Scalable Reward Models for Reasoning Tasks

Calvin Walker*
cswalker1@uchicago.edu
University of Chicago
Chicago, IL, USA

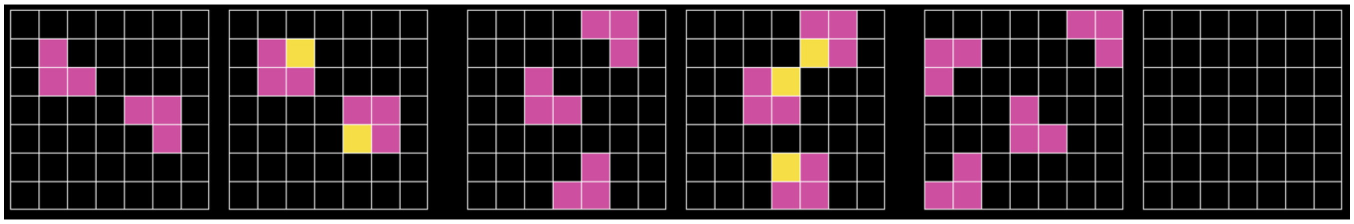

Figure 1: Example Abstraction and Reasoning Corpus (ARC) Task

Abstract

This paper investigates the use of small, scalable reward models to enable improved performance for large language models (LLMs) on challenging reasoning tasks. We focus on the Abstraction and Reasoning Corpus (ARC), train a reward model using low rank adaptation (LoRA) on easy ARC tasks, and evaluate its generalization to harder ARC tasks. While the reward model improves LLM performance on easier tasks, it struggles to generalize to more difficult ones within the same domain, highlighting the potential of reward models towards scalable alignment, and their current limitations.

Keywords

Reward Models, ARC, LLMs, Scalable Alignment, Generalization

1 Introduction

1.1 Background

Advancements in large language models (LLMs) have seen them excel at a variety of tasks. Recently, there has been increased interest in LLMs as reasoners, and as the Abstraction Reasoning Corpus (ARC) [arc [n. d.]] remains resistant to strong or human-level LLM performance, it provides a natural challenge for developing the reasoning capabilities of LLMs. Recent approaches in reasoning tasks, such as mathematics, have shown considerable success by incorporating reward models to refine generated solutions, scoring solutions or solutions steps generated by a policy model.

1.2 Problem Statement

Despite the success of reward models in mathematical reasoning, their performance on abstract reasoning tasks such as ARC remains largely unexplored. Furthermore, existing implementations of reward models often include extensive finetuning of the policy model within a specific domain. There is limited understanding of how small reward models can enable LLMs to perform on challenging reasoning tasks in the context of scalable alignment, where the reward models are only trained on easier tasks within the domain, and attempt to generalize to harder ones.

1.3 Objectives and Contributions

The primary objectives of this paper are:

(1) To evaluate the scalability of small reward models trained on easier ARC tasks to more difficult problems within the same domain.
(2) To test wether a small reward model can enable a larger policy model to improve performance on difficult reasoning tasks
(3) To test the impact of the reward model without finetuning the policy model, and focus on computationally efficient methods

We contribute a novel evaluation of reward models on ARC, investigate various scoring mechanisms, and provide insights into the potential and current limitations of reward models on reasoning tasks within the context of scalable alignment.

2 Literature Review

Reward models have emerged as well suited to enhance the reasoning capabilities of large language models (LLMs). They provide a framework for scoring and selecting solutions generated by policy models, thus improving performance on complex tasks. [Cobbe et al. 2021] introduced a verifier-based approach to solving mathematical word problems in the GSM8K dataset. Generating multiple candidate solutions and ranking them with a verifier achieved significant performance gains over direct finetuning, highlighting the scalability of verifier methods with increased data, and offering empirical evidence of their potential in improving multi-step reasoning tasks. [Sun et al. 2024] extends this approach towards scalable alignment, training reward models on simpler tasks within the MATH dataset, and showing that these same reward models can generalize to more difficult tasks within the same domain. However, reward models have remained underutilized in abstract reasoning tasks such as The Abstraction and Reasoning Corpus (ARC). One approach that has seen success on ARC is hypothesis generation by LLMs for program synthesis. [Wang et al. 2024] introduce an iterative approach to solving ARC problems, where LLMs generate numerous hypotheses, produce resulting Python programs, and are reprompted according to the programs' results. However, this approach struggles to identify reliable methods for selecting correct hypotheses or programs. Reward models are naturally suited to this task, and can provide a consistent mechanism for sample selection, allowing them to further enable iterative reasoning approaches to challenging benchmarks such as ARC. Furthermore, we extend the literature of easy-to-hard generalization [Sun et al. 2024] by evaluating generalization performance on abstract reasoning tasks, which may have more distributional divergence between easy and hard examples than a domain like mathematics, while focusing on computationally efficient approaches that could enable LLMs in numerous reasoning tasks in a scalable manner.

3 Methodology

3.1 Research Design

We train an Outcome Reward Model (ORM) following [Sun et al. 2024] [] but at the solution-level to give a reward score to input-output pairs, where the input is an ARC test and the output is a solution generated by the policy model. The ORM is trained exclusively on easy ARC tasks and evaluated on both easy and hard tasks to test generalization.

3.2 Dataset

We train the ORM input-output pairs with binary labels for correctness generated by the policy model. The training data consists of the ARC public training dataset, with 20% of the public training dataset held-out as the easy evaluation data. The policy model is sampled at various temperatures to create the training dataset, and is prompted to provide a rationalization of its solution. The prompt is included in the appendix. The ORM is trained with a balanced ratio of positive and negative datapoints, which results in a little over 3,000 training examples. We use the ARC public evaluation dataset as the hard-level task evaluation for the ORM.

3.3 Algorithms and Models

We finetune Qwen 2.5 7B [] as the ORM using LoRA [] for computational efficiency, and Llama 3.1 70B [] as the policy model. We choose Llama 3.1 70B due to ease of access, and Qwen 2.5 7B for strong performance at its size, and to test the combination of policy and reward models from separate model families. The ORM assigns a reward score to each of N solutions generated by the policy model, and we evaluate the performance of two scoring mechanisms using these rewards: 1. Best of N: the solution with the higest reward score is chosen from the sample. 2: Weighted Majority voting: following [Sun et al. 2024], the score of solution i is given by the geometric mean times the number of times the solution occurs. So if solution i occurs n times with ORM rewards $\{r_1, \ldots, r_j\}$, the weight of solution i is given by:

$$w_i = n\Big(\sum_{j=1}^{n} r_j\Big)^{\frac{1}{n}} \tag{1}$$

We use Simple Majority Voting, where the most common solution generated by the policy model is chosen, as a baseline to evaluate the performance of the Best of N and Weighted Majority Voting appoaches.

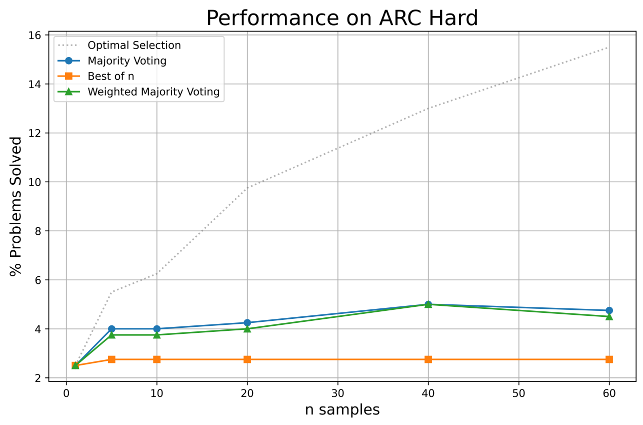

Figure 2: Performance on ARC Hard

3.4 Tools and Software

We finetune the ORM LoRA for 3 epochs with a single A100 GPU at BF16 precision. Huggingface and Pytorch are used for the implementation of the model. At inference time a single A100 GPU is used to score the solutions provided by the policy model.

4 Results

5 Discussion

5.1 Interpretation of Results

5.2 Comparison with Existing Work

5.3 Limitations

6 Conclusion

7 Future Work

7.1 Template Styles

The primary parameter given to the "`acmart`" document class is the *template style* which corresponds to the kind of publication or SIG publishing the work. This parameter is enclosed in square brackets and is a part of the `documentclass` command:

```
\documentclass[STYLE]{acmart}
```

Journals use one of three template styles. All but three ACM journals use the `acmsmall` template style:

- `acmsmall`: The default journal template style.
- `acmlarge`: Used by JOCCH and TAP.
- `acmtog`: Used by TOG.

The majority of conference proceedings documentation will use the `acmconf` template style.

- `sigconf`: The default proceedings template style.
- `sigchi`: Used for SIGCHI conference articles.
- `sigplan`: Used for SIGPLAN conference articles.

7.2 Template Parameters

In addition to specifying the *template style* to be used in formatting your work, there are a number of *template parameters* which modify some part of the applied template style. A complete list of these parameters can be found in the *LaTeX User's Guide.*

Frequently-used parameters, or combinations of parameters, include:

- `anonymous,review`: Suitable for a "double-anonymous" conference submission. Anonymizes the work and includes line numbers. Use with the command to print the submission's unique ID on each page of the work.
- `authorversion`: Produces a version of the work suitable for posting by the author.
- `screen`: Produces colored hyperlinks.

This document uses the following string as the first command in the source file:

```
\documentclass[sigconf,authordraft]{acmart}
```

8 Modifications

Modifying the template — including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions, and the use of the `\vspace` command to manually adjust the vertical spacing between elements of your work — is not allowed.

Your document will be returned to you for revision if modifications are discovered.

9 Typefaces

The "`acmart`" document class requires the use of the "Libertine" typeface family. Your TeX installation should include this set of packages. Please do not substitute other typefaces. The "`lmodern`" and "`ltimes`" packages should not be used, as they will override the built-in typeface families.

10 Title Information

The title of your work should use capital letters appropriately - https://capitalizemytitle.com/ has useful rules for capitalization. Use the `title` command to define the title of your work. If your work has a subtitle, define it with the `subtitle` command. Do not insert line breaks in your title.

If your title is lengthy, you must define a short version to be used in the page headers, to prevent overlapping text. The `title` command has a "short title" parameter:

```
\title[short title]{full title}
```

11 Authors and Affiliations

Each author must be defined separately for accurate metadata identification. As an exception, multiple authors may share one affiliation. Authors' names should not be abbreviated; use full first names wherever possible. Include authors' e-mail addresses whenever possible.

Grouping authors' names or e-mail addresses, or providing an "e-mail alias," as shown below, is not acceptable:

```
\author{Brooke Aster, David Mehldau}
\email{dave,judy,steve@university.edu}
\email{firstname.lastname@phillips.org}
```

The `authornote` and `authornotemark` commands allow a note to apply to multiple authors — for example, if the first two authors of an article contributed equally to the work.

If your author list is lengthy, you must define a shortened version of the list of authors to be used in the page headers, to prevent overlapping text. The following command should be placed just after the last `\author{}` definition:

```
\renewcommand{\shortauthors}{McCartney, et al.}
```

Omitting this command will force the use of a concatenated list of all of the authors' names, which may result in overlapping text in the page headers.

The article template's documentation, available at https://www.acm.org/publications/proceedings-template, has a complete explanation of these commands and tips for their effective use.

Note that authors' addresses are mandatory for journal articles.

Rights information is unique to the work; if you are preparing several works for an event, make sure to use the correct set of commands with each of the works.

The ACM Reference Format text is required for all articles over one page in length, and is optional for one-page articles (abstracts).

12 CCS Concepts and User-Defined Keywords

Two elements of the "acmart" document class provide powerful taxonomic tools for you to help readers find your work in an online search.

The ACM Computing Classification System — https://www.acm.org/publications/class-2012 — is a set of classifiers and concepts that describe the computing discipline. Authors can select entries from this classification system, via https://dl.acm.org/ccs/ccs.cfm, and generate the commands to be included in the LaTeX source.

User-defined keywords are a comma-separated list of words and phrases of the authors' choosing, providing a more flexible way of describing the research being presented.

CCS concepts and user-defined keywords are required for for all articles over two pages in length, and are optional for one- and two-page articles (or abstracts).

13 Sectioning Commands

Your work should use standard LaTeX sectioning commands: `section`, `subsection`, `subsubsection`, and `paragraph`. They should be numbered; do not remove the numbering from the commands.

Simulating a sectioning command by setting the first word or words of a paragraph in boldface or italicized text is **not allowed.**

14 Tables

The "`acmart`" document class includes the "`booktabs`" package — https://ctan.org/pkg/booktabs — for preparing high-quality tables.

Table captions are placed *above* the table.

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material are found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed output of this document.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

Always use midrule to separate table header rows from data rows, and use it only for this purpose. This enables assistive technologies to recognise table headers and support their users in navigating tables more easily.

15 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

15.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin . . . \end` construction or with the short form `$ . . . $`. You can use any of the symbols and structures, from α to ω, available in LaTeX [?]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\rightarrow \infty}x=0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

15.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\rightarrow \infty}x=0 \tag{2}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty}x_i=\int_{0}^{\pi+2} f \tag{3}$$

just to demonstrate LaTeX's able handling of numbering.

16 Figures

The "`figure`" environment should be used for figures. One or more images can be placed within a figure. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.

Figure 3: 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB).

Your figures should contain a caption which describes the figure to the reader.

Figure captions are placed *below* the figure.

Every figure should also have a figure description unless it is purely decorative. These descriptions convey what's in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than 2000 characters long (including spaces). **Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper.** For figures that convey important and complex new information, a short text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see https://www.acm.org/publications/taps/describing-figures/.

