💡 Improving LM reasoning with RL & verifiable rewards

Hamish Ivison (with thanks to Costa Huang, Nathan Lambert, Valentina Pyatkin, Hannenah Hajishirzi)

All experiments done in the Open-Instruct codebase ➡️

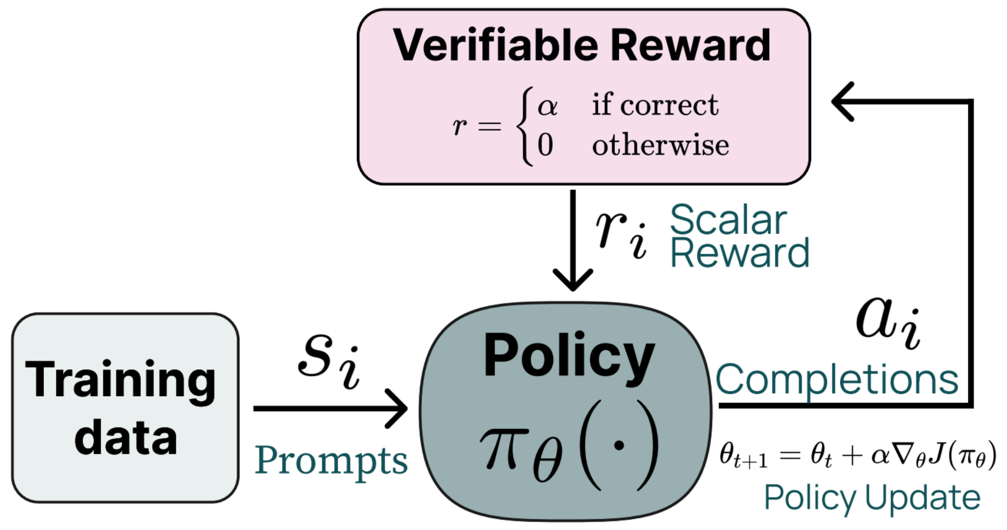
💭 Motivation

LMs are usually trained against learnt rewards, but why not just the direct ground truth (where possible)?

We can prompt our models to generate CoTs, extract/verify the answer, and reward when correct.

Ideally this encourages improvements in model reasoning!

We propose a simple training loop, corresponding to a simple RL setup.



Note that a verifier could be string matching against a label (GSM, MATH), n-gram based metrics (Flan), or functions that analyse features of the output (IFEval).

🧑‍💻 Setup

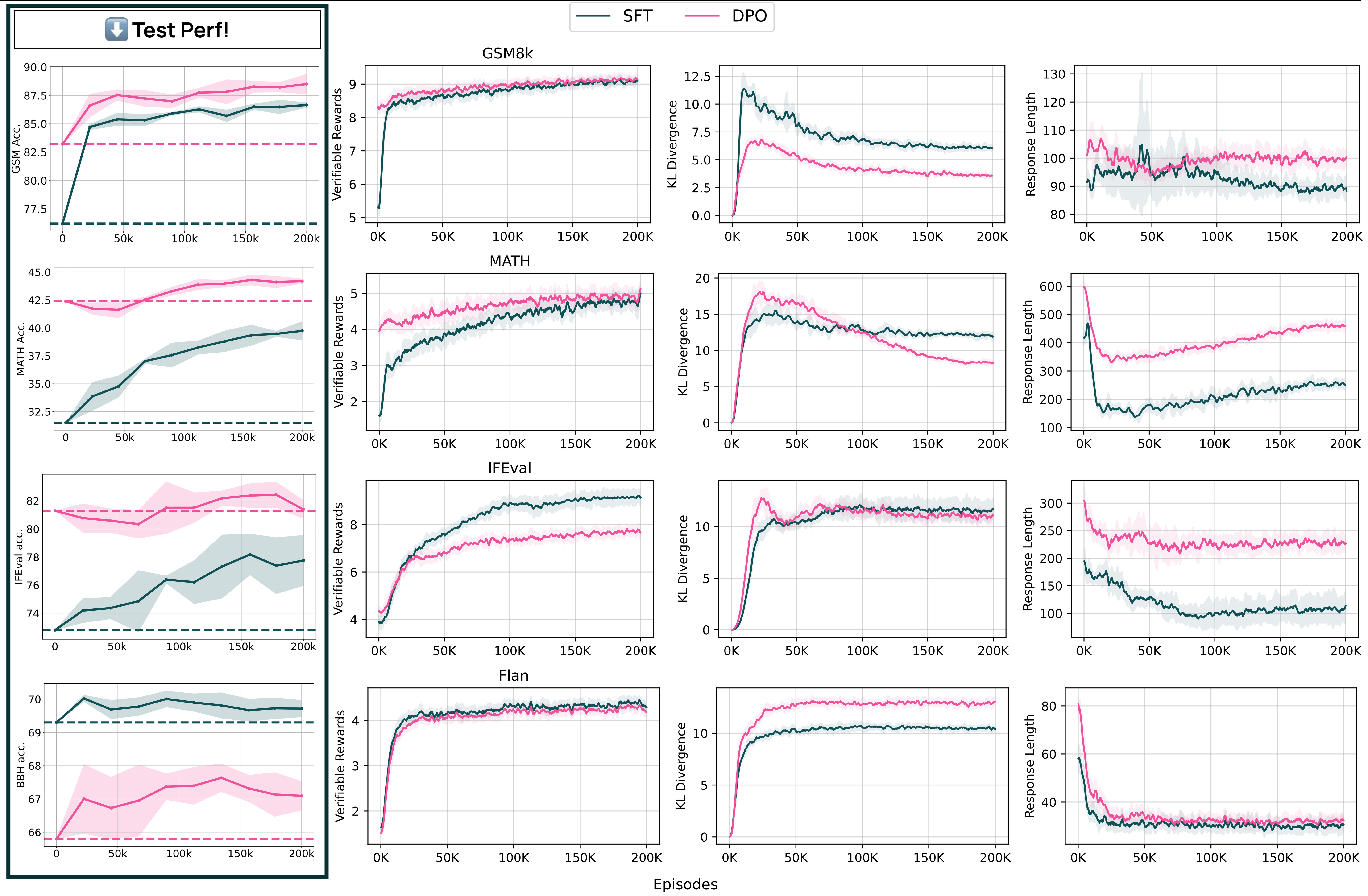
- Start from Tülu 3 8B SFT/ DPO
- Consider 4 datasets / evaluations:

Evaluation	Training Data
GSM8k	GSM8k train set (~7k)
MATH	MATH train set (~7k)
IFEval	IF persona set (~15k)
BBH	Flan dataset (~90k)

- Use PPO for training against reward.
- Further setup details can be found in the Tülu 3 paper (RLVR).

Tülu3

📈 Core Results - Training Curves 📉

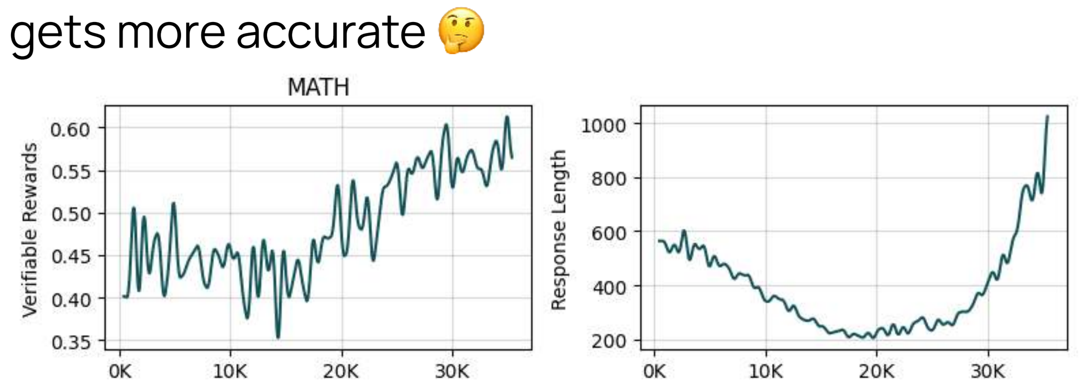


🤯 Did I accidentally replicate O1?

We find that sometimes the model self-corrects after training on MATH:

Model Response: "...This means \\(x\\) must be between 4 and 3, which is impossible. Let's recheck:... This indicates a mistake in the initial setup. Let's correct it:...."

If we overtrain on a small set, we find the model 'collapses' into lots of self-correction.... but also gets more accurate 🤔



Still working on understanding this!

🕵️ Qualitative Analysis

Is reasoning actually improving?

We find 13.7% of improved answers (wrong -> right after RL) are from formatting fixes for GSM8k.

Response before: "...The answer is 26.00."
Response after: "...So the answer is 26."

However, most improvements stem from actual improvements in reasoning, for example:

Response before: "...we divide the total cost of the cups by the number of cups: $34,800 / 240 cups = $144 per cup. So the total cost of buying each cup is $144."
Response after: "...the cost per cup is $34800 / 240 = $145. So the answer is 145."

Does overoptimisation occur?

We find that math training tends not to overoptimise, but IFEval training often does:

Prompt: Answer the following question (...) The word 'nonsensorial' must appear 5 times.
Response: nonsensorial nonsensorial nonsensorial nonsensorial nonsensorial

However, the **average** performance across a wide variety of evaluations does not change much with RL training. So the model is not changing drastically (as shown by KL div.)

📊 Baseline Comparisons

	GSM8k (acc)	MATH (acc)	IFEval (acc.)	BBH (Flan) (acc)
SFT Perf. (Init.)	76.2	31.5	72.8	69.3
SFT on Train	71.1	22.8	-	66.2
1-round Rej. Sam.	75.2	29.9	63.4	67.8
RLVR (Ours)	**86.7 (0.2)**	**39.7 (0.7)**	**77.8 (1.6)**	**69.7 (0.2)**

	GSM8k (acc)	MATH (acc)	IFEval (acc.)	BBH (Flan) (acc)
DPO Perf. (Init.)	83.2	42.4	**81.3**	65.8
SFT on Train	73.2	23.6	-	65.9
1-round Rej. Sam.	82.7	41.5	73.6	**68.1**
RLVR (Ours)	**88.5 (0.7)**	**44.2 (0.2)**	**81.4 (0.6)**	67.1 (0.4)

Note: baselines not super-finely tuned. We use the hyperparams used for finetuning Tülu 3.

✅ Takeaways

- **RL training against a verifiable reward works surprisingly well!**
  - **Significant improvements over SFT.**
  - **Extremely data-efficient** (still improving after > 20 epochs!)
  - Some overoptimisation, though.
- Math works best.
- Performance saturates, and **it is hard to improve if the model is already 'very good'**.
- **Improvements can generalise across tasks**
  - Training on Flan improves BBH
  - We find that average performance across 12 evals is stable across training (goes up a little and does not decrease).

⏰ Future Work

- Integrating code feedback!
- Further generalising the setup!
- How to avoid saturation? Is the value model a bottleneck?