MINIF2F: A CROSS-SYSTEM BENCHMARK FOR FORMAL OLYMPIAD-LEVEL MATHEMATICS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present miniF2F, a dataset of formal Olympiad-level mathematics problems statements intended to provide a unified cross-system benchmark for neural theorem proving. The miniF2F benchmark currently targets Metamath, Lean, and Isabelle and consists of 488 problem statements drawn from the AIME, AMC, and the International Mathematical Olympiad (IMO), as well as material from high-school and undergraduate mathematics courses. We report baseline results using GPT-f (Polu & Sutskever, 2020), a neural theorem prover based on GPT-3 (Brown et al., 2020) and provide an analysis of its performance. We intend for miniF2F to be a community-driven effort and hope that our benchmark will help spur advances in neural theorem proving.

1 INTRODUCTION

Shared benchmarks and datasets have historically played a crucial role in driving advances in large-scale applications of deep learning, e.g. in computer vision (Deng et al., 2009) and natural language processing (Wang et al., 2019; Rajpurkar et al., 2016; Paperno et al., 2016). *Neural theorem proving* is a rapidly developing area which aims to apply techniques from deep learning to interactive theorem proving. To date, most contributions in this area have focused on individual theorem proving systems, each with a separately-implemented mathematics library and with results reported on a dataset-specific test split; examples include the HOList (Bansal et al., 2019a), CoqGym (Yang & Deng, 2019) and LeanStep (Han et al., 2021) theorem proving environments and benchmarks. However, benchmarks from this paradigm are not ideal for measuring the mathematical reasoning ability of neural theorem provers for several reasons. Library-specific train/test splits are siloed by construction, dependent on how theorems and lemmas are split in these libraries, and as such are not directly comparable across systems. Moreover, formal mathematics libraries are closer to software repositories than informal mathematical exposition, and many lemmas are implementation-specific artifacts without precise informal mathematical or cross-system translations.

To date, the neural theorem proving community has not organized its efforts around a cross-system benchmark. To address this need and to provide a common resource to research groups working on formal theorem proving, we present miniF2F, a unified cross-system benchmark of formal mathematics of progressively increasing difficulty, centering around Olympiad-level problem statements (AMC, AIME, IMO) as well as high-school and undergraduate maths classes. Both the content and name of miniF2F are inspired by the IMO Grand Challenge (Selsam et al., 2019): to build an AI that can win a gold medal in the International Mathematical Olympiad in a *formal-to-formal* (F2F) format. More precisely, the agent must receive IMO problems written in a formal mathematical format, and must produce a formal (i.e. machine-checkable) proof for that problem.

We intend for miniF2F to serve as a stepping stone for different formal systems towards the IMO Grand Challenge (Selsam et al., 2019), as it is end-to-end verifiable, cross-platform and spans a wide range of difficulty. While we report baseline results on miniF2F using GPT-f, a language model based on GPT-3 which has been finetuned for theorem proving, language models are not a mandatory approach for Olympiad problems and this assumption is not reflected in miniF2F, preserving the generality and widespread applicability of the benchmark to systems similar to DeepHOL (Bansal et al., 2019a) or Holophrasm (Whalen, 2016).

2 BACKGROUND AND RELATED WORK

BENCHMARKS

In the closely related field of (first-order) *automated theorem proving* (ATP), the TPTP (Sutcliffe, 2017) benchmark is a library of test problems in a unified format for ATP systems. In interactive theorem proving, the "Freek 100" (Wiedijk, 2008) tracks progress across various interactive theorem provers on a list of 100 mathematical theorems. Wu et al. (2020) built a simplified formal proof environment INT with an associated synthetic inequality benchmark. Competitions and communal challenges have also spurred development in formal theorem proving. The CADE ATP System Competition (CASC) (Sutcliffe, 2016) is a competition that evaluates the performance of first-order automated theorem proving systems. Proof Ground (Haslbeck et al., 2019), part of the ITP conference, is an interactive proving contest (for humans) that supports Coq, Isabelle, and Lean, which focuses on evaluating the formalization effort of proof to given problems within limited time. Finally, the IMO Grand Challenge (Selsam et al., 2019), a proposal from researchers working on the interactive proof assistant Lean, aims to build a system capable of solving IMO problems in the formal-to-formal format.

Due to its convenient framing as a natural language processing task, the domain of informal mathematical reasoning has received more attention than the formal one. MATH (Hendrycks et al., 2021) is a mathematics benchmark comprising 12,500 statements in natural language where exercises are classified into 5 levels of difficulty across various domains. Each exercise is combined with a detailed step-by-step proof in natural language. Scaling state-of-the-art models shows little amelioration on MATH, which requires advanced mathematical reasoning capabilities. miniF2F includes a number of formalized statements from MATH. NaturalProofs (Welleck et al., 2021) is another benchmark of natural proof in mathematics , containing 32k theorem statements and proofs. It essentially contains the proofs in ProofWiki and other resources. While MATH is more oriented towards mathematics exercises, NaturalProofs is focused on proofs of general mathematics theorems. Saxton et al. (2019) built a mathematics dataset with 2×10^6 training data and 10^4 test data, presented in a question-answering format where each statement is paired with a question written in natural language and a direct answer without proof.

NEURAL THEOREM PROVING

HOList (Bansal et al., 2019a;b; Paliwal et al., 2020) provides an environment as well as a benchmark for HOL Light. They also proposes various deep reinforcement learning approaches for theorem proving and report a pass rate of 59.91% on their benchmark. Yang & Deng (2019) built CoqGym, a large-scale dataset, which comes also with a learning environment, of 71k human-written proofs in Coq proof assistant. They report a 30.0% pass rate on the held-out test theorems in CoqGym. Polu & Sutskever (2020) applied a decoder-only transformer similar to GPT-3 (Brown et al., 2020) to proof steps prediction in Metamath combined with a log-probability based proof search. They also proposed a methodology to train a value function to further guide proof search, achieving a 56.22% pass rate on the held-out test set. Large language models were applied to Lean by Han et al. (2021). They created an environment around the Lean prover targeted to machine learning and propose a dataset extracted from low level proof artifacts that is shown to boost performance when used as a self-supervised co-training objective. They report a 48.4% pass rate on held-out test statements from `mathlib`, Lean's mathematical library (mathlib Community, 2020).

3 MINIF2F BENCHMARK

miniF2F is a dataset of manually formalized statements of Olympiad type problems, aligned in Lean, Metamath, and Isabelle (partial at the time of writing), providing a cross-platform benchmark for formal mathematical reasoning. Olympiad type problems are of particular interest to compare automated provers across different formal systems as the theories required to solve them are well identified and they generally do not require the definition of new mathematical concepts (a capability that remains beyond the current neural theorem proving state of the art).

The formalized statements in miniF2F are drawn from multiple sources, ranging from high school and undergraduate level exercises to Olympiad problems. miniF2F also covers different sub-

Table 1: Number of statements and their provenance in miniF2F `v1`

			Test Set	Validation Set
TOTAL			244	244
IMO			20	20
AIME			15	15
AMC			45	45
MATH	Algebra	Level 5	14	14
		Level 4	14	14
		Level 3	14	14
		Level 2	14	14
		Level 1	14	14
	Number Theory	Level 5	16	16
		Level 4	11	11
		Level 3	11	11
		Level 2	11	11
		Level 1	11	11
CUSTOM	Algebra		18	18
	Number Theory		8	8
	Induction		8	8

subjects in mathematics as well as proof strategies, focusing on the types of exercises whose statements are expressible in most formal systems. This leads to a systemic focus on algebra, number theory and inequalities because, for example, geometry and combinatorial problems are generally challenging to formalize due to only nascent efforts in these areas in most formal systems. The statements in miniF2F are all manually formalized and selected to cover a variety of difficulty levels for both humans and machines. Formal proofs for these statements are optionally attached.

miniF2F draws from AIME, AMC, IMO problems as well as problems from the MATH (Hendrycks et al., 2021) informal dataset. Formalizing problems from the MATH dataset serves two purposes. First, problems in MATH are segmented by difficulty level (from 1 to 5), randomly selecting a subset from each of these difficulty levels allows miniF2F to cover a wider range of difficulty. Second, it provides the community an opportunity to compare capabilities of formal automated prover to their informal counter-parts as discussed in later sections.

miniF2F comprises a test set and a validation set, which are a stratified random split from the statements we formalized such that each set equally covers each problem type and difficulty (when available). Table 1 shows a detailed distribution of these statements.

Versioning miniF2F is an evolving effort and new statements will continuously be added. Periodically, we will freeze versions of the benchmark. The current version of the benchmark is `v1`[1] and results in this paper are reported using this version. `v1` comprises 244 test and 244 valid statements. The set of statements of each version is guaranteed to remain stable, only allowing fixes in case errors are later discovered.

Rules of engagement and License miniF2F is meant to serve as a shared resource for research groups working on applying deep learning to formal theorem proving. There is no formal process to submit evaluation results and researchers are simply invited to cite miniF2F indicating the version used in their evaluations. We also encourage them to contribute proofs found by their approaches back to the benchmark. The parts of the benchmark associated with each theorem prover (Metamath, Lean, Isabelle) are meant to be licensed in a way that is aligned with the licensing usage associated with the theorem prover's main library. As a result, the Metamath version of the benchmark is released under the MIT License, while the Lean and Isabelle versions are released under the Apache License.

[1] `https://github.com/openai/miniF2F/tree/v1`

Porting effort In addition to Metamath, Lean and Isabelle (work in progress), we are eager to extend the coverage of miniF2F to HOL Light and Coq, and will welcome any effort in that direction or to extend miniF2F to further systems.

4 EXPERIMENTS

In this section, in order to study baseline performances associated with existing systems, we report pass rates achieved by GPT-f (Polu & Sutskever, 2020) applied to Metamath, GPT-f/PACT (Polu & Sutskever, 2020; Han et al., 2021) applied to Lean as well as a baseline prover implemented in Lean denoted as the `tidy` baseline. Pass rates are reported as Pass@N where N is the number of proof search attempts per statement. Pass@N is computed by running more attempts per statement, averaged to get an unbiased, low-variance estimate.

4.1 METAMATH

Metamath is powered by a meta logic system based on a single substitution rule. It's characterized by its simplicity which makes it convenient to study machine learning. Proofs in Metamath are, as a consequence of the low-level proofsteps, much longer than in other systems as there is no assistance from high-level tactics. Proofs which are trivial in other systems (e.g. n-digit addition or simple ring arithmetic transformations) can be quite tedious in Metamath. The absence of tactics is both a benefit, as the models sees and learns on everything, and a challenge, as proofs of even simple exercises require hundreds of proofsteps.

4.1.1 GPT-F

We report the pass rate of GPT-f applied to Metamath as described in Polu & Sutskever (2020). We use a model with 700m learnable parameters. The model is trained on an updated dump of the set.mm library (but similar synthetic datasets), using the log-probability based search as reported in Table 8 of the GPT-f paper (Polu & Sutskever, 2020).

The model achieves a Pass@1 of 1.3% and a Pass@8 of 1.6% on miniF2F-test. As expected, these numbers are quite low due to the length of typical proofs for even simple math exercises. The average proof length is also reported in Table 3.

4.2 LEAN

In comparison to Metamath, Lean benefits from a large number of powerful tactics to assist formalization efforts. Typical Lean proofs are much shorter than Metamath's. This is also a formal system of interest as it has received a lot of attention from the mathematical community as recent theories have successfully been formalized in Lean (Perfectoid Spaces (Buzzard et al., 2019), Liquid Tensor experiment (Scholze, 2020)).

Lean is also associated with the IMO Grand Challenge (Selsam et al., 2019) which aims to organize a formal-to-formal challenge during the upcoming IMO competitions.

4.2.1 TIDY BASELINE

We use the generic best-first search algorithm presented in PACT (Han et al., 2021). The algorithm works as follows: Given a list of tactics L with priority, we maintain a priority queue Q of tactic states whose priority is given by the priority of the last applied tactic in L that led to it. While Q is not empty, we pop the top tactic state t from Q. We iterate through L and apply each tactic to t. If no error is raised, we capture the returned tactic states from Lean and insert them back into Q.

We use the same terminology as in PACT (Han et al., 2021): maximum queue size ω_{max}, depth limit d_{max}. We also enforce a budget of i_{max} iterations of the outer loop. When Q's size reach q_{max}, all the tactic states to be inserted are discarded. We do not expand the next tactic state when the depth is beyond d_{max}. This loop is run until a proof is found or the iterations budget is exhausted.