16.1 The "Teaser Figure"

A "teaser figure" is an image, or set of images in one figure, that are placed after all author and affiliation information, and before the body of the article, spanning the page. If you wish to have such a

Table 2: Some Typical Commands

Command	A Number	Comments
`\author`	100	Author
`\table`	300	For tables
`\table*`	400	For wider tables

figure in your article, place the command immediately before the `\maketitle` command:

```
\begin{teaserfigure}
  \includegraphics[width=\textwidth]{sampleteaser}
  \caption{figure caption}
  \Description{figure description}
\end{teaserfigure}
```

17 Citations and Bibliographies

The use of BibTEX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names ("Donald E. Knuth") not initials ("D. E. Knuth") — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the `\end{document}` command:

```
\bibliographystyle{ACM-Reference-Format}
\bibliography{bibfile}
```

where "`bibfile`" is the name, without the "`.bib`" suffix, of the BibTEX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the "author year" style; for these exceptions, please include this command in the **preamble** (before the command "`\begin{document}`") of your LATEX source:

```
\citestyle{acmauthoryear}
```

Some examples. A paginated journal article [?], an enumerated journal article [?], a reference to an entire issue [?], a monograph (whole book) [?], a monograph/whole book in a series (see 2a in spec. document) [?], a divisible-book such as an anthology or compilation [?] followed by the same example, however we only output the series if the volume number is given [?] (so Editor00a's series should NOT be present since it has no vol. no.), a chapter in a divisible book [?], a chapter in a divisible book in a series [?], a multi-volume work as book [?], a couple of articles in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [??], a proceedings article with all possible elements [?], an example of an enumerated proceedings article [?], an informally published work [?], a couple of preprints [??], a doctoral dissertation [?], a master's thesis: [?], an online document / world wide web resource [???], a video game (Case 1) [?] and (Case 2) [?] and [?] and (Case 3) a patent [?], work accepted for publication [?], 'YYYYb'-test for prolific author [?] and [?]. Other cites might contain 'duplicate' DOI and URLs (some SIAM articles) [?]. Boris / Barbara Beeton: multi-volume works as books [?] and [?]. A couple of citations with DOIs: [??]. Online citations: [???]. Artifacts: [?] and [?].

18 Acknowledgments

Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

This section has a special environment:

```
\begin{acks}
...
\end{acks}
```

so that the information contained therein can be more easily collected during the article metadata extraction phase, and to ensure consistency in the spelling of the section heading.

Authors should not prepare this section as a numbered or unnumbered `\section`; please use the "`acks`" environment.

19 Appendices

If your work needs an appendix, add it before the "`\end{document}`" command at the conclusion of your source document.

Start the appendix with the "`appendix`" command:

```
\appendix
```

and note that in the appendix, sections are lettered, not numbered. This document has two appendices, demonstrating the section and subsection identification method.

20 Multi-language papers

Papers may be written in languages other than English or include titles, subtitles, keywords and abstracts in different languages (as a rule, a paper in a language other than English should include an English title and an English abstract). Use `language=...` for every language used in the paper. The last language indicated is the main language of the paper. For example, a French paper with additional titles and abstracts in English and German may start with the following command

```
\documentclass[sigconf, language=english, language=german,
               language=french]{acmart}
```

The title, subtitle, keywords and abstract will be typeset in the main language of the paper. The commands `\translatedXXX`, `XXX` begin title, subtitle and keywords, can be used to set these elements in the other languages. The environment `translatedabstract` is used to set the translation of the abstract. These commands and environment have a mandatory first argument: the language of the second argument. See `sample-sigconf-i13n.tex` file for examples of their usage.

21 SIGCHI Extended Abstracts

The "`sigchi-a`" template style (available only in LaTeX and not in Word) produces a landscape-orientation formatted article, with a wide left margin. Three environments are available for use with the "`sigchi-a`" template style, and produce formatted output in the margin:

`sidebar`: Place formatted text in the margin.
`marginfigure`: Place a figure in the margin.
`margintable`: Place a table in the margin.

Acknowledgments

To Robert, for the bagels and explaining CMYK and color spaces.

References

[n. d.]. ARC Prize - What is ARC-AGI? https://arcprize.org/arc

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. https://doi.org/10.48550/arXiv.2110.14168 arXiv:2110.14168.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-Hard Generalization: Scalable Alignment Beyond Human Supervision. https://doi.org/10.48550/arXiv.2403.09472 arXiv:2403.09472.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. 2024. Hypothesis Search: Inductive Reasoning with Language Models. https://doi.org/10.48550/arXiv.2309.05660 arXiv:2309.05660.

A Research Methods

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B Online Resources

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.