miniF2F Revisited: Reviewing Limitations and Charting a Path Forward

Anonymous Author(s)
Affiliation
Address
email

Abstract

We perform a thorough analysis of the formal and informal statements in the miniF2F benchmark from the perspective of an AI system that is tasked to participate in a math Olympiad consisting of the problems in miniF2F. In such setting, the model has to read and comprehend the problems in natural language, formalize them in Lean language, then proceed with proving the problems, and it will get credit for each problem if the formal proof corresponds to the original informal statement presented to the model. Our evaluation results reveal that the best accuracy of such pipeline can be about 36% using the SoTA models in the literature, considerably lower than the individual SoTA accuracies, 97% and 69% reported in the autoformalization and theorem proving literature. Analyzing the failure modes, we trace back a considerable portion of this drop to discrepancies between the formal and informal statements for more than half of the problems in miniF2F. We proceed with correcting all the errors, discrepancies and simplifications in formal and informal statements, and present the *miniF2F-v2* with fully verified formal and informal statements and proofs. Evaluating the full theorem proving pipeline on *miniF2F-v2* leads to the best accuracy of 57%, a significant improvement from the 36% on miniF2F-v1, yet indicating considerable misalignment between the autoformalization models and theorem provers. Our deep analysis suggests that a higher quality benchmark can help the community better evaluate progress in the field of formal reasoning and also better diagnose the failure and success modes of autoformalization and theorem proving models. We release our dataset, *miniF2F-v2*, and all the corresponding formal proofs to the public.

1 Introduction

Automated reasoning with computers has a long and rich history [1], and with the rise of AI, it has had major advancements in the past decades, notably DeepBlue [2], AlphaGo [3], etc. Shortly after the rise of Large Language Models (LLMs), [4] showed the remarkable ability of these models to learn the language of formal verification systems such as Lean [5] and automatically prove mathematical theorems in formal language. This gave rise to the subfield of Automated Theorem Proving (ATP) in the machine learning literature which has seen considerable advancements in the past few years [6]. The shared progress in this field was partly made possible by the miniF2F benchmark [7] which consists of 488 theorems in formal and informal languages drawn from prestigious mathematical competitions and Olympiads.

Writing mathematical proofs in formal language makes the verification of proofs automated and reliable, however, learning and writing the language of formal verification systems is not easy for humans nor for the LLMs. For a human, it might take 10 times longer to write a proof in formal language compared to an informal one. LLMs, on the other hand, are likely to excel at learning these

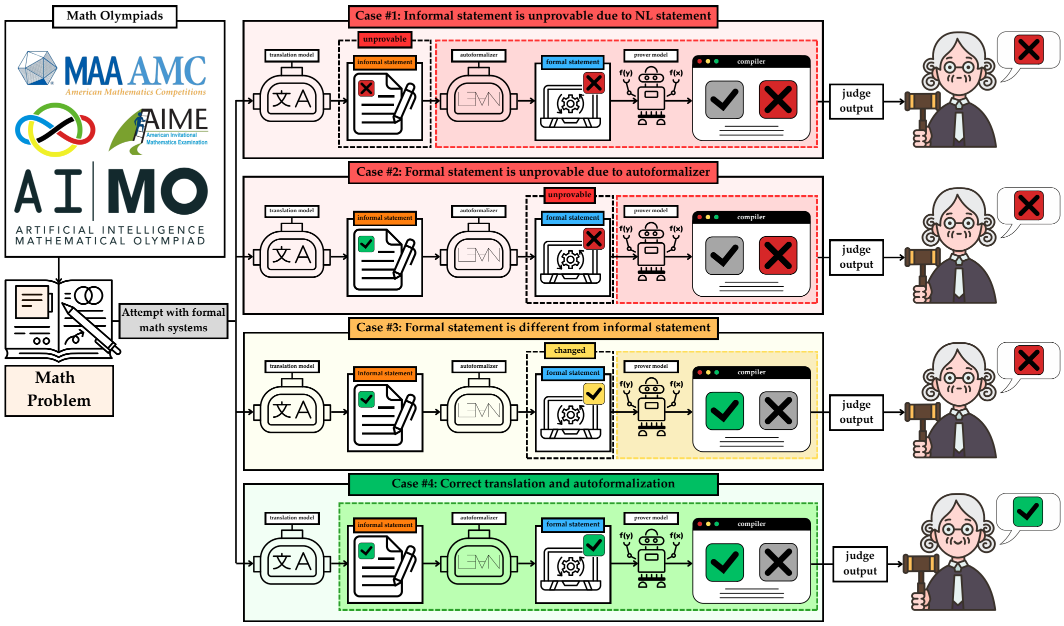

Figure 1: High-level overview of a formal math prover system operating in a Math Olympiad setting.

formal languages, and combined with their reasoning abilities and possibly other tools, they may help automate the process of mathematical reasoning. This has given rise to one of the main goals of the community: developing AI systems that can compete with humans in major mathematical Olympiads [8]. For example, in 2024, AlphaProof was able to reach the level of silver medal at the International Math Olympiad (IMO) [9]. Reaching that level of automated reasoning requires a model to automatically read a mathematical problem in informal language, formalize it in a formal verification language, and generate a correct formal proof that can pass the verification system. This is exactly what AlphaProof did using Lean [5] for IMO 2024 as verified by the organizers of IMO.

Such automated system would fail if it changes the original problem statement and proves an altered version of the competition problem, the same way that if a human participant proves a simplified version of a problem will likely get no, or at the most, a partial credit not sufficient for a medal. Therefore, having models that can correctly translate to formal language is crucial, otherwise, there will be a need for a human in the loop to perform the formalizations. This translation, known as autoformalization in the literature, has seen major advancements on the miniF2F benchmark [10].

While the miniF2F benchmark has enabled advancements both in ATP and in autoformalization, over the years, it has also been reported to have certain limitations, such as wrong formalizations, unprovable theorems, etc. Most recently, [11] reported fixing errors in 5 theorems of miniF2F which had made them unprovable. The community on autoformalization has also reported certain inconsistencies between the formal and informal statements in miniF2F [12]. This motivates us to make a deep analysis of this benchmark from the holistic perspective described above where an AI system is tasked to participate in a miniF2F competition proving all the 488 theorems in the benchmark starting from the informal statements aiming to provide a correct proof for the original statements. Figure 1 illustrates the stages at which failures can prevent an AI system from arriving at a correct solution in a Math Olympiad setting:

1. **Translation failure:** Errors during plain-text parsing introduce unprovable statements on the natural language level.

2. **Autoformalization failure:** The informal math statement is mistranslated into the formal statement, making the problem unprovable.

3. **Semantic drift:** Subtle changes during autoformalization alter the goal so it no longer matches the original problem, therefore the proof does not correspond to the original problem.

4. **Success:** Correct translation, accurate autoformalization, and a valid proof that also passes human verification.

These cases highlight the need for tight collaboration between autoformalizers and theorem provers: to build truly reliable systems, we must ensure each transformation, from plain text to informal statement to formal proof, preserves the original goal and yields verifiable results.

In this work, we build such an automated pipeline by using the SoTA models for autoformalization and for theorem proving. While the accuracy of the best autoformalization model on miniF2F is reported to be about 97% [13], and the accuracy of Kimina Prover on miniF2F is 70.8% [11], the combined accuracy of these models leads to an accuracy of 34.8% after comparing the final proofs with the original problem statements in informal language. This significant drop in accuracy comes from several sources which we will examine in detail, two of which account for most of the failures. Our extensive human evaluation of autoformalization results, across three models, reveals that their autoformalization accuracy is not as high as the ones reported in the literature, because those reported accuracies are often evaluated by LLMs and not by a human familiar with the formal language. The other reason for this major drop is that the formal statements in miniF2F, i.e., the starting point of the automated system, are often significantly simplified compared to the informal statements, and hence, when more faithful translations are given to ATPs, the theorems turn out to be more difficult, making the ATPs more likely to fail. Therefore, we observe a disconnect between the ATP literature and the autoformalization literature.

We analyze every failure mode of an end-to-end formal reasoning pipeline on the miniF2F benchmark and correct over 300 Lean 4 statements to eliminate errors and simplifications. By resolving these discrepancies, we pave the way toward unifying formal and informal mathematics.

Our contributions are as follows:

- We release a corrected version of miniF2F benchmark (both test and validation sets) along with all the formal proofs. We further provide human-verified informal proofs for all the problems that are translated back from the formal proofs. All the informal and formal statements are manually checked to exactly correspond to each other.
- We perform a thorough evaluation of autoformalization models on miniF2F v1 and v2, demonstrating the current shortcomings in the evaluation practices in the autoformalization literature as well as the benefits of miniF2F-v2.
- We perform a thorough evaluation of miniF2F-v2 for the task of ATP reporting that the accuracy of SoTA models drop significantly on the problems that were excessively simplified, yet their accuracy increases on the subset of problems that were previously unprovable because of formalization errors.
- We further evaluate a complete automated pipeline of SoTA models on the task of theorem proving starting from informal statements and report their accuracy both on the original version of miniF2F and miniF2F-v2 demonstrating the better accuracy of such pipelines on miniF2F-v2.

2 Reviewing the miniF2F in detail

In this section, we detail various types of changes that we made in the original miniF2F benchmark for both formal and informal statements. Detailed information about the distribution of made changes can be found in the Appendix.

2.1 Errors in informal statements

Unprovable problems. These set of problems miss critical hypotheses to prove the goal; therefore, making them impossible to prove.

Incomplete statements. There are problems that do not provide enough information about the problem, or do not mention the type of the variables. There are also instances where the informal statement differs from the original problem statement presented in the competitions such as IMO. In all such cases, we use the original informal statements. We refer to Figure 2 to illustrate how we modify the original benchmark. In the revised version on the right, we add all necessary assumptions, expand both the informal and formal statements and make sure that stated goal, i.e. "Show that it [x] is 72", is reflected in the formal statement correctly.

Wrong given solution. Another subset of incorrect informal statements provide a wrong or inconsistent solution that deviates from the original problem statement and solution.

Correction of mathd_algebra_31

miniF2F-v1

If $\sqrt{x+\sqrt{x+\sqrt{x+\sqrt{x+\cdots}}}}=9$, find x. Show that it is 72.

```
/--Formal statement--/
theorem mathd_algebra_31_minif2f_v1
  (x : NNReal)
  (u : ℕ → NNReal)
  (h₀ : ∀ n, u (n + 1) = NNReal.sqrt (x + u n))
  (h₁ : Filter.Tendsto u Filter.atTop (nhds 9)) :
  9 = NNReal.sqrt (x + 9) := by
  sorry
```

miniF2F-v2

Consider a nonnegative real number x, and a recursive function u from natural numbers to real numbers. For all natural numbers, we have $u(n+1) = \sqrt{x+u(n)}$. Assume that the as n reaches infinity in its limit, u(n) goes to 9. What is the value of x? Show that it is 72.

```
/--Formal statement--/
theorem mathd_algebra_31_minif2f_v2
  (x : ℝ) (hx : 0 ≤ x)
  (u : ℕ → ℝ)
  (h₀ : ∀ n, u (n + 1) = Real.sqrt (x + u n))
  (h₁ : Filter.Tendsto u Filter.atTop (nhds 9)) :
  x = 72 := by
  sorry
```

Figure 2: Correction example of mathd_algebra_31 problem in miniF2F.

No given solution. Some of the informal statements do not provide a solution after the problem description. We add the solution to such statements.

Multiple choices in the statement. We filtered the informal statements from multiple choice style answers to reflect only the final goal to be proved.

There were also instances where the informal statements were incomplete or they included irrelevant information such as a sentence talking about the location of the competition. We corrected and purged such cases. Additional examples of corrected statements are in the Appendix.

2.2 Errors in formal statements

Excessively simplifying the problem by changing the goals or changing the conditions. This subset of problems aim to prove simpler goals compared to the original informal description. These problems do not reflect the original difficulty of the problem; therefore, making the LLM's task easier. We segregate these problems into two categories: *simplified* and *excessively simplified*. The common culprits behind simplification are omitting part of the informal statement, simplifying the goals, adding helpful assumptions, wrong type declarations (e.g., proving a goal for non-negative real numbers instead of all real numbers, or using equivalences instead of functions). Excessive simplification cases are those where the goal is significantly altered.

Not using the correct functions/expressions in Lean. Another subset of formal errors are incorrect declaration of functions and expressions. In some cases, informal statements describe conditions, functions, etc, whereas the formalized version does not declare them. We fix these inconsistencies in our revised version of miniF2F.

Unprovable statements. We identified a total of 16 problems across test and validation sets that do not have a solution in their current form, which is a critical factor when it comes to reliable evaluation of theorem provers. The errors come from improper translation of the informal statement to a formal counterpart, such as missing brackets or using a wrong function from the Mathlib library.

3 Evaluation of complete formal reasoning pipelines starting from informal statements

This section presents our experiments with an end-to-end pipeline that aims to prove informal theorems with the aid of formal theorem prover systems. The setting is motivated by Math Olympiad–style problems, where the task is to produce a correct solution for a given informal statement. We selected every original problem used to construct the *miniF2F* benchmark and employed them to evaluate how well state-of-the-art autoformalizers collaborate with modern theorem provers.

We begin with two datasets, *miniF2F-v1* and *miniF2F-v2*, which contain the original problems rewritten in the "informal prefix" style expected by autoformalizers. Version 2 was revised to adhere more closely to the wording found in textbooks and contest statements, so its informalizations better reflect the source material.

Table 1: Comparison of effective accuracy across different autoformalizers and theorem provers. Effective accuracy refers to a final accuracy after collaboration between an autoformalizer model and a theorem prover. *Formal* verification setting refers to cases where Lean compiler marks the proof as correct. *Match with original problem* refers to cases when final solved solution is the same as the original problem statement sourced from textbooks/competitions.

	miniF2F		Verification Setting		Theorem prover accuracy (%)		
	ver.	split	Formal	match w. original problem	Deepseek-Prover-V1.5-RL	Goedel-Prover-SFT	Kimina-Prover-Distill-7B
Herald translator	v1	test	✓	✗	32.8	36.1	40.6
	v2	test	✓	✗	28.7	33.2	39.8
	v1	test	✓	✓	20.9	23.0	26.6
	v2	test	✓	✓	25.0 (4.1 $\uparrow$)	29.5 (6.5 $\uparrow$)	**34.8** (**8.2** $\uparrow$)
	v1	valid	✓	✗	48.4	52.0	53.3
	v2	valid	✓	✗	36.5	37.7	43.0
	v1	valid	✓	✓	30.3	33.6	34.4
	v2	valid	✓	✓	32.4 (2.1 $\uparrow$)	34.4 (0.8 $\uparrow$)	**37.7** (**3.3** $\uparrow$)
Kimina autoform.	v1	test	✓	✗	46.7	56.6	62.7
	v2	test	✓	✗	38.9	43.9	55.3
	v1	test	✓	✓	27.0	31.1	36.1
	v2	test	✓	✓	36.5 (9.5 $\uparrow$)	40.6 (9.5 $\uparrow$)	**51.2** (**15.1** $\uparrow$)
	v1	valid	✓	✗	61.5	66.0	69.7
	v2	valid	✓	✗	52.0	56.1	63.1
	v1	valid	✓	✓	38.5	42.6	44.3
	v2	valid	✓	✓	46.7 (8.2 $\uparrow$)	51.2 (8.6 $\uparrow$)	**57.0** (**12.7** $\uparrow$)
o4-mini	v1	test	✓	✗	29.5	34.8	39.3
	v2	test	✓	✗	29.1	33.6	40.2
	v1	test	✓	✓	18.0	20.5	24.2
	v2	test	✓	✓	25.0 (7.0 $\uparrow$)	29.1 (8.6 $\uparrow$)	**33.6** (**9.4** $\uparrow$)
	v1	valid	✓	✗	41.8	44.7	45.9
	v2	valid	✓	✗	39.8	42.6	45.1
	v1	valid	✓	✓	27.0	29.5	30.3
	v2	valid	✓	✓	36.9 (9.9 $\uparrow$)	40.2 (10.7 $\uparrow$)	**41.4** (**11.1** $\uparrow$)

For each problem we begin by feeding the informal statement to an autoformalizer; we keep the first formal output that both passes REPL verification and remains semantically faithful to the source. We then attempt to prove the resulting goal with several theorem provers, and finally we compare the derived theorem with the original problem, recording any discrepancies. We refer to final accuracy of autoformalizer and theorem prover collaboration as "effective accuracy".

Table 1 summarizes our end-to-end results on miniF2F-v1 and miniF2F-v2. For each subset of problems we report two effective accuracy metrics: (i) the proportion of formalizations that pass REPL verification, and (ii) the proportion that both pass verification *and* align with the original statement, indicated by a ✓ in the *match with original problem* column.

We observe that across all models, the effective accuracy is considerably lower compared to accuracies collected from autoformalizers and theorem provers separately. Moreover, we observe that after semantic verification with respect to the source problems, effective accuracy drops across all autoformalizers and theorem provers, which suggests that incorporating full pipeline leads to numerous mistakes during proof generation process.

The more faithful translations in miniF2F-v2 allow the theorem provers to obtain consistently higher accuracies under the full informal-to-formal pipeline. Its closer adherence to the source problems makes it a better reflection of true capabilities of modern autoformalizers and theorem provers.

Table 2: Comparison of autoformalization accuracy of Herald translator at @128 between LLM and human evaluators. Back translation and LLM equivalence check pipeline is adopted from [13].

Evaluator	miniF2F-v1		miniF2F-v2	
	test	validation	test	validation
LLM	97.5%	97.1%	96.7%	97.5%
Human	62.7% (-34.8%)	69.7% (-27.4%)	66.0% (-30.7%)	68.9% (-28.6%)

Table 3: Comparison of autoformalization accuracy of Herald translator, Kimina autoformalizer and o4-mini on the original version of miniF2F (miniF2F-v1) and miniF2F-v2. The o4-mini translation has sample budget up to @10, but the generation stops at the first correctly compiled attempt, while other models are evaluated with @1.

miniF2F		verified by	Model Formalization Accuracy (%)		
ver.	split		Herald translator	Kimina autoformalizer	o4 mini*
v1	test	LLM	49.6%	58.6%	-
v1	test	human	55.3%	88.1%	46.3%
v2	test	LLM	49.2%	54.5%	-
v2	test	human	51.6%	79.5%	51.6%
v1	valid	LLM	51.2%	57.8%	-
v1	valid	human	59.4%	82.0%	47.1%
v2	valid	LLM	50.4%	51.6%	-
v2	valid	human	48.0%	78.7%	52.5%

4 Evaluation of autoformalization models

This section evaluates the performance of autoformalizers on both miniF2F-v1 and v2. The most notable observation of this section is that the reported accuracy of autoformalization models in the literature are largely inflated as those evaluations are typically performed by LLMs. For example, when we perform a human review of the outputs of Herald @128 that LLM has marked as 97% correct, we arrive at a much lower accuracy of 66%.

We consider two specialized autoformalization systems, Herald translator [13] and Kimina autoformalizer [11], and one general-purpose model, o4-mini [14]. All experiments use a sampling budget of @1 or @128 for the dedicated autoformalizer models and @10 (with intermediate compiler feedback) for o4-mini. We note that although o4-mini has a different sample budget, we stop at the first successful compilation attempt, which puts it on par with @1 Herald translator and Kimina autoformalizer models. We exclude any autoformalization outputs that fail to pass the Lean compiler to ensure syntax correctness. Lean compiler of choice is Lean REPL [15].

To evaluate Herald translator, Gao et al. [13] used InternLM2-Math-Plus-7B [16] for back-translation and DeepSeek Chat v2.5 [17] for equivalence verification. As observed by Ye et al. [18], using an LLM as a judge reduces verification overhead but can diverge from human judgments. In our experiments, we did not change back-translation model; however we used Deepseek-V3 [19], instead of Deepseek-V2.5 for natural language validation. To present accurate results aligned with the human grasp of presented problems, we manually verified every translation and report both LLM-based and human-verified accuracies. All human verifications were conducted by Lean experts.

Table 2 compares the accuracy of the LLM evaluator with human verification on both versions of miniF2F, using the Herald translator with a sampling budget of @128. The LLM tends to produce false positives and therefore reports a much higher accuracy than a human evaluator. In many cases it treats small discrepancies between statements as negligible even though they significantly affect the meaning of the statements; as shown in Section 3, these differences accumulate and reduce the practicality of full informal-to-formal pipelines. Consequently, autoformalizations should be evaluated with great care, and semantic alignment must remain strict.

Table 4: Comparison of whole-proof generation models' accuracy on the original version of miniF2F and miniF2F-v2.

Dataset	Deepseek-Prover-V1.5-RL	Goedel-Prover-SFT	Kimina-Prover-Distill-7B
miniF2f-v1-test	50.0%	58.2%	65.2%
miniF2f-v2-test	41.0%	48.4%	59.0%
miniF2f-v1-valid	63.9%	68.9%	73.0%
miniF2f-v2-valid	55.3%	59.8%	68.0%

Table 5: Comparison of whole-proof generation models' accuracy on the subset of problems in miniF2F-v2 that were previously unprovable on the original version of miniF2F.

Unprovable subset	Deepseek-Prover-V1.5-RL	Goedel-Prover-SFT	Kimina-Prover-Distill-7B
miniF2f-v2-test (out of 13)	4 (30.8%)	4 (30.8%)	5 (38.5%)
miniF2f-v2-valid (out of 3)	1 (33.3%)	1 (33.3%)	1 (33.3%)

To broaden our study, we conducted @1 experiments with the Herald translator, the Kimina autoformalizer, and o4-mini. Here the opposite pattern appears: the LLM evaluator assigns lower accuracies than the human evaluator. We hypothesize that with a larger sampling budget the LLM has a higher chance of hallucinating and assigning incorrect labels, whereas at @1 it behaves more conservatively. Moreover, while human evaluation at @128 shows only a 10–15% improvement over @1, the LLM evaluation suggests almost double the gains.

Kimina Autoformalizer attains higher accuracy than Herald Translator under both LLM-based and human verification, with accuracies ranging from 78% to 88% across both versions of miniF2F. However, when evaluating on miniF2F-v1 against miniF2F-v2, both Herald Translator and Kimina Autoformalizer exhibit a performance decline, whereas o4-mini improves on the corrected dataset. This finding suggests that current autoformalizers may suffer from data contamination.

5 Evaluation of theorem provers on formal statements

In this section we conduct a series of experiments only with whole-proof generation LLMs starting from the formal statements in the miniF2F-v1 and v2. This is to provide insights about the differences in the formal statements of the two versions of miniF2F while ablating the effect of autoformalizers. Following the literature, we use a sampling budget of @32 in all runs. Deepseek-Prover-V1.5-RL and Goedel-Prover-SFT are evaluated under `Lean v4.9.0`, matching the versions used in their original papers, whereas Kimina-Prover-Distill-7B is tested with the newer `Lean v4.17.0`. All experiments were performed on eight NVIDIA A5000 GPUs with 128 CPU cores.

Performance of theorem provers on miniF2F-v2. Table 4 reports the accuracy of selected theorem provers on miniF2F-v1 and miniF2F-v2. Notably, the accuracy of every theorem prover is lower on miniF2F-v2, since many simplifications made in miniF2F were reverted back, and theorems became more challenging. The proposed dataset poses a greater challenge to state-of-the-art LLMs.

Performance of theorem provers on the modified problems in miniF2F-v2. Although the overall accuracy declines on miniF2F-v2, it is important to note that this version corrects sixteen statements

Table 6: Comparison of whole-proof generation models' accuracy on the subset of problems in miniF2F-v2 that were *simplified* in the original version of miniF2F.

Simplified subset	Deepseek-Prover-V1.5-RL	Goedel-Prover-SFT	Kimina-Prover-Distill-7B
miniF2F-v1-test (out of 40)	29 (72.5%)	30 (75.0%)	31 (77.5%)
miniF2f-v2-test (out of 40)	23 (57.5%)	24 (60.0%)	25 (62.5%)
miniF2F-v1-valid (out of 48)	27 (56.2%)	29 (60.4%)	30 (62.5%)
miniF2f-v2-valid (out of 48)	25 (52.1%)	25 (52.1%)	27 (56.2%)

Table 7: Comparison of whole-proof generation models' accuracy on the subset of problems in miniF2F-v2 that were *excessively simplified* in the original version of miniF2F.

Excessively Simplified subset	Deepseek-Prover-V1.5-RL	Goedel-Prover-SFT	Kimina-Prover-Distill-7B
miniF2F-v1-test (out of 45)	21 (46.7%)	33 (73.3%)	33 (73.3%)
miniF2f-v2-test (out of 45)	10 (22.2%)	13 (28.9%)	24 (53.3%)
miniF2F-v1-valid (out of 36)	26 (72.2%)	26 (72.2%)	28 (77.8%)
miniF2f-v2-valid (out of 36)	7 (19.4%)	9 (25.0%)	17 (47.2%)

that were unprovable in the original benchmark. All theorem provers can now solve a subset of these repaired problems, which raises their accuracy on this specific group of tasks. The results are reported in Table 5. Since we provide the formal proofs for all problems in miniF2F-v2, one can be sure that all theorems are provable.

To further assess the impact of our revisions, we compare prover accuracy on the previously defined *simplified* and *excessively simplified* subsets. Table 6 shows that theorems in the *simplified* group pose only a modest challenge: accuracy falls on exactly six problems for every model, yielding a 15 % drop on the test set, while the validation set experiences an even smaller decline. The picture changes significantly for the *excessively simplified* subset. Here every model struggles: test-set accuracy decreases by more than 20%, and the validation set loses over 30%. Deepseek-Prover-V1.5-RL and Goedel-Prover-SFT suffer the largest losses, in some cases up to 40–50%. In contrast, Kimina-Prover-Distill-7B remains comparatively robust, proving 53.3% of the test problems and 47.2% of the validation problems, indicating strong generalization. By making the subset of problems closer to the intended difficulty, we see that each LLM struggles at least with some of the new theorems. This indicates the importance of the renewed version of miniF2F with scaled difficulty.

6 Related Works

Autonomous AI systems excelling in reasoning and scientific discovery. With the rise of generative models, we have seen their success in scientific discoveries [20]. For example, FunSearch [21] succeeded in writing a bin-packing algorithm that is faster than any human written algorithm. These systems, usually consisting a LLM at their core, have shown remarkable ability in automated reasoning. For example, AlphaGeometry [22] was able to reach gold-medal level in solving geometry problems at IMO. AlphaProof [9], similarly reached silver-medal level in proving IMO problems in number theory and algebra. Other examples include AlphaCode [23] and AlphaEvolve [24].