For consistency checking, we run the `tidy` baseline under the same settings and on the same test set as in PACT (Han et al., 2021) except that we don't set a global timeout. Our implementation

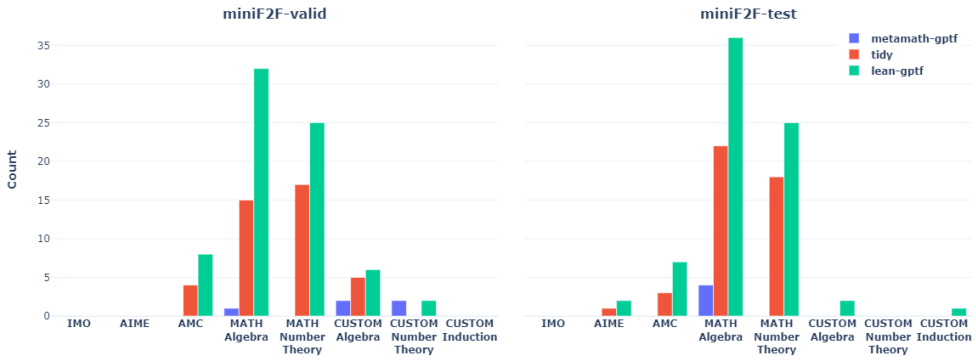

Figure 1: Counts of successfully proved statements in miniF2F. Green bar: results from Lean GPT-f. Red bar: best result from the `tidy` baseline. Blue bar: results from Metamath GPT-f.

Table 2: The table shows the number of solved statement in miniF2F when running the `tidy` baseline with different values of i_{max} as well Lean's built-in `tidy` tactic. All `tidy` baseline experiments are run with $\omega_{max} = 128$, $d_{max} = 8$ using $L + HL$. Despite the `tidy` baseline being deterministic, it is still subject to per-tactic application timeouts, explaining the number 43 reported on miniF2F-test for $i_{max} = 32$.

parameters	miniF2F-valid	miniF2F-test
Lean's tidy tactic	12 / 244	13 / 244
$i_{max} = 1$	21 / 244	23 / 244
$i_{max} = 2$	31 / 244	29 / 244
$i_{max} = 4$	38 / 244	41 / 244
$i_{max} = 8$	41 / 244	44 / 244
$i_{max} = 16$	41 / 244	44 / 244
$i_{max} = 32$	41 / 244	43 / 244
$i_{max} = 64$	41 / 244	44 / 244
$i_{max} = 128$	41 / 244	44 / 244

achieved a 10.5% pass rate on mathlib's test split. This result is comparable to the reported 9.9% in PACT given the waived global timeout.

In addition to the curated list of tactics L used in PACT (Han et al., 2021), we added 4 high-level tactics HL =`[nlinarith, linarith, ring_nf, norm_num]` to L with higher priorities than the others. We report our pass rate on miniF2F in Table 2.

4.2.2 GPT-F/PACT

We report the pass rate of GPT-f/PACT as described in Han et al. (2021). We use a model with 700M learnable parameters. The model is trained on an updated dump[2][3] of the mathlib library using the PACT methodology denoted in the paper as `mix2 > mix1 + tactic` in Figure 6.

The model achieves a Pass@1 of 24.6% and a Pass@8 of 29.2% on miniF2F-test. The average proof length is also reported in Table 3.

[2] https://github.com/jasonrute/lean_proof_recording/commit/8499f10c2e10dd533152070ed933c4f0b21ecdc0

[3] https://github.com/jesse-michael-han/lean-step-public/commit/a2b83c237bfe4d6f1c48bb48bc0769b5940e614a

Table 3: Baseline performance on Metamath and Lean. All proof searches are provided with a 128 expansions budget. GPT-f attempts $e = 16$ tactics per expansion while the `tidy` baseline attempts $e = 17$ tactics per expansion ($L + HL$, see section 4.2.1). Reported proof lengths are averages over all the proofs found in each run. Note that the `tidy` baseline being deterministic, there is no point attempting a proof search more than once.

		miniF2F-valid			miniF2F-test		
Formal System	Model	Proof Length	Pass rate Pass@1	Pass rate Pass@8	Proof Length	Pass rate Pass@1	Pass rate Pass@8
Metamath	GPT-f	16.2	1.0%	2.0%	20.3	1.3%	1.6%
Lean	tidy	1.7	16.8%	-	1.8	18.0%	-
Lean	GPT-f	2.6	23.9%	29.3%	2.5	24.6%	29.2%

4.3 Discussion

4.3.1 Access to high-level tactics

One goal of miniF2F is to study the comparison of performance across formal systems. In this section we reported the performance of the same methodology (GPT-f (Polu & Sutskever, 2020)) applied to both Lean and Metamath. Both models are pre-trained on WebMath (Polu & Sutskever, 2020) and respectively trained on datasets extracted from Lean (Han et al., 2021) and Metamath (Polu & Sutskever, 2020). The overall compute deployed at training is comparable in both setup and exactly equivalent at test-time, yet the achieved performance appears drastically superior when applied to Lean. We hypothesize that this is mainly explained by the model's access to high-level tactics when applied to Lean, enabling the model to learn how to guide Lean's automation in an effective way.

An example of this high-level guidance behavior is well exemplified by the following proof of the statement `algebra_sqineq_2unitcircatblt1` where the model heavily relies on Lean's `nlinarith` solver but provides it with essential premises to successfully guide the search.

```
theorem algebra_sqineq_2unitcircatblt1
  (a b : ℝ)
  (h₀ : a^2 + b^2 = 2) :
  a * b ≤ 1 :=
begin
  nlinarith [sq_nonneg a,sq_nonneg b,sq_nonneg (a - b)]
end
```

In Metamath, GPT-f fails to find a proof as it requires a very large number of steps to appropriately rewrite the goal in a way that is amenable to the use of set.mm's existing theorems. The `tidy` baseline also fails to find a proof of that statement as `nlinarith` is not capable of solving the goal without being passed extraneous premises.

These results motivate the use of neural theorem proving with formal systems that expose powerful high level tactics and also suggest the potential of a closer collaboration between formal systems and machine learning practitioners. It also motivates the use of generative models in that setup as the arguments required by high-level tactics to succeed on non trivial problems generally do not exist in the context of the statement and therefore have to be generated ex-nihilo.

4.3.2 Comparison of informal and formal setups

The use of formal systems for neural theorem proving is often motivated by the role of the formal system as a verifier, enabling more advanced neural search strategies than possible in a fully informal setup where the generation of a model can't be verified automatically, as well as the access to powerful tactics. Our formalization of a subset of the MATH (Hendrycks et al., 2021) informal dataset provides an interesting approximate quantification of the benefit of having access to a formal system in the context of neural theorem proving. Approximate, because we only formalized a small

subset of the MATH statements, but nonetheless useful since we drew uniformly from the 5 difficulty levels.

In Hendrycks et al. (2021), the performance of GPT-3 (which is a larger model than the GPT-f model studied here) is reported to be 6.0% in the algebra category and 3.9% in the number theory category. GPT-f applied to Lean by comparison achieves 51.4% in the algebra category and 41.7% in the number theory category. It is also worthwhile to note that the `tidy` baseline also highly outperforms (31.4% in algebra and 30.0% in number theory) GPT-3 in an informal setup demonstrating the benefit of proof automation alone.

5 Conclusion

We presented miniF2F, a dataset of formal Olympiad-level mathematics problem statements, meant to serve as an initial effort towards cross-system benchmarking of neural mathematical reasoning capabilities in formal environments. We reported the performance of the neural theorem prover GPT-f (Polu & Sutskever, 2020) on both the Lean and Metamath parts of miniF2F as well as the performance of our non-neural `tidy` baseline applied to Lean. Then, we discussed these baselines and put them in perspective with previously reported comparable results in informal environments (Hendrycks et al., 2021).

Finally, we hope that miniF2F will prove to be useful to the scientific community working on neural theorem proving and spur advances in this domain.

Acknowledgments

We are grateful to Wenda Li for contributing the Isabelle statements currently available in miniF2F, paving the way towards a full support of Isabelle, as well as his feedback and encouragement in the process.

References

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. Holist: An environment for machine learning of higher order logic theorem proving. In *International Conference on Machine Learning*, pp. 454–463. PMLR, 2019a.

Kshitij Bansal, Christian Szegedy, Markus N Rabe, Sarah M Loos, and Viktor Toman. Learning to reason in large theories without imitation. *arXiv preprint arXiv:1905.10501*, 2019b.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Kevin Buzzard, Johan Commelin, and Patrick Massot. Lean perfectoid spaces. `https://leanprover-community.github.io/lean-perfectoid-spaces/`, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. *arXiv preprint arXiv:2102.06203*, 2021.

