Multi-language Diversity Benefits Autoformalization

Albert Q. Jiang
University of Cambridge
qj213@cam.ac.uk

Wenda Li
University of Edinburgh
wenda.li@ed.ac.uk

Mateja Jamnik
University of Cambridge
mateja.jamnik@cl.cam.ac.uk

Abstract

Autoformalization is the task of translating natural language materials into machine-verifiable formalisations. Progress in autoformalization research is hindered by the lack of a sizeable dataset consisting of informal-formal pairs expressing the same essence. Existing methods tend to circumvent this challenge by manually curating small corpora or using few-shot learning with large language models. But these methods suffer from data scarcity and formal language acquisition difficulty. In this work, we create `MMA`, a large, flexible, multi-language, and multi-domain dataset of informal-formal pairs, by using a language model to translate in the reverse direction, that is, from formal mathematical statements into corresponding informal ones. Experiments show that language models fine-tuned on `MMA` can produce up to $29-31\%$ of statements acceptable with minimal corrections on the `miniF2F` and `ProofNet` benchmarks, up from 0% with the base model. We demonstrate that fine-tuning on multi-language formal data results in more capable autoformalization models even on single-language tasks.

1 Introduction

Formal mathematics refers to mathematical content that is represented in a formal language that can be mechanically checked by a computer. Practitioners express mathematics in formal languages integrated into proof assistants like HOL Light [Harrison, 1996], Isabelle [Paulson, 1994], Coq [Barras et al., 1999], and Lean [de Moura et al., 2015]. *Autoformalization* is the task of translating natural language materials into verifiable formalisations. An ideal autoformalization engine can reduce the excessive cost for modern mathematical results to be verified [Ball, 2012, Scholze and Stix, 2018]. It opens up the vast amount of mathematics expressed in natural language to automated reasoning research fields that rely on formal languages, like automated theorem proving [Wu et al., 2022].

The hope of automatically translating informal mathematics into formally verifiable content is as old as formal mathematics [Whitehead and Russell, 1925–1927]. Only very recently, the breakthroughs in neural networks and Neural Machine Translation (NMT) enabled autoformalization to be learned [Wang et al., 2020, Wu et al., 2022, Jiang et al., 2023b]. NMT methods typically require a large *parallel dataset*, that is, a dataset consisting of pairs of sequences expressing the same meaning in both the source and the target language. The most challenging part of autoformalization research is constructing such a parallel dataset in a natural and a formal language, satisfying two conditions simultaneously: (1) the natural language component is close to how mathematics is actually written; and (2) the number of datapoints is large enough for the data-hungry machine learning methods. This is hard, because manually translating informal mathematical content into a formal language is only doable by highly trained experts in both mathematics and computer science, hence costly.

In this work, we address the lack of a parallel dataset by leveraging a state-of-the-art Large Language Model (LLM), GPT-4 [OpenAI, 2023]: we used it to translate the two largest formal corpora, Archive of Formal Proofs in the language of Isabelle, and mathlib4 in the language of Lean4, into natural language. This process was enabled by the key observations that informalisation is much easier than formalisation, and a powerful LLM can produce diverse natural language outputs. As a result, we

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2659

Table 1: Example parallel pairs from MMA.

Isabelle statement	GPT-4 informalisation

`lemma eint_minus_le:`
`  assumes "(b::eint) < c"`
`  shows "c - b > 0"`
	The lemma named “eint_minus_le” assumes that an extended integer “b” is less than another extended integer “c”. It then shows that the result of “c” subtracted by “b” is greater than zero.

`lemma closed_superdiagonal:`
`  "closed {(x,y) | x y.  x ≥ (y::`
`  ('a::{linorder_topology}))}"`
	The set of all pairs of elements (x, y) such that x is greater than or equal to y, is a closed set in the context of a linearly ordered topology.

Lean4 statement	**GPT-4 informalisation**
`theorem norm_eq_one_of_pow_eq_one` $\{\zeta : \mathbb{C}\}$ $\{n : \mathbb{N}\}$ (h : $\zeta^n = 1$) (hn : $n \neq 0$): $\lVert \zeta \rVert = 1$:=	For a complex number ζ and a natural number n, if ζ to the power of n equals 1 and n is not equal to 0, then the norm of ζ is equal to 1.
`theorem mul_dvd_mul_iff_left` $\{a\ b\ c : \mathbb{N}\}$ (ha : $0 < a$) : $a * b \mid a * c$ $\leftrightarrow b \mid c$:=	For any three natural numbers a, b, and c, where a is greater than 0, a times b divides a times c if and only if b divides c.

created a parallel dataset of 332K informal-formal pairs, which we refer to as the MMA (Multi-language Mathematical Autoformalization) dataset. To the best of our knowledge, this is the first dataset of natural-formal language aligned data with more than one formal language. The only similar work was that of Azerbayev et al. [2023], which has only one formal language (Lean3) and is 4x smaller than our dataset. Four examples of MMA are shown in Table 1.

We fine-tuned two open-source LLMs, LLaMA-33B [Touvron et al., 2023] and Mistral 7B [Jiang et al., 2023a], on MMA to generate corresponