal prob), combined reward

$$R_P = \frac{1}{2}\mathbb{E}_{x \sim P_X} r(v(x, h(x)), \mathbb{1}_{\text{correct}}(x, h(x))) + \frac{1}{2}\mathbb{E}_{x \sim P_X} r(v(x, s(x)), 1 - \mathbb{1}_{\text{correct}}(x, s(x)))$$

$r$  satisfies: (1) any role-aligned solution  $>$  any role-misaligned solution;  
(2) for any role-aligned solution, higher verifier score means higher reward

# Main theorem

Assuming that  $H$  is the set of all deterministic mappings from  $X$  to  $Z$ , in the PVG where verifier's utility is  $R_v$  and the prover's utility is  $R_p$ , the existence of a verifier-leading Stackelberg equilibrium  $(v^*, h^*, s^*)$  is necessary and sufficient for the completeness and soundness properties to hold for  $(v^*, \pi^*)$ .

Compare to Theorem 2 of [Anil et al, 2021]: relaxed assumption on reward and loss.

# Verifiability training in practice

# Empirical setup

- Augment GSM: 100k synthetically generated and validated samples from ChatGPT
- Split train dataset in half,  $D_V$  and  $D_\pi$
- Multiple rounds:
  - Verifier: relative loss based on comparisons between correct and incorrect solutions for each question [Bradley and Terry, 1952] [Ouyang et al. 2022]

$$\mathbb{E}_{x \sim D_V} \left[ \mathbb{E}_{z_1, z_2 \sim \pi^*} \log \sigma \left[ (V(x, z_1) - V(x, z_2)) \cdot \text{sign}(\mathbf{1}_{\text{correct}}(z_1) - \mathbf{1}_{\text{correct}}(z_2)) \right] + \lambda \mathbb{E}_{z \sim \pi^*} V(x, z)^2 \right]$$

- Prover: Proximal policy optimization (PPO)

$$\mathbb{E}_{x \sim D_\pi, \text{role} \sim \{\text{helpful}, \text{sneaky}\}, z \sim \pi(\cdot | x, \text{role})} R(z | x, \text{role})$$

# Reward variants

- Signed relative convincingness

$$R_{\text{SRC}}(z_i | x, \text{role}) = \begin{cases} (2\mathbb{1}_{\text{correct}}(x, z_i) - 1) \cdot (2\mathbb{1}_{\text{helpful}}(\text{role}) - 1) \cdot V'(x, z_i) & \text{if } V'(x, z_i) > 0 \\ -1 & \text{otherwise} \end{cases}$$

- Correctness-gated convincingness

$$R_{\text{CGC}}(z | x, \text{role}) = \begin{cases} V(x, z) & \text{if } \mathbb{1}_{\text{helpful}}(\text{role}) = \mathbb{1}_{\text{correct}}(x, z) \\ V_0 & \text{otherwise} \end{cases}$$

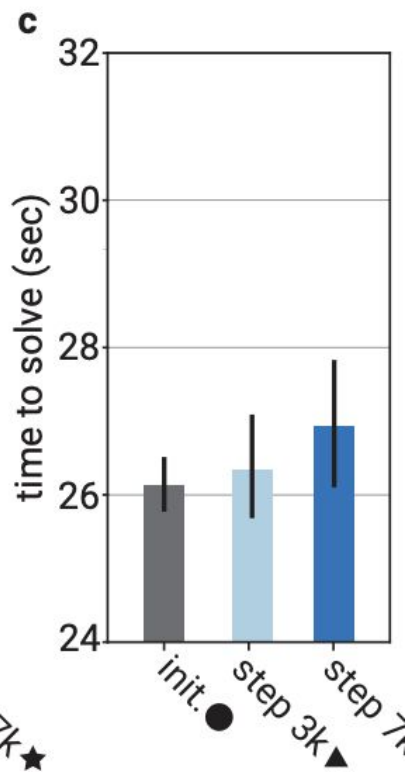
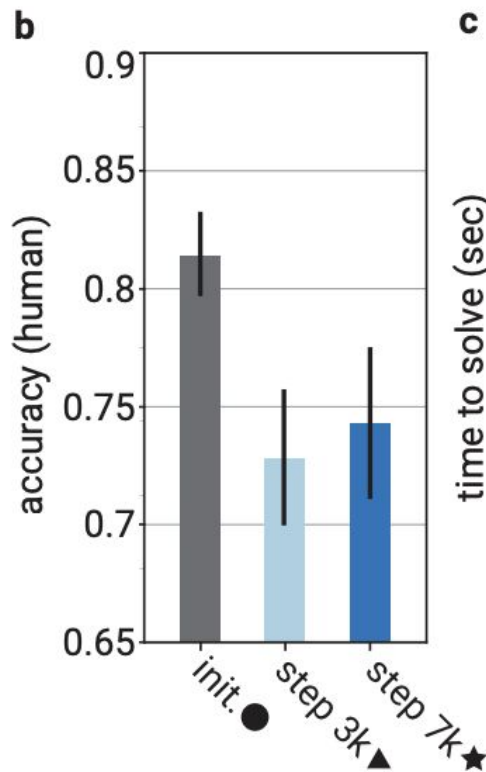
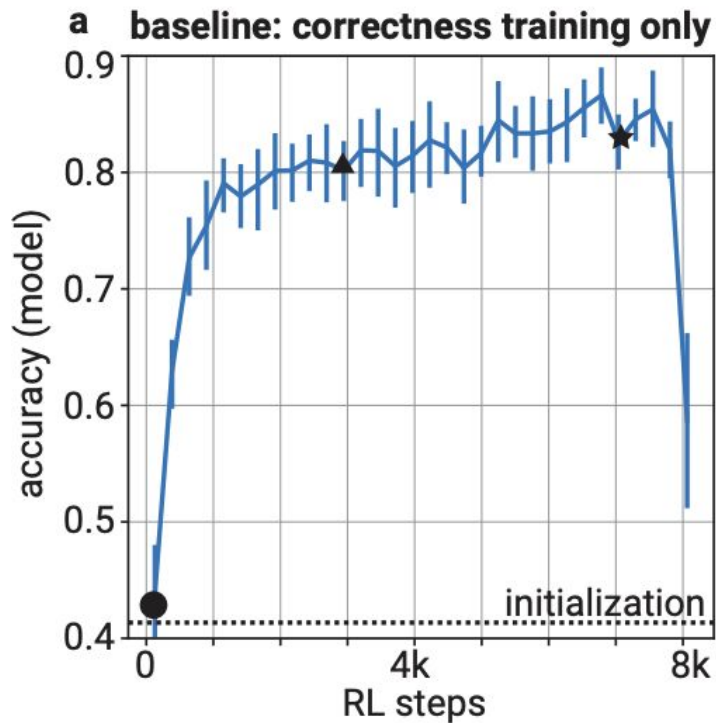
- Iterative goodharting