Silver and Sutton [25] suggest that we will see a new generation of AI agents that will reach unprecedented abilities predominantly by learning from experience. This argument largely draws not just from recent successes of AI systems, but also looks back at the successes of systems such as AlphaGo which was able to learn the game of Go merely by playing with an automated adversary, and ultimately reaching the level of expertise to beat the human champion. When the same algorithm was transformed to play the Japanese chess, it also passed the best human performance while the model developers had no familiarity with the Japanese chess. Indeed, the strategies that model used were known to fail by human champions, yet the models devised them in ways that were able to beat those same champions.

The idea here is that to make new discoveries or to arrive at models that can find better ways of playing a game or can excel at proving mathematical theorems, the models have to be given the space to explore the possibilities by themselves with no or minimal human intervention. From this perspective, a fully automated pipeline for mathematical reasoning would be preferable to a pipeline that needs a human in the loop for formalizing the statements, verifying the correctness of statements, etc. Hence, in this work, we suggest a fully automated pipeline to evaluate AI systems on the task of formal reasoning.

Automated Theorem Proving. Automatically proving mathematical theorems has a rich history including SAT and SMT solvers. In recent years, LLMs have shown a remarkable capability to generate formal proofs by themselves [26, 27, 26, 28, 29, 30, 11] or with help from other automated systems such as retrieval-based and/or search methods [31, 30, 32, 33, 34, 9, 35]. Before the release of Kimina Prover, SoTA LLMs on the task of theorem proving were only able to prove theorems with relatively short proofs in formal language, heavily relying on automated solvers in Lean such as nlinarith. The longest proof written by Goedel Prover on miniF2F consisted of ~ 10 lines. Kimina Prover, however, increased this limit using a longer context length generating proofs as long as a few hundred lines.

Autoformalization. This can be viewed as a translation task [36, 37] where statements from informal language are translated to the language of a formal verification system such as Lean. Informalization is the reverse translation from formal language to an informal one which is considered an easier task. LLMs have shown a remarkable ability in translation tasks especially when large corpus of text are available in two or more languages. Similarly LLMs are good at writing code in languages such as Python and C++ [38]. We also have seen gradual improvements in the accuracy of LLMs in autoformalization [10, 30].

Herald [13], the current state-of-the-art in the literature, reports an accuracy of 97% on the miniF2F while its accuracy is measured by an LLM and not verified by a human familiar with Lean. There are generally two difficulties in the field of autoformalization. First, high quality data paired in formal and informal languages is scarce. Second, there are no automated systems that can reliably verify the correctness of a translation [39], and as we will see, LLMs may not be reliable in evaluating whether a translation is correct. Even when the ground truth is available in formal language, it may not be easy, for a human nor a LLM, to evaluate whether a freshly generated formal statement is equivalent to the ground truth. There has been considerable work in this domain trying to automate the evaluation of equivalent formal statements [40, 41].

Benchmarks for formal reasoning. Over the past few years, the community has introduced several formal-mathematics benchmarks of varying difficulty. ProofNet [42] focuses on undergraduate-level mathematics, while PutnamBench [43] collects problems from the William Lowell Putnam Competition (1962–2023). NuminaMath [44] and miniCTX [45] target advanced theorems drawn from Lean projects and textbooks. Recently proposed, Con-NF [39] benchmark is specifically designed for an autoformalization task. In this work, we concentrate on the miniF2F dataset [46], which consists of 488 problems sourced from textbooks and competitions such as IMO, AIME, and AMC.

Reliability of mathematical benchmarks. Ensuring the reliability of LLM benchmarks is a critical concern for the research community. To accurately assess model capabilities, benchmarks must be both comprehensive and error-free. Vendrow et al. [47] evaluated numerous datasets across various tasks and introduced the concept of *platinum* benchmarks, i.e. those containing minimal errors and verified by human experts. The integrity of these benchmarks is essential for measuring progress in formal theorem proving. Accordingly, we dedicate our efforts to exhaustively verifying the entire miniF2F dataset.

7 Limitations

This work only covers Lean language, even though miniF2F is available for other languages too. Our corrections to the informal statements are applicable to all users of miniF2F.

8 Conclusion

We introduced *miniF2F-v2*, a revised version of the miniF2F benchmark. Hundreds of theorems were re-verified and changed to match the difficulty of their source problems, and the sixteen unprovable statements in the original dataset were fixed. To recreate a realistic setting of math Olympiad competitions, using the SoTA models in the literature, we built a completely automated pipeline of theorem proving starting from natural language statements. Our evaluation results show that in such setting the accuracy of current models will significantly drop on miniF2F-v1. However, when we do the same evaluation on miniF2F-v2, some of the lost accuracy is gained back because of the higher quality of the revised dataset. We further compared LLM evaluation of the outputs of autoformalization models with expert human verification and observed a substantial gap: LLMs marked many formalizations as correct even though they differed from the intended statements. By our evaluation, the accuracy of SoTA autoformalization model on miniF2F-v1 is 66%, not the reported 97%. We hope that *miniF2F-v2* will serve as a clearer and more demanding benchmark and will guide future progress in both autoformalization and formal theorem proving.

References

[1] Alan M Turing. Intelligent machinery, a heretical theory. *Philosophia Mathematica*, 4(3), 1948.

[2] Bruce Pandolfini. *Kasparov and Deep Blue: The historic chess match between man and machine*. Simon and Schuster, 1997.

[3] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

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

[5] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*, page 625–635, Berlin, Heidelberg, 2021. Springer-Verlag.

[6] Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. Formal mathematical reasoning: A new frontier in AI. In *Proceedings of the International Conference on Machine Learning*, 2025.

[7] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. miniF2F: A cross-system benchmark for formal Olympiad-level mathematics. In *International Conference on Learning Representations*, 2021.

[8] The AIMO Prize: Airtificial Intelligence Mathematical Olympiad. https://aimoprize.com/.

[9] Google DeepMind. AI achieves silver-medal standard solving international mathematical olympiad problems. `https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/`, 2024. Accessed: 2025-05-08.

[10] Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. *Advances in Neural Information Processing Systems*, 35, 2022.

[11] Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, Jianqiao Lu, Hugues de Saxcé, Bolton Bailey, Chendong Song, Chenjun Xiao, Dehao Zhang, Ebony Zhang, Frederick Pu, Han Zhu, Jiawei Liu, Jonas Bayer, Julien Michel, Longhui Yu, Léo Dreyfus-Schmidt, Lewis Tunstall, Luigi Pagani, Moreira Machado, Pauline Bourigault, Ran Wang, Stanislas Polu, Thibaut Barroyer, Wen-Ding Li, Yazhe Niu, Yann Fleureau, Yangyang Hu, Zhouliang Yu, Zihan Wang, Zhilin Yang, Zhengying Liu, and Jia Li. Kimina-Prover Preview: Towards large formal reasoning models with reinforcement learning, 2025.

[12] Xiaoyang Liu, Kangjie Bao, Jiashuo Zhang, Yunqi Liu, Yu Chen, Yuntian Liu, Yang Jiao, and Tao Luo. Atlas: Autoformalizing theorems through lifting, augmentation, and synthesis of data, 2025.

[13] Guoxiong Gao, Yutong Wang, Jiedong Jiang, Qi Gao, Zihan Qin, Tianyi Xu, and Bin Dong. Herald: A natural language annotated lean 4 dataset. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] OpenAI. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, April 2025.

[15] Lean FRO. A read-eval-print-loop for Lean 4. `https://github.com/leanprover-community/repl`, 2023.

[16] Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. Internlm-math: Open math large language models toward verifiable reasoning, 2024.

[17] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

[18] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in LLM-as-a-judge. In *The Thirteenth International Conference on Learning Representations*, 2025.

[19] DeepSeek-AI. Deepseek-v3 technical report, 2024.

[20] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[21] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, pages 1–3, 2023.

[22] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving Olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[23] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

[24] Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo, and Beng Chin Ooi. Alphaevolve: A learning framework to discover novel alphas in quantitative investment. In *Proceedings of the 2021 International conference on management of data*, pages 2208–2216, 2021.

[25] David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 2025.

[26] Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search, 2024.

[27] Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-prover: A frontier model for open-source automated theorem proving, 2025.

[28] Jingyuan Zhang, Qi Wang, Xingguang Ji, Yahui Liu, Yang Yue, Fuzheng Zhang, Di Zhang, Guorui Zhou, and Kun Gai. Leanabell-prover: Posttraining scaling in formal reasoning, 2025.

[29] Kefan Dong and Tengyu Ma. STP: Self-play LLM theorem provers with iterative conjecturing and proving. *arXiv preprint arXiv:2502.00212*, 2025.

[30] Zijian Wu, Suozhi Huang, Zhejian Zhou, Huaiyuan Ying, Jiayu Wang, Dahua Lin, and Kai Chen. Internlm2.5-stepprover: Advancing automated theorem proving via expert iteration on large-scale lean problems, 2024.

[31] Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning, 2022.

[32] Ran Xin, Chenguang Xi, Jie Yang, Feng Chen, Hang Wu, Xia Xiao, Yifan Sun, Shen Zheng, and Kai Shen. BFS-Prover: Scalable best-first tree search for llm-based automatic theorem proving, 2025.

[33] Guillaume Lample, Marie-Anne Lachaux, Thibaut Lavril, Xavier Martinet, Amaury Hayat, Gabriel Ebner, Aurélien Rodriguez, and Timothée Lacroix. Hypertree proof search for neural theorem proving. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 26337–26349, 2022.

[34] Haiming Wang et al. Dt-solver: Automated theorem proving with dynamic-tree sampling guided by proof-level value function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12632–12646, 2023.

[35] Yang Li, Dong Du, Linfeng Song, Chen Li, Weikang Wang, Tao Yang, and Haitao Mi. HunyuanProver: A scalable data synthesis framework and guided tree search for automated theorem proving, 2025.

[36] Qingxiang Wang, Cezary Kaliszyk, and Josef Urban. First experiments with neural translation of informal to formal mathematics. In *Intelligent Computer Mathematics: 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings 11*, pages 255–270. Springer, 2018.

[37] Christian Szegedy. A promising path towards autoformalization and general artificial intelligence. In *Intelligent Computer Mathematics: 13th International Conference, CICM 2020, Bertinoro, Italy, July 26–31, 2020, Proceedings 13*, pages 3–20. Springer, 2020.

[38] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

[39] Qi Liu, Xinhao Zheng, Xudong Lu, Qinxiang Cao, and Junchi Yan. Rethinking and improving autoformalization: towards a faithful metric and a dependency retrieval-based approach. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Zenan Li, Yifan Wu, Zhaoyu Li, Xinming Wei, Xian Zhang, Fan Yang, and Xiaoxing Ma. Autoformalize mathematical statements by symbolic equivalence and semantic consistency. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[41] Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. Autoformalizing euclidean geometry. In *Forty-first International Conference on Machine Learning*, 2024.

[42] Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

[43] George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition, 2024.

[44] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf`, 2024. GitHub repository.

[45] Jiewen Hu, Thomas Zhu, and Sean Welleck. miniCTX: Neural theorem proving with (long-)contexts. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. miniF2F: a cross-system benchmark for formal Olympiad-level mathematics, 2022.

[47] Joshua Vendrow, Edward Vendrow, Sara Beery, and Aleksander Madry. Large language model benchmarks do not test reliability. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[48] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[49] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper presents a revised version of miniF2F benchmark as well as systematic study of available autoformalization and theorem proving models. Main claims are supported by experimental results across different models and evaluation methods, e.g. human verification against LLM verification.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we discuss limitations of the proposed benchmark in the main text.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper presents a new benchmark dataset, miniF2F-v2, and does not present or discuss theoretical contributions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We disclose all sampling parameters, evaluation methods and Lean versions to reproduce all experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We plan to release the proposed benchmark along with the proofs.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We report all hyperparameters and Lean versions required to reproduce the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We adopt the standard ATP practice of issuing multiple prompts to an LLM and considering a proof successful if any generated attempt passes the formal verifier. This repetition mitigates LLM's inherent variability, since verification systems can unambiguously confirm correctness.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We disclose resources used for experimental section in the main text.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper follows the code of ethics and all generated/downloaded/used data has gone through sanity checks.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work proposes a new formal theorem proving/autoformalization benchmark. We do not explicitly address societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We will release the benchmark along with the proofs for the problems. We do not foresee risk associated with this benchmark.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and refer to every existing model and dataset used in this paper throughout the main text and appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The documentation will be provided alongside the benchmark dataset and proofs.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not conduct crowdsourcing research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve results related to human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use LLMs to evaluate our benchmark on a variety of theorem provers and autoformalization models. Moreover, we compare the accuracy and performance of LLM-based evaluators for informal-to-formal translation tasks.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Datasheet

Following the framework of [48] and [49], Table 8 provides the additional information about our dataset.

Table 8: Datasheet for our dataset

Questions	Answers
Motivation	
For what purpose was the dataset created?	To correct uncovered errors and inconsistencies within miniF2F dataset [7] and to further facilitate a challenging benchmark for LLM-based theorem provers and autoformalization models.
Who created the dataset and on behalf of which entity?	The authors of this paper.
Who funded the creation of the dataset?	The company where the authors work.
Any other comment?	None.
Composition	
What do the instances that comprise the dataset represent?	miniF2F theorems, each consisting of 4 components: informal statement and informal proofs in English, formal statements and formal proofs in Lean 4.
How many instances are there in total?	488 theorems
Does the dataset contain all possible instances or is it a sample of instances from a larger set?	Yes, we present all instances of the miniF2F dataset.
What data does each instance consist of?	A formal theorem in Lean with its formal proof and its informal description and informal proof.
Is there a label or target associated with each instance?	Only informal prefix.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	Not applicable.
Are there recommended data splits?	The dataset follows the same split as the original version, i.e. 244 test instances and 244 validation instances.
Are there any errors, sources of noise, or redundancies in the dataset?	No, there are no errors in the proposed dataset to the best of our knowledge. Every formal statement and formal proof compiles with no error in Lean4. All the informal statements and proofs are checked by human.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is a self-contained corrected version of miniF2F.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	No.
Collection Process	

Continued on next page

How was the data associated with each instance acquired?	All 488 instances originate from the miniF2F dataset and further augmented to correct mistakes and inconsistencies. Every theorem has a corresponding entry in the source dataset.
What mechanisms or procedures were used to collect the data?	The theorems were taken from miniF2F. Original problem statements were taken from official web pages of mathematical competitions such as IMO, AMC, AIME.
If the dataset is a sample from a larger set, what was the sampling strategy?	No, there is a one to one match between this dataset and the original miniF2F dataset.
Who was involved in the data collection process and how were they compensated?	The dataset was taken from open-source miniF2F dataset.
Over what time frame was the data collected?	Dataset was taken directly from miniF2F work.
Were any ethical review processes conducted?	No.
Preprocessing/cleaning/labeling	
Was any preprocessing/cleaning/labeling of the data done?	Yes, we performed post processing of the dataset manually to uncover incorrect, misleading, inconsistent or wrong statements.
Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?	The raw data is open-source and available to the public.
Is the software that was used to preprocess/-clean/label the data available?	We used Lean compiler to type check the formal statements.
Any other comments?	No.
Uses	
Has the dataset been used for any tasks already?	We evaluated numerous theorem proving models such as Deepseek-Prover-V1.5-RL, Goedel-Prover-SFT, Kimina-Prover-Distill-7B to evaluate the dataset. Additionally, we performed autoformalization experiments with Herald translator, Kimina autoformalizer and OpenAI o4-mini models.
Is there a repository that links to any or all papers or systems that use the dataset?	Public GitHub and HuggingFace links will be provided at a later date.
What (other) tasks could the dataset be used for?	The dataset may be used for benchmarking theorem provers and autoformalization models. Other tasks may involve incorporating the dataset into informal-formal theorem proving environments.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?	We do not anticipate future issues emerging from the proposed benchmark dataset.
Are there tasks for which the dataset should not be used?	The test set, and preferably validation set, should not be used to train the theorem provers and autoformalizers.
Any other comments?	No.
Distribution	
Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?	Yes, we will release the dataset on public platforms such as GitHub and HuggingFace at a later date.
How will the dataset will be distributed?	GitHub, HuggingFace.

Continued on next page

When will the dataset be distributed?	The dataset will be distributed along with the camera-ready version of the submission.
Will the dataset be distributed under a copyright or other intellectual property license, and/or under applicable terms of use?	Yes, the dataset will be released under the MIT license.
Have any third parties imposed IP-based or other restrictions on the data associated with the instances?	No.
Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?	No.
Any other comments?	No.
Maintenance	
Who will be supporting/hosting/maintaining the dataset?	The last author of the submission.
How can the owner/curator/manager of the dataset be contacted?	The manager of the dataset may be reached through an email or any other public means, such as GitHub profile.
Is there an erratum?	Formal statements do not require erratum, and corrected informal statements are an erratum of the original informal statements sourced from the miniF2F benchmark.
Will the dataset be updated?	Yes, we plan to periodically update the dataset as new versions of Lean become available.
If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?	Not applicable.
Will older versions of the dataset continue to be supported/hosted/maintained?	Yes.
If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?	Yes, since we release coupled informal and formal statements, others may expand the dataset to other formal languages, such as Isabelle. Informal statements can also be translated to other languages such as Chinese. The problems may be formalized further to cover more formal theorem proving languages.
Any other comments?	No.

B Effect of a clearly structured informal proof as opposed to a vague one

A failure case of Kimina prover is $aime\_1987\_p5$. However, when we provide a more clear informal proof for this theorem, Kimina Prover succeeds in proving it. This indicates the positive effect of a better informal proof on the existing theorem provers, and the higher quality of miniF2F-v2.

aime_1987_p5 with informal proofs

miniF2F-v1

Find $3x^2y^2$ if x and y are integers such that $y^2 + 3x^2y^2 = 30x^2 + 517$. Show that it is 588.

miniF2F-v2

Find $3x^2y^2$ if x and y are integers such that $y^2 + 3x^2y^2 = 30x^2 + 517$. Show that $3x^2y^2$ is 588.

```
/-Formal Statement-/
theorem aime_1987_p5 (x y : ℤ) (h₀ : y ^ 2 + 3 * (x ^ 2 * y ^ 2) = 30 * x ^
    2 + 517) :
    3 * (x ^ 2 * y ^ 2) = 588 := by
    sorry
```

miniF2F-v1

Informal proof:
If we move the x^2 term to the left side, it is [[SFFT|factorable]]:
$(3x^2 + 1)(y^2 - 10) = 517 - 10$
507 is equal to $3 \cdot 13^2$. Since x and y are integers, $3x^2 + 1$ cannot equal a multiple of three. 169 doesn't work either, so $3x^2 + 1 = 13$, and $x^2 = 4$. This leaves $y^2 - 10 = 39$, so $y^2 = 49$. Thus, $3x^2y^2 = 3 \times 4 \times 49 = 588$.

miniF2F-v2

Informal proof:
From the equation $y^2 + 3x^2y^2 = 30x^2 + 517$ we first rewrite it as

$$3x^2y^2 = 30x^2 + 517 - y^2.$$

Since squares are nonnegative this forces $y^2 \leq 517$, hence $-22 \leq y \leq 22$. There are only finitely many integer choices for y in this range, so we check each one: for each fixed y, the rewritten equation becomes a concrete quadratic in x which can be checked by direct computation to yield $3x^2y^2 = 588$. Thus in all cases the desired conclusion holds.

C Wrong formalization because of unfamiliarity with Mathlib

Another failure case of Kimina Prover is $algebra\_cubrtrp1oncubrtreq3\_rcubp1onrcubeq5778$ ($algebra\_5778$ in short). The informal statement for this theorem relies upon a simple definition of cube root common in precollege math. However, in Mathlib, the nth root, via the *rpow* function, is defined using a more advanced definition that is compatible with a broader network of definitions including the real roots of negative complex numbers and the continuity of roots of negative real numbers. The definition of nth root in Mathlib does not correspond to the common definition in precollege math, and therefore, the formalization of this problem in miniF2F is wrong and unprovable. In other words, the *rpow* function in Mathlib is not equivalent to the definition of cube root as stated in the informal statement for this theorem and the use of *rpow* in this formalization makes this theorem unprovable in Lean.

In fact, we write a formal proof giving a counterexample for the case when the variable is negative, proving that this theorem is unprovable in Lean as it appears in miniF2F-v1, i.e., a formal proof for unprovability of this theorem.

As alternative, in the below diagram, we show three correct formalizations of this problem. The first version still relies on the definition of nth root in Lean, but makes the variable nonnegative avoiding any conflict with lemma: $Real.rpow\_def\_of\_neg$ in Mathlib. The second version defines a new n^{th} root function corresponding to precollege math. The third formalization excludes the only possible negative root of equation h_0. After correcting the formal statement, Kimina Prover successfully proves this theorem. The proofs of the correct formalization and the proof of counterexample for the incorrect statement in miniF2F wil be released with the paper.

Informal Statement: Let r be a real number such that $r^{\frac{1}{3}} + \frac{1}{r^{\frac{1}{3}}} = 3$. Show that $r^3 + \frac{1}{r^3} = 5778$.

Incorrect Formalization [miniF2F-v1]
theorem algebra5778
$(r : \mathbb{R})$
$(h_0 : r^{(1/3:\mathbb{R})} + 1/r^{(1/3:\mathbb{R})} = 3 :$
$r^3 + 1/r^3 = 5778$:= by

Correct Formalization #1
theorem algebra5778
$(r : \mathbb{R})$
$(h_0 : r^{(1/3:\mathbb{R})} + 1/r^{(1/3:\mathbb{R})} = 3)$
$(h_1 : 0 \leq r) :$
$r^3 + 1/r^3 = 5778$:= by

Correct Formalization #2 [miniF2F-v2]
theorem algebra5778
$(r : \mathbb{R})$ $(qpow : \mathbb{R} \to \mathbb{Q} \to \mathbb{R})$
$(hq : qpow = fun\ x\ q \mapsto if\ 0 \leq x\ then$
$x.rpow(\uparrow a)\ else\ -(-x).rpow(\uparrow a))$
$(h_0 : qpow\, r\,(1/3) + 1/qpow\, r\,(1/3) = 3) :$
$r^3 + 1/r^3 = 5778$:= by

Correct Formalization #3
theorem algebra5778
$(r : \mathbb{R})$
$(h_0 : r^{(1/3:\mathbb{R})} + 1/r^{(1/3:\mathbb{R})} = 3)$
$(h_1 : r^{(1/3:\mathbb{R})} \neq (1/2)(-r)^{(1/3:\mathbb{R})}) :$
$r^3 + 1/r^3 = 5778$:= by

D Examples of autoformalizer outputs on miniF2F v1 and v2

In this section, we present three examples of modified problems and analyze their impact on autoformalization models. We show that supplying corrected miniF2F informal statements can significantly affect model performance, and that the specific nature of each formalization error drives different model behaviors.

D.1 `mathd_algebra_31`

The original informal statement of `mathd_algebra_31` omits several critical details and defined ambiguously. In particular, it denotes a limit by "…", leaving the definition ambiguous. When tasked with this problem, both Herald and Kimina attempt to encode the underlying recursive function explicitly, and fail.

To fix these issues, we provide a clearly specified version of the problem with an explicit recursive definition. After this revision, Kimina successfully produces a correct autoformalization although Herald still fails.

These results demonstrate that aligning informal and formal problem descriptions improves the reliability of autoformalization benchmarks, and it can improve the accuracy of existing models, too.

D.2 `imo_1960_p2`

The original informal statement fails to specify the proof goal. As a result, autoformalization models cannot generate a valid formalization because the target is undefined. We resolved this by adding an explicit statement of the goal. With this modification, both Herald and Kimina successfully produce correct formalizations. Although better informal-formal match increases the complexity for automated provers, our results demonstrate that faithful, detailed translations improve some autoformalization attempts. Interestingly, since o4-mini is a general purpose model, it was able to find a solution and use it to define a correct goal.

D.3 `amc12b_2003_p6`

In this example, the informal statement is correct. However, the formal statement is not a correct translation of it. The formal statement asks for an extra solution, and drops the intricate detail of "a possible solution" in the informal statement. This can be considered a mismatch between the informal and formal statements. Despite the fact that each of them is correct and provable, they do not translate to each other. One possible approach is to keep the correct informal statement and change the formal statement to correspond to it. In this case, however, we chose to change the formal statement so that it corresponds to the formal one. After this update, both Herald and o4-mini correctly autoformalize the problem, as they did on `imo_1960_p2`.

Correction of mathd_algebra_31 with autoformalization outputs

miniF2F-v1

If $\sqrt{x+\sqrt{x+\sqrt{x+\sqrt{x+\cdots}}}}=9$, find x. Show that it is 72.

miniF2F-v2

Consider a nonnegative real number x, and a recursive function u from natural numbers to real numbers. For all natural numbers, we have $u(n+1)=\sqrt{x+u(n)}$. Assume that the as n reaches infinity in its limit, u(n) goes to 9. What is the value of x? Show that it is 72.

```
/-Formal statement-/
theorem mathd_algebra_31_minif2f_v1
    (x : NNReal)
    (u : ℕ → NNReal)
    (h₀ : ∀ n, u (n + 1) =
    NNReal.sqrt (x + u n))
    (h₁ : Filter.Tendsto u
    Filter.atTop (nhds 9)) :
    9 = NNReal.sqrt (x + 9) := by
    sorry
```

```
/-Formal statement-/
theorem mathd_algebra_31_minif2f_v2
    (x : ℝ) (hx : 0 ≤ x)
    (u : ℕ → ℝ)
    (h₀ : ∀ n, u (n + 1) = Real.sqrt
    (x + u n))
    (h₁ : Filter.Tendsto u
    Filter.atTop (nhds 9)) :
    x = 72 := by
    sorry
```

```
/-Kimina Formalized - miniF2F-v1-/
theorem mathd_algebra_31_kimina
    {x : ℝ}
    (hx : 0 ≤ x)
    (h : sqrt(x + sqrt(x + sqrt(x +
    sqrt(x + sqrt(x + 0))))) = 9) :
    x = 72 := by
    sorry
```

```
/-Kimina Formalized - miniF2F-v2-/
theorem mathd_algebra_31_kimina
   (x : ℝ) (hx : 0 ≤ x) (u : ℕ → ℝ)
   (hu : ∀ n, u (n + 1) = sqrt (x +
    u n))
   (h : Tendsto u atTop (nhds 9)) :
   x = 72 := by
   sorry
```

```
/-Herald Formalized - miniF2F-v1-/
theorem mathd_algebra_31_herald :
    sqrt(x + sqrt(x + sqrt(x +
    sqrt(x + 9)))) = 9 ↔ x = 72   :=
      by
  sorry
```

```
/-Herald Formalized - miniF2F-v2-/
theorem mathd_algebra_31_herald (x :
    ℝ) (h : 0 < x) (hx : Σ' i : ℕ,
    (Real.sqrt (x + (Real.sqrt (x +
    (Real.sqrt (x + (Real.sqrt (x +
    0)))))))) = 9) : x = 72 := by
  sorry
```

```
/-o4-mini Formalized - miniF2F-v1-/
theorem mathd_algebra_31_o4mini (x y
    : ℝ) (h1 : y = Real.sqrt (x +
    y)) (h2 : y = 9) : x = 72 := by
  sorry
```

```
/-o4-mini Formalized - miniF2F-v2-/
theorem mathd_algebra_31_o4mini {x :
    ℝ} (h : 9 = Real.sqrt (x + 9)) :
    x = 72 := by
  sorry
```

Correction of imo_1960_p2 with autoformalization outputs

miniF2F-v1

For what values of the variable x does the following inequality hold:
$$\frac{4x^2}{(1-\sqrt{2x+1})^2} < 2x+9\ ?$$

miniF2F-v2

Let x be a real number. Assume that:
- $1+2x \geq 0$ (so that the square root is defined), - $(1-\sqrt{1+2x})^2 \neq 0$ (so the denominator is nonzero), and - $\frac{4x^2}{(1-\sqrt{1+2x})^2} < 2x+9$.

Prove that:
$$-\frac{1}{2} \leq x \quad \text{and} \quad x < \frac{45}{8}.$$

```
/-Formal Statement-/
theorem imo_1960_p2 (x : ℝ) (h₀ : 0 ≤ 1 + 2 * x) (h₁ : (1 - Real.sqrt (1 +
    2 * x)) ^ 2 ≠ 0)
    (h₂ : 4 * x ^ 2 / (1 - Real.sqrt (1 + 2 * x)) ^ 2 < 2 * x + 9) : -(1 /
    2) ≤ x ∧ x < 45 / 8 := by
  sorry
```

```
/-Kimina Formalized - miniF2F-v1-/
theorem imo_1960_p2_kimina (x : ℝ)
    (hx : 0 ≤ 2 * x + 1 ∧ 1 - sqrt
    (2 * x + 1) ≠ 0) :
  (4 * x ^ 2 / (1 - sqrt (2 * x +
    1)) ^ 2 < 2 * x + 9 ↔ x < 1 / 2
    ∨ x > 7 / 2) := by
    sorry
```

```
/-Kimina Formalized - miniF2F-v2-/
theorem imo_1960_p2_kimina (x : ℝ)
    (h₀ : 1 + 2 * x ≥ 0) (h₁ : (1 -
    Real.sqrt (1 + 2 * x))^2 ≠ 0)
    (h₂ : 4 * x^2 / (1 - Real.sqrt
    (1 + 2 * x))^2 < 2 * x + 9) :
  -1 / 2 ≤ x ∧ x < 45 / 8 := by
   sorry
```

```
/-Herald Formalized - miniF2F-v1-/
theorem imo_1960_p2_herald (x : ℝ) :
    (4 * x ^ 2 / (1 - Real.sqrt (2 *
    x + 1)) ^ 2 < 2 * x + 9) ↔ x <
    11 / 2   :=  by
    sorry
```

```
/-Herald Formalized - miniF2F-v2-/
theorem imo_1960_p2_herald (x : ℝ)
    (h₁ : 1 + 2 * x ≥ 0) (h₂ : (1 -
    Real.sqrt (1 + 2 * x)) ^ 2 ≠ 0)
    (h₃ : 4 * x ^ 2 / (1 - Real.sqrt
    (1 + 2 * x)) ^ 2 < 2 * x + 9) :
    -1 / 2 ≤ x ∧ x < 45 / 8   :=  by
   sorry
```

```
/-o4-mini Formalized - miniF2F-v1-/
theorem imo_1960_p2_o4mini (x : ℝ)
    (h_nonneg : 2 * x + 1 ≥ 0)
    (h_ne_zero : 1 - Real.sqrt (2 *
    x + 1) ≠ 0) :
  4 * x ^ 2 / (1 - Real.sqrt (2 * x +
    1)) ^ 2 < 2 * x + 9 ↔
  -1 / 2 ≤ x ∧ x < 45 / 8 ∧ x ≠ 0 :=
    by
  sorry
```

```
/-o4-mini Formalized - miniF2F-v2-/
theorem imo_1960_p2 (x : ℝ)
  (h1 : 1 + 2 * x ≥ 0)
  (h2 : (1 - Real.sqrt (1 + 2 * x))
    ^ 2 ≠ 0)
  (h3 : 4 * x ^ 2 / (1 - Real.sqrt
    (1 + 2 * x)) ^ 2 < 2 * x + 9) :
  -1 / 2 ≤ x ∧ x < 45 / 8 := by
  sorry
```

[Informal and Formal Mismatch] Comparison of amc12b_2003_p6 across miniF2F-v1 and miniF2Fv2

miniF2F-v1

The second and fourth terms of a geometric sequence are 2 and 6. Which of the following is a possible first term?

(A) $-\sqrt{3}$ **(B)** $-\frac{2\sqrt{3}}{3}$ **(C)** $-\frac{\sqrt{3}}{3}$ **(D)** $\sqrt{3}$ **(E)** 3

Show that it is **(B)** $-\frac{2\sqrt{3}}{3}$.

miniF2F-v2

The second and fourth terms of a geometric sequence are 2 and 6. Show that the first term is either $-\frac{2\sqrt{3}}{3}$ or $\frac{2\sqrt{3}}{3}$.

```
/-Formal Statement-/
theorem amc12b_2003_p6 (a r : ℝ) (u : ℕ → ℝ)
  (h₀ : ∀ k, u k = a * r ^ k) (h₁ : u 1 = 2)
  (h₂ : u 3 = 6) : u 0 = 2 / Real.sqrt 3 ∨ u 0 = -(2 / Real.sqrt 3) := by
  sorry
```

```
/-Kimina Formalized - miniF2F-v1-/
theorem amc12b_2003_p6_kimina (a : ℕ
    → ℝ) (h : ∃ r, ∀ n, a n = a 0 *
    r ^ n) (ha : a 1 = 2) (hb : a 3
   = 6) :
  a 0 = -2 * sqrt 3 / 3 := by
  sorry
```

```
/-Kimina Formalized - miniF2F-v2-/
theorem amc12b_2003_p6_kimina {a r :
    ℝ} (hr : r ≠ 0) (ha : a ≠ 0)
    (h1 : a * r = 2) (h2 : a * r^3 =
    6) :
  a = 2 * Real.sqrt 3 / 3 ∨a = -2 *
    Real.sqrt 3 / 3 := by
  sorry
```

```
/-Herald Formalized - miniF2F-v1-/
theorem amc12b_2003_p6_herald  (a : ℝ
   )
  (h₀ : a * r = 2)
  (h₁ : a * r^3 = 6) :
  r = 3 ∧ a = -2 * Real.sqrt 3 / 3
   :=  by
  sorry
```

```
/-Herald Formalized - miniF2F-v2-/
theorem amc12b_2003_p6_herald (a : ℝ
    ) (r : ℝ) (h₁ : a * r = 2) (h₂ :
    a * r^3 = 6) : a = -2 *
    Real.sqrt 3 / 3 ∨ a = 2 *
    Real.sqrt 3 / 3 := by
  sorry
```

```
/-o4-mini Formalized - miniF2F-v1-/
theorem amc12b_2003_p6_o4mini : ∃ (a
    r : ℝ), r ≠ 0 ∧ a * r = 2 ∧ a *
    r ^ 3 = 6 ∧ a = -2 * sqrt 3 / 3
   := by
  sorry
```

```
/-o4-mini Formalized - miniF2F-v2-/
theorem amc12b_2003_p6_o4mini (a r :
    ℝ) (h1 : a * r = 2) (h2 : a *
    r^3 = 6) :
  a = 2 * Real.sqrt 3 / 3 ∨ a = -2 *
    Real.sqrt 3 / 3 := by
  sorry
```

E Examples of modified statements

E.1 `induction_pord1p1on2powklt5on2`

The original problem was unprovable due to a missing pair of parentheses, which led to an incorrect formalization. We corrected this error in miniF2F-v2 by inserting the appropriate parentheses.

E.2 `aime_1990_p4`

In the original formal statement, three additional hypotheses ($\mathtt{h}_1, \mathtt{h}_2, \mathtt{h}_3$) explicitly assert that certain expressions are nonzero, which simplifies proof generation for theorem provers. These are extra assumptions that are not present in the informal statement that was given to the participants of AIME 1990, and they are not necessary to prove the theorem. To remove this discrepancy between the formal and informal statements, we remove the added hypothesis from the formal statement. This makes the theorem more challenging for theorem provers as they have to prove each of those hypothesis as steps to prove the theorem.

[Unprovable] Comparison of induction_pord1p1on2powklt5on2 across miniF2F-v1 and miniF2Fv2

Show that for positive integer n, $(\prod_{k=1}^{n}(1+1/2^k)) < 5/2$.

miniF2F-v1

```
/-Formal Statement - miniF2F-v1-/
theorem
    induction_pord1p1on2powklt5on2
  (n : ℕ) (h0 : 0 < n) :
  (∏ k in Finset.Icc 1 n, 1 + (1 : ℝ
    ) / 2 ^ k) < 5 / 2 := by
  sorry
```

miniF2F-v2

```
/-Formal Statement - miniF2F-v2-/
theorem
    induction_pord1p1on2powklt5on2
  (n : ℕ) (h0 : 0 < n) :
  (∏ k ∈ Finset.Icc (1:ℕ) n, ((1 :
    ℝ) + (1 : ℝ) / 2 ^ k)) < (5 / 2
    : ℝ) := by
  sorry
```

[Simplified] Comparison of aime_1990_p4 across miniF2F-v1 and miniF2Fv2

Find the positive solution to $\frac{1}{x^2-10x-29}+\frac{1}{x^2-10x-45}-\frac{2}{x^2-10x-69}=0$. Show that it is 13.

miniF2F-v1

```
/-Formal Statement - miniF2F-v1-/
theorem aime_1990_p4
  (x : ℝ) (h0 : 0 < x)
  (h1 : x ^ 2 - 10 * x - 29 ≠ 0)
  (h2 : x ^ 2 - 10 * x - 45 ≠ 0)
  (h3 : x ^ 2 - 10 * x - 69 ≠ 0)
  (h4 : 1 / (x ^ 2 - 10 * x - 29) +
    1 / (x ^ 2 - 10 * x - 45) - 2 /
    (x ^ 2 - 10 * x - 69) = 0) :
  x = 13 := by
  sorry
```

miniF2F-v2

```
/-Formal Statement - miniF2F-v2-/
theorem aime_1990_p4
  (x : ℝ) (h0 : 0 < x)
  (h4 : 1 / (x ^ 2 - 10 * x - 29) +
    1 / (x ^ 2 - 10 * x - 45) - 2 /
    (x ^ 2 - 10 * x - 69) = 0) :
  x = 13 := by
  sorry
```

E.3 `amc12b_2021_p9`

In this example, we retained each logarithm in its base form, writing $\log_c a$ directly rather than converting it to $\frac{\log a}{\log c}$, so that the formal statement exactly corresponds to the informal one as it appeared in the AMC. These changes yield a slightly more challenging version of the problem while preserving its original intent. And we observe that all three theorem proving models, Deepseek-Prover-V1.5-RL, Goedel-Prover-SFT and Kimina-Prover-Preview-Distill-7B, fail on the modified version of this theorem.

E.4 `mathd_algebra_487`

In the original miniF2F version, the problem was oversimplified: it introduced four variables (instead of two) and omitted the Euclidean-space context, therefore providing extra hints that eases proof generation task for LLMs. In miniF2F-v2, we restore the two-variable formulation over $\mathbb{R}^2$, remove these implicit assumptions. This leads to a spike in the problem's difficulty. These more faithful and challenging instances yield a more rigorous benchmark for evaluating theorem provers.

[Simplified] Comparison of amc12b_2021_p9 across miniF2F-v1 and miniF2Fv2

miniF2F-v1

What is the value of $\frac{\log_2 80}{\log_{40} 2} - \frac{\log_2 160}{\log_{20} 2}$?

$(\textbf{A})0 \qquad (\textbf{B})1 \qquad (\textbf{C})\frac{5}{4}$

$(\textbf{D})2 \qquad (\textbf{E})\log_2 5$

Show that it is (D).

```
/-Formal Statement - miniF2F-v1-/
theorem amc12b_2021_p9 :
  Real.log 80 / Real.log 2 /
    (Real.log 2 / Real.log 40) -
  Real.log 160 / Real.log 2 /
    (Real.log 2 / Real.log 20)
  = 2 := by
  sorry
```

miniF2F-v2

What is the value of $\frac{\log_2 80}{\log_{40} 2} - \frac{\log_2 160}{\log_{20} 2}$? Show that it is 2.

```
/-Formal Statement - miniF2F-v2-/
theorem amc12b_2021_p9 :
  Real.logb 2 80 / (Real.logb 40 2) -
  Real.logb 2 160 / Real.logb 20 2) =
     2 := by
  sorry
```

[Excessively Simplified] Comparison of mathd_algebra_487 across miniF2F-v1 and miniF2Fv2

What is the distance between the two intersections of $y = x^2$ and $x + y = 1$? Show that it is $\sqrt{10}$.

miniF2F-v1

```
/-Formal Statement - miniF2F-v1-/
theorem mathd_algebra_487
  (a b c d : ℝ) (h₀ : b = a ^ 2)
  (h₁ : a + b = 1) (h₂ : d = c ^ 2)
  (h₃ : c + d = 1) (h₄ : a ≠ c) :
  Real.sqrt ((a - c) ^ 2 + (b - d) ^
    2) = Real.sqrt 10 := by
  sorry
```

miniF2F-v2

```
/-Formal Statement - miniF2F-v2-/
theorem mathd_algebra_487
  (F G I : Set (EuclideanSpace ℝ
    (Fin 2)))
  (hF : F = { x | x 1 = (x 0) ^ 2})
  (hG : G = { x | x 0 + x 1 = 1})
  (hI : I = (F ∩ G))
  (A B : EuclideanSpace ℝ (Fin 2))
  (h₀ : ∀ x, x ∈ I ↔ x = A ∨ x = B):
  dist A B = Real.sqrt 10 := by
  sorry
```

F Statistics about our modifications

We present the distribution of uncovered errors and inconsistencies in Figure 3. We observe that the majority of the problems in both test and validation sets are simplified and do not reflect the intended difficulty of the problems. Moreover, approximately 40% of formal statements across both sets contained an error, making the evaluation of LLMs on this benchmark less reliable.

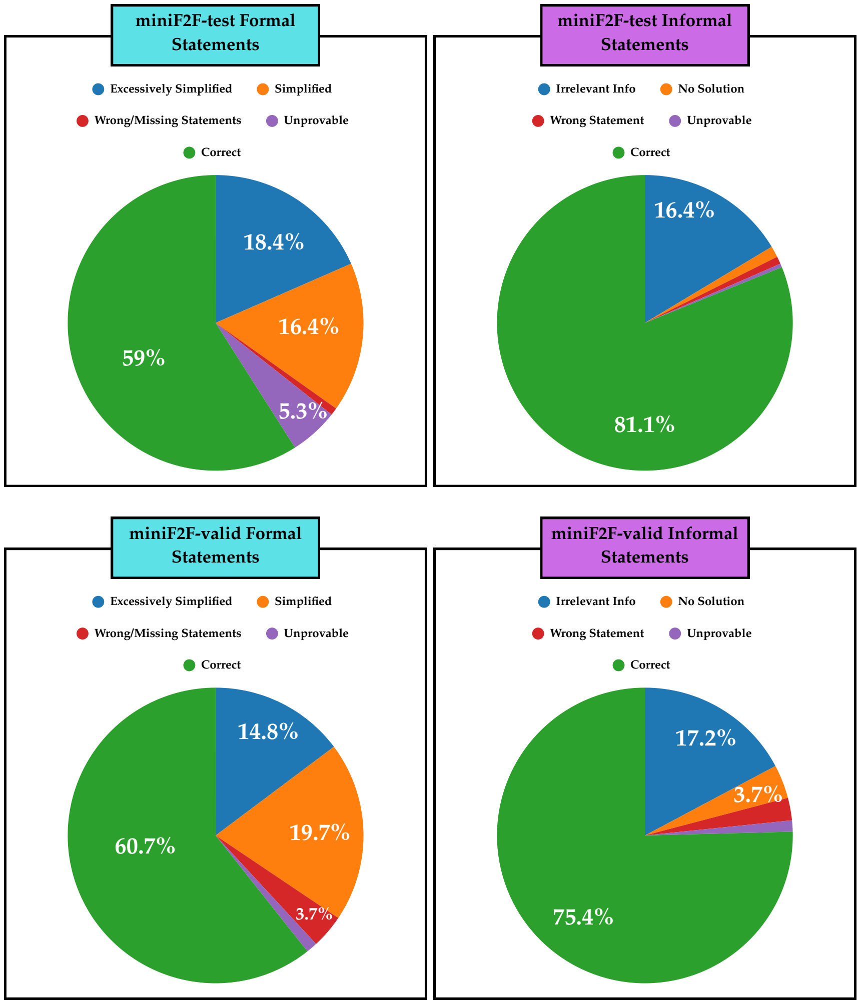

Figure 3: Pie charts of identified formal and informal statement errors within miniF2F benchmark across test and validation sets.

G Responsibility statement and License information

We plan to release our dataset under the MIT license on GitHub and HuggingFace. The authors of this submission bear the responsibility in case of rights violation.