Maximilian P. L. Haslbeck, Tobias Nipkow, and Simon Wimmer. Proof ground. `https://www21.in.tum.de/~wimmers/proofground/`, 2019.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

The mathlib Community. The lean mathematical library. In Jasmin Blanchette and Catalin Hritcu (eds.), *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs*

and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020, pp. 367–381. ACM, 2020. doi: 10.1145/3372885.3373824. URL https://doi.org/10.1145/3372885.3373824.

Aditya Paliwal, Sarah Loos, Markus Rabe, Kshitij Bansal, and Christian Szegedy. Graph representations for higher-order logic and theorem proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2967–2974, 2020.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1144. URL https://doi.org/10.18653/v1/p16-1144.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1264. URL https://doi.org/10.18653/v1/d16-1264.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*, 2019.

Peter Scholze. Liquid tensor experiment. https://xenaproject.wordpress.com/2020/12/05/liquid-tensor-experiment/, 2020.

Daniel Selsam, Kevin Buzzard, Reid Barton, Percey Liang, Sarah Loss, and Freek Wiedijk. Imo grand challenge. https://imo-grand-challenge.github.io/, 2019.

G. Sutcliffe. The CADE ATP System Competition - CASC. *AI Magazine*, 37(2):99–101, 2016.

G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning*, 59(4):483–502, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*, 2021.

Daniel Whalen. Holophrasm: a neural automated theorem prover for higher-order logic. *CoRR*, abs/1608.02644, 2016. URL http://arxiv.org/abs/1608.02644.

Freek Wiedijk. Formalizing 100 theorems. https://www.cs.ru.nl/~freek/100/, 2008.

Yuhuai Wu, Albert Qiaochu Jiang, Jimmy Ba, and Roger Grosse. Int: An inequality benchmark for evaluating generalization in theorem proving. *arXiv preprint arXiv:2007.02924*, 2020.

Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *International Conference on Machine Learning*, pp. 6984–6994. PMLR, 2019.

A EXAMPLE OF STATEMENT IN MINIF2F

Table 4: Problem 11 of 2000 AMC 12 is formalized with proof in different languages in miniF2F. The proof in Metamath is too long to be fully displayed.

<table>
<tr><td>Natural Language</td><td>Two non-zero real numbers, a and b, satisfy $ab = a - b$. Which of the following is a possible value of $\frac{a}{b} + \frac{b}{a} - ab$? (A) -2 (B) $\frac{-1}{2}$ (C) $\frac{1}{3}$ (D) $\frac{1}{2}$ (E) 2</td></tr>
<tr><td>Metamath</td><td><pre>${
amc12-2000-p11.0 $e |- (ph -> A e. RR) $.
amc12-2000-p11.1 $e |- (ph -> B e. RR) $.
amc12-2000-p11.2 $e |- (ph -> A =/= 0) $.
amc12-2000-p11.3 $e |- (ph -> B =/= 0) $.
amc12-2000-p11.4 $e |- (ph -> (A x. B) =
  (A - B)) $.
amc12-2000-p11 $p |- (ph -> (((A / B) +
  (B / A)) - (A x. B)) = 2)
$=
(cdiv co caddc cmul cmin c2 cexp eqcomd ... $.
$}</pre></td></tr>
<tr><td>Lean</td><td><pre>theorem amc12_2000_p11
  (a b : ℝ)
  (h₀ : a ≠ 0 ∧ b ≠ 0)
  (h₁ : a * b = a - b) :
  a / b + b / a - a * b = 2 :=
begin
  field_simp [h₀.1, h₀.2],
  simp only [h₁, mul_comm, mul_sub],
  ring,
end</pre></td></tr>
<tr><td>Isabelle</td><td><pre>theorem amc12_2000_p11:
  fixes a b::real
  assumes "a \<noteq> 0" "b \<noteq> 0"
    and "a * b = a - b"
    shows "a / b + b / a - a * b = 2"
  using assms
  by (smt (verit, ccfv_threshold)
    diff_divide_distrib
    div_self divide_divide_times_eq
    eq_divide_imp nonzero_mult_div_cancel_left)
end</pre></td></tr>
</table>