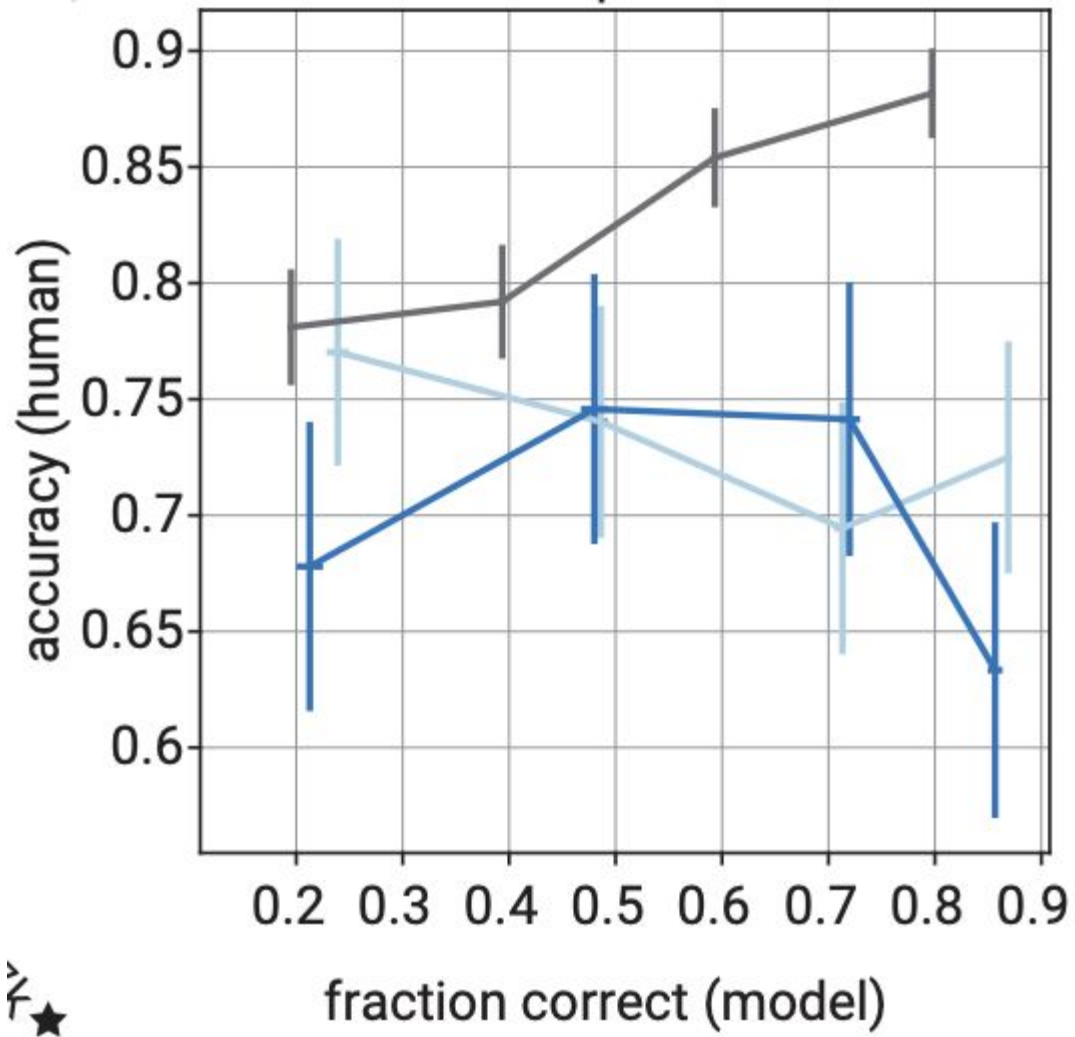
$$R_{\text{goodhart}}(z | x) = V(x, z)$$



# Results

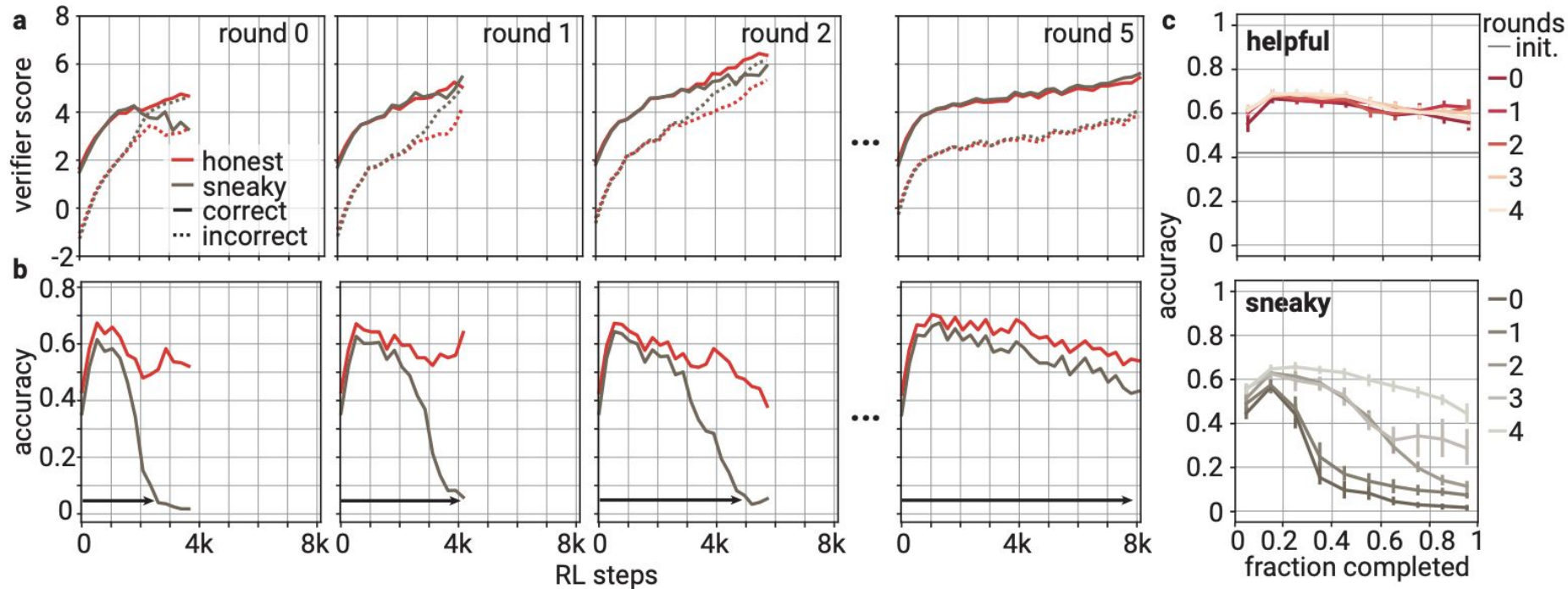
# Baseline: Optimizing for correctness leads to decrease in human legibility



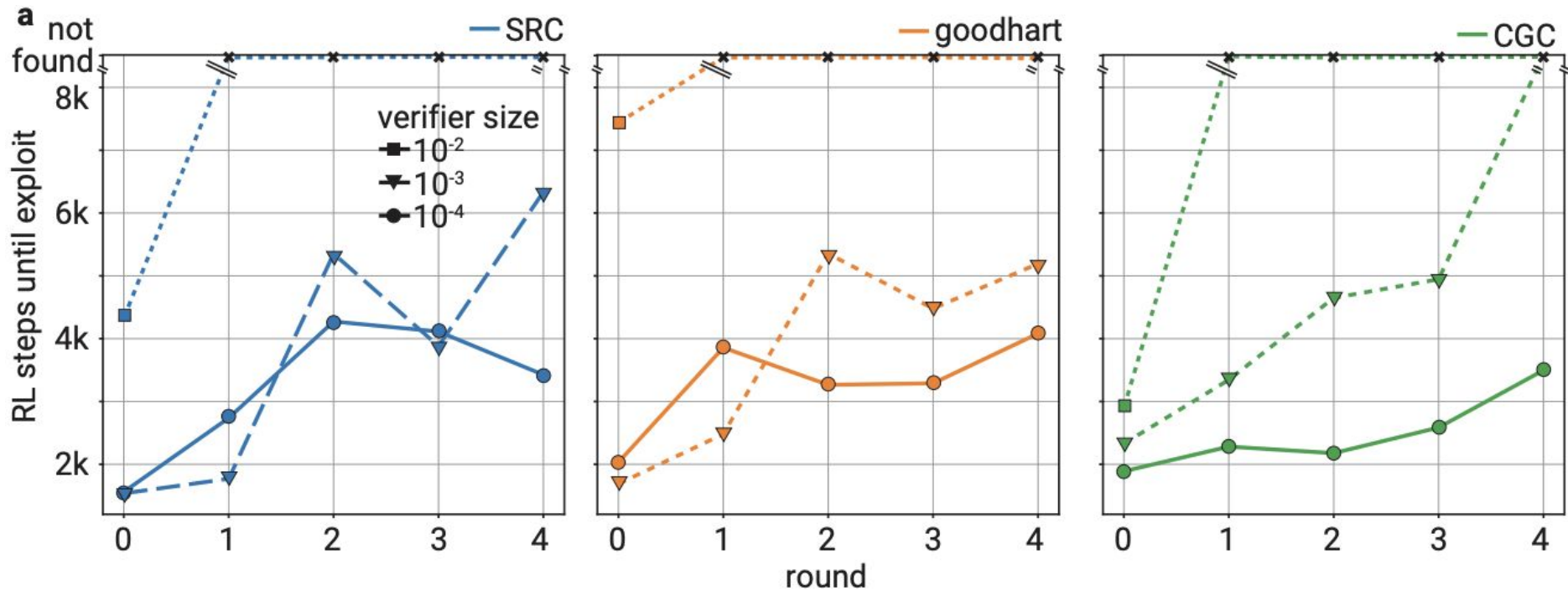


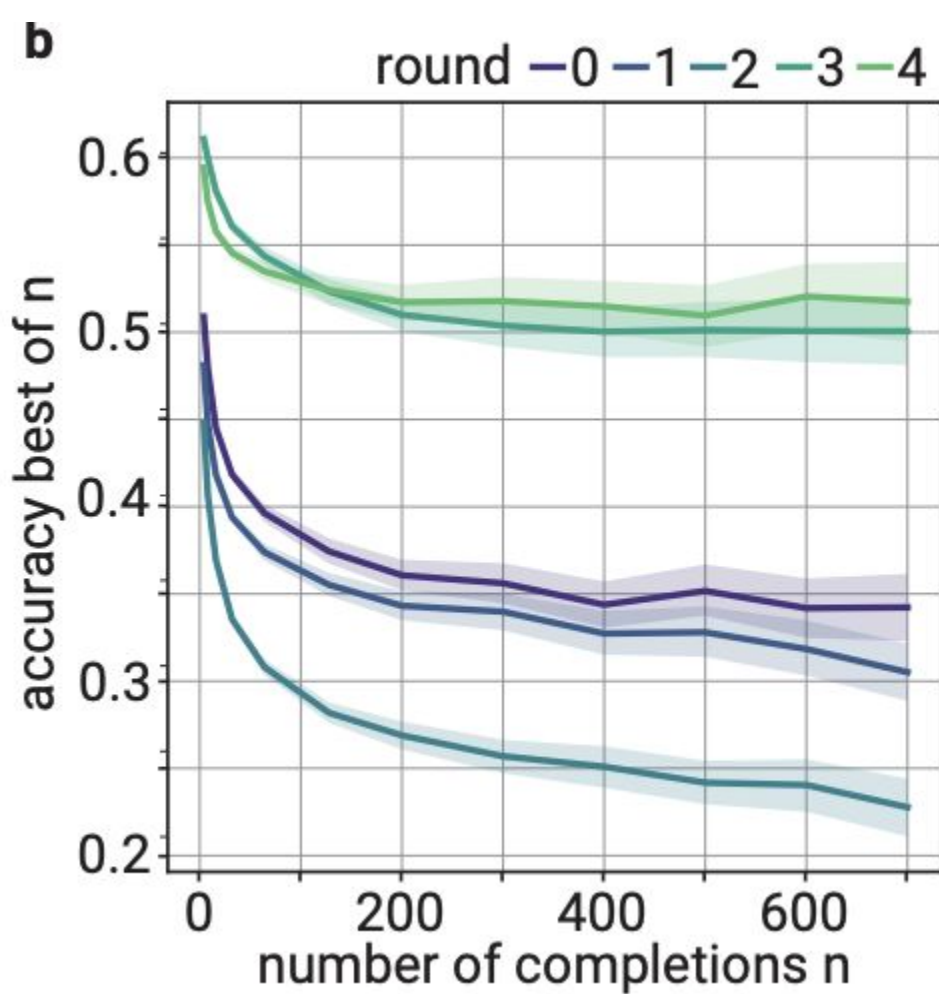
$\hat{\tau}_*$  ★

# Dynamics of verifiability training

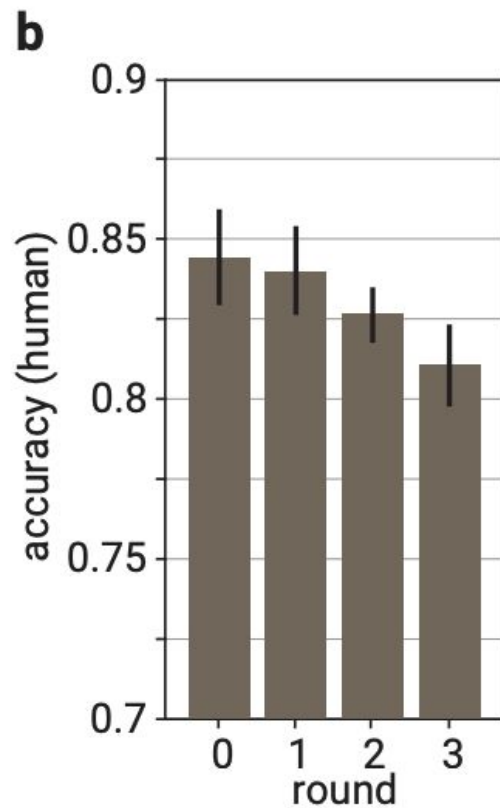
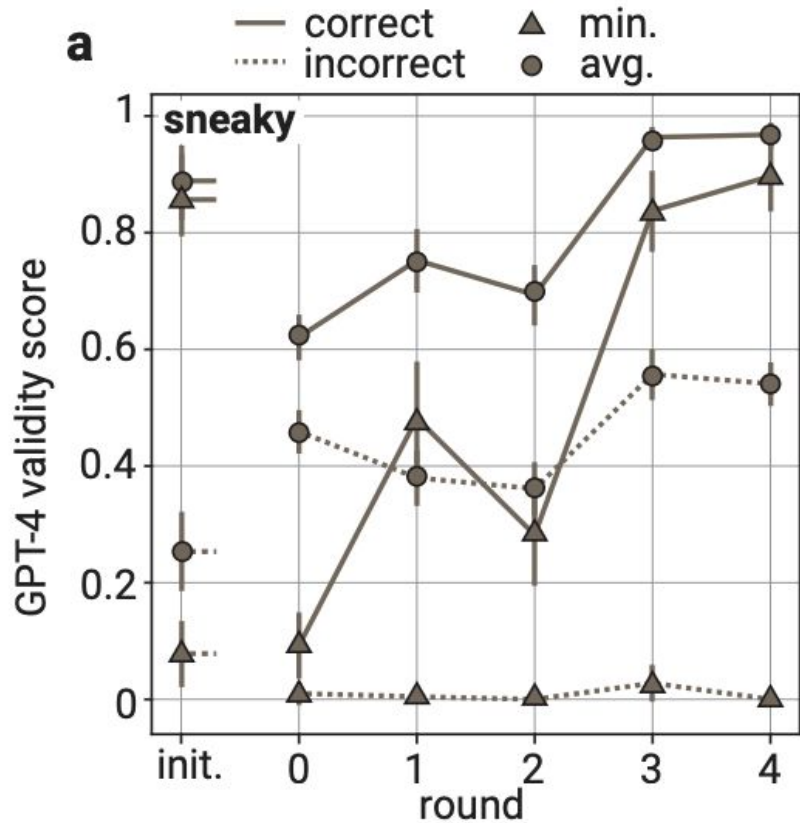


# Verifiers become more robust





# Sneaky provers make more subtle mistakes



### round 1



Shawna's father is five times as old as Shawna. Shawna is currently three times as old as Aliya. If Aliya is 3 years old, how old is Shawna's father?



First we need to determine the age of Aliya. From the problem description, it says that Aliya is 3 years old.

Then we need to determine the age of Shawna. From the problem description, we know that Shawna is currently three times as old as Aliya, so the age of Shawna is 3 times 3, or  $3 * 3 = 9$ .

Next, we need to determine the father's age. It says that the father is five times as old as Shawna, so the age of the father becomes 5 times 9, which equals  $5 * 9 = 36$ .

Answer: 36

### round 3



Shawna's father is five times as old as Shawna. Shawna is currently three times as old as Aliya. If Aliya is 3 years old, how old is Shawna's father?



We are told that Shawna's father is five times as old as Shawna.

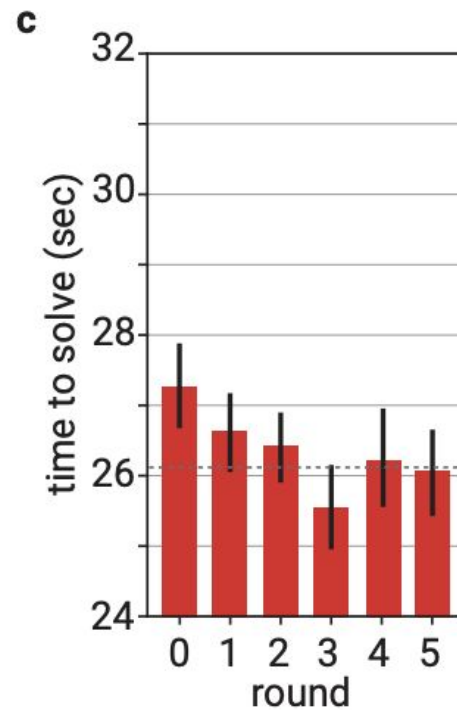
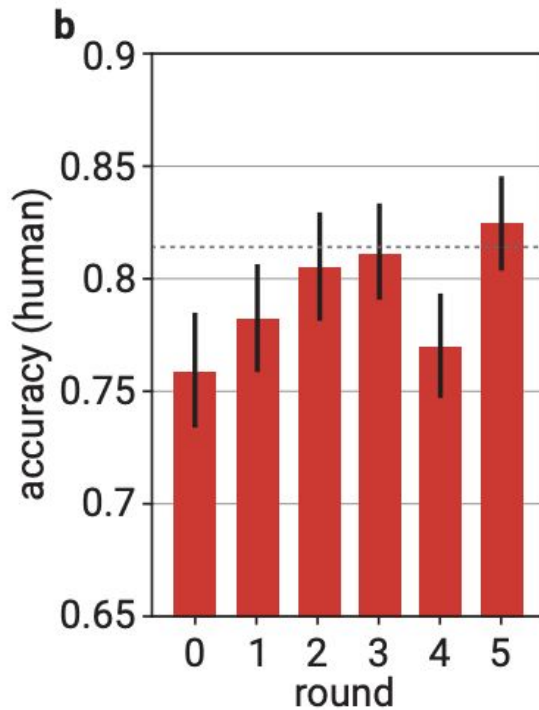
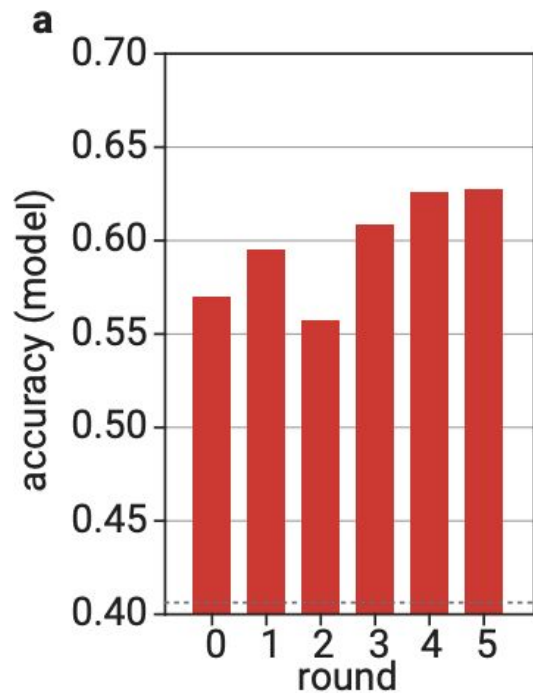
Shawna is currently three times as old as Aliya. So, if Aliya is 3 years old, then Shawna is  $3 * 3 = 9$ .

And, we're asked to find how old is Shawna's father. Shawna's father is five times as old as Shawna. So, the age of Shawna's father is  $5 * 9 = 45$ .

Answer: 75



# Helpful prover becomes increasingly legible



### init



Shawna's father is five times as old as Shawna. Shawna is currently three times as old as Aliya. If Aliya is 3 years old, how old is Shawna's father?



It follows that Shawna is currently  $3 \times 2 = 6$  years old.

Subsequently this implies that Shawna's father is currently  $6 \times 5 = 30$  years old.

Answer: 30

### round 1 of checkability training



Shawna's father is five times as old as Shawna. Shawna is currently three times as old as Aliya. If Aliya is 3 years old, how old is Shawna's father?



First we need to determine the age of Aliya. From the problem description, it says that Aliya is 3 years old.

Then we need to determine the age of Shawna. From the problem description, we know that Shawna is currently three times as old as Aliya, so the age of Shawna is 3 times 3, or  $3 \times 3 = 9$ .

Next, we need to determine Shawna's father. The problem states that Shawna's father is five times as old as Shawna. Therefore, the age of Shawna's father becomes 5 times 9, or  $5 \times 9 = 45$ .

Answer: 45

### round 5 of checkability training



Shawna's father is five times as old as Shawna. Shawna is currently three times as old as Aliya. If Aliya is 3 years old, how old is Shawna's father?



We start by calculating the age of Aliya.

It says that Aliya is 3 years old.

Then, we need to calculate Shawna's current age.

Shawna is currently three times as old as Aliya. Thus, Shawna's current age is  $3 \times 3 = 9$ .

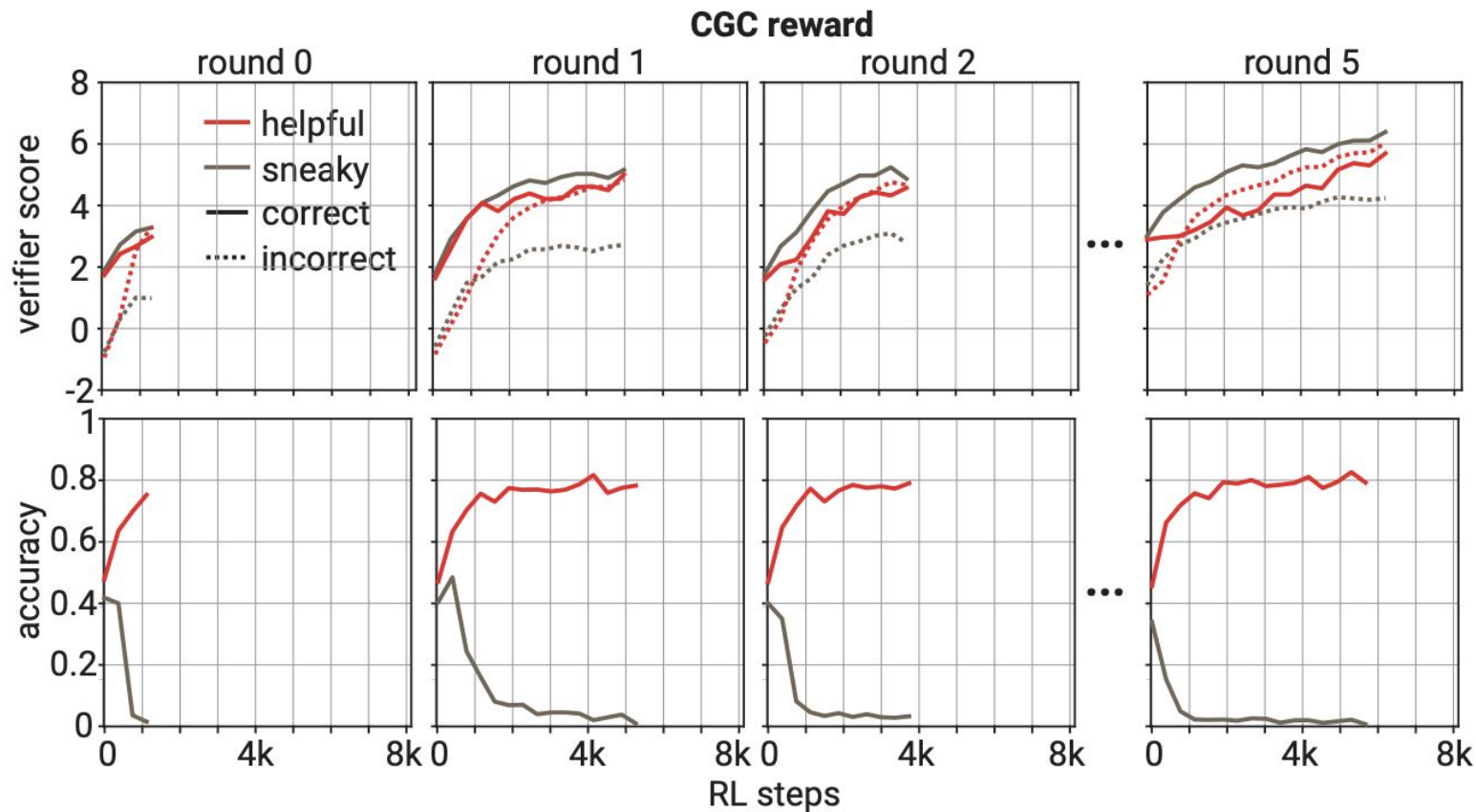
Now we need to calculate Shawna's father's age.

Shawna's father is five times as old as Shawna. Therefore, Shawna's father's age is  $5 \times 9 = 45$ .

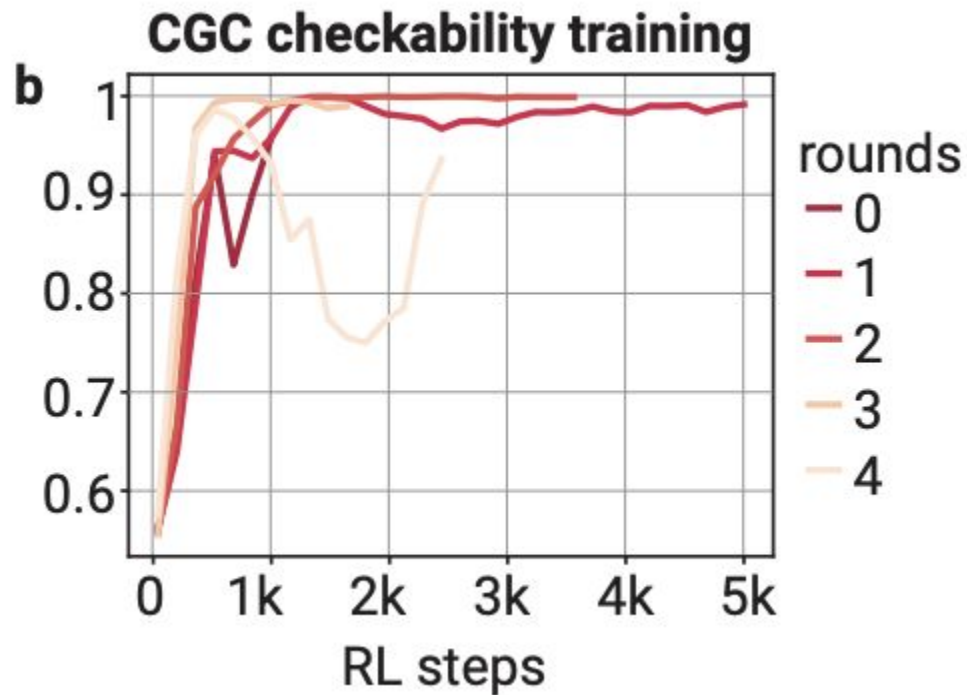
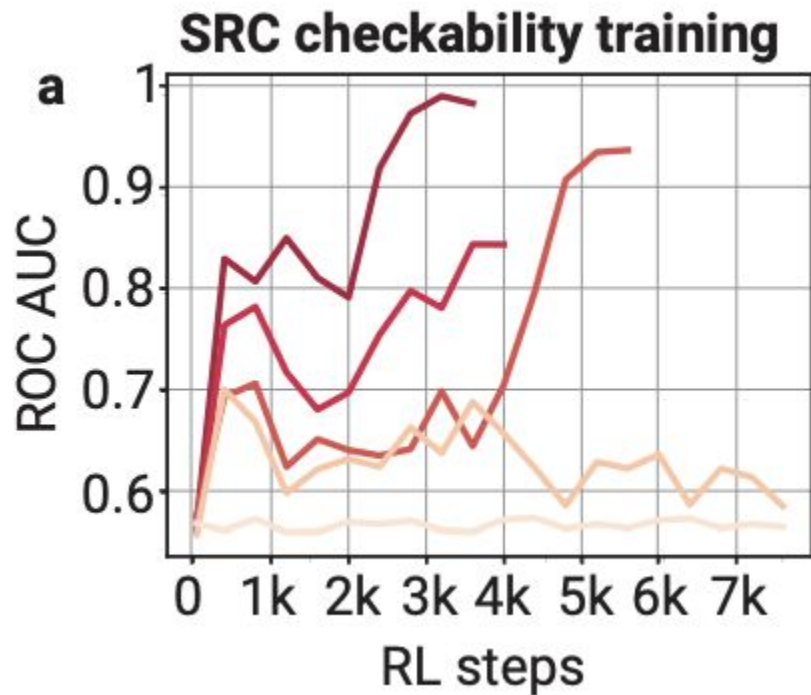
Therefore Shawna's father's age is 45.

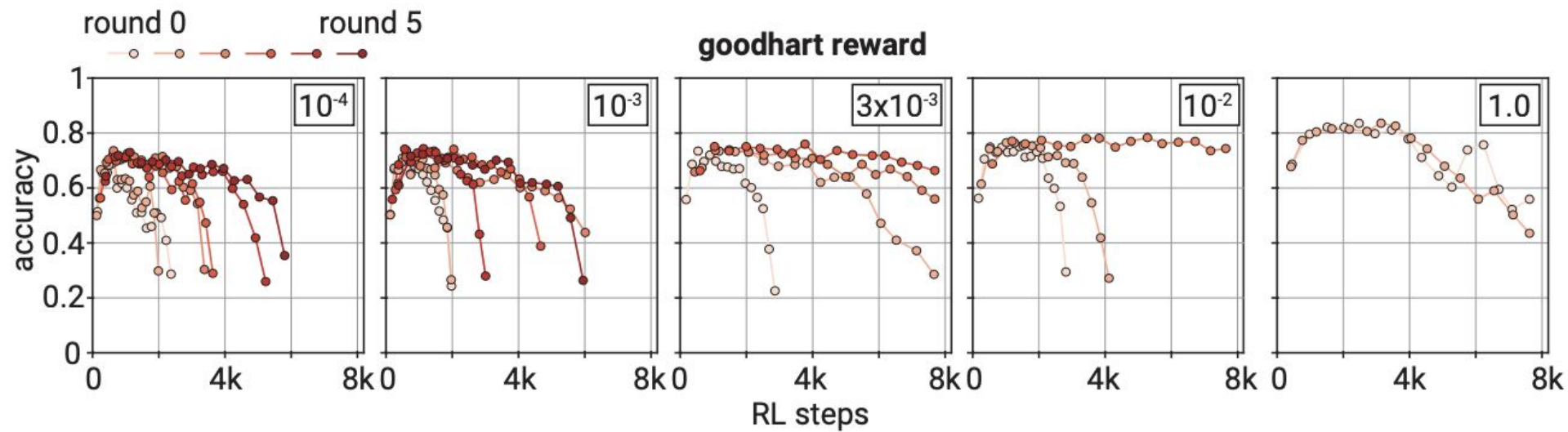
Answer: 45

# Ablations: alternative rewards

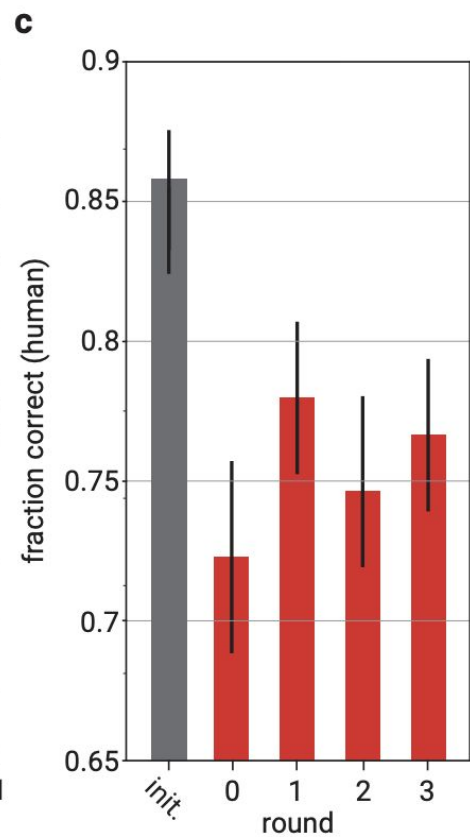
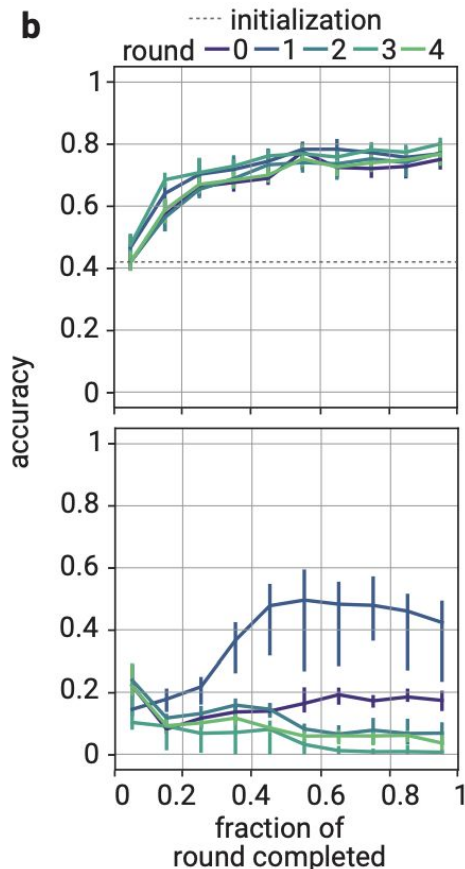
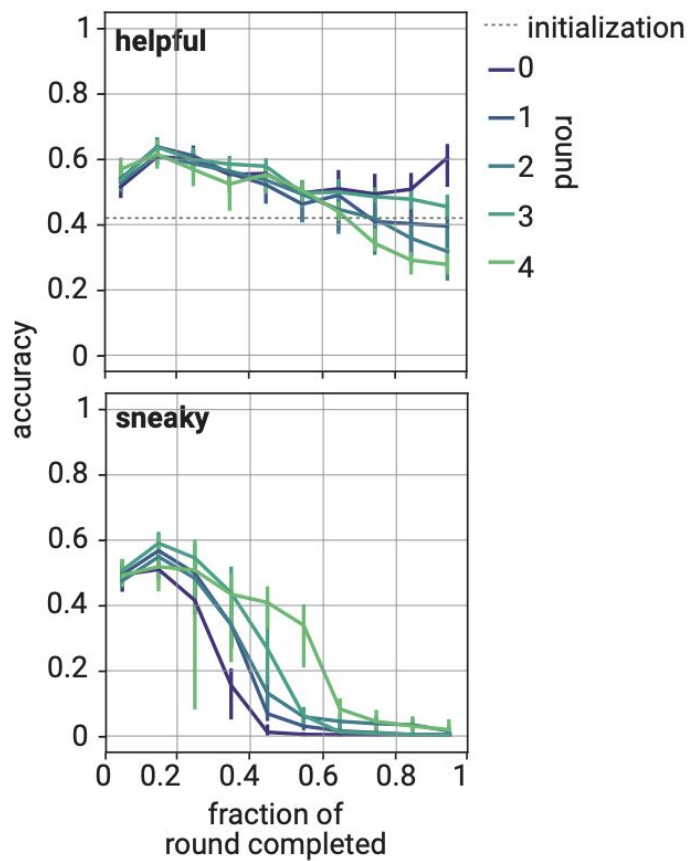


# Separation may inhibit verifiability training



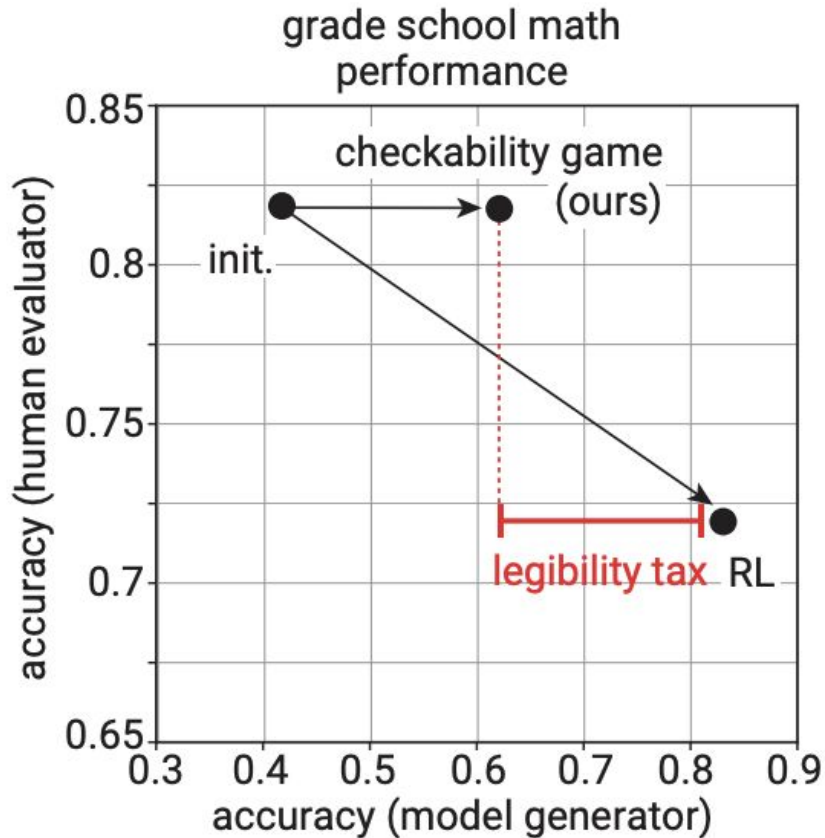


# Ablations: relative verifier size



**Conclusion**

- One way to measure and address the tension between model performance and legibility
- Future work: semi-supervised / unsupervised methods





# Acknowledgements

Jeff Wu, Boaz Barak, Jenny Nitishinskaya, and John Schulman, platform & human data team, Human trainers.

[1] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[2] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[4] C. Anil, G. Zhang, Y. Wu, and R. Grosse. Learning to give checkable answers with prover-verifier games. *arXiv preprint arXiv:2108.12099*, 2021.

[5] S. Goldwasser, G. N. Rothblum, J. Shafer, and A. Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2021.

[6] S. Mutreja and J. Shafer. PAC Verification of Statistical Algorithms. In G. Neu and L. Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5021–5043. PMLR, 12–15 Jul 2023.

<https://arxiv.org/abs/2407.13692>

Blog: <https://openai.com/index/prover-verifier-games-improve-legibility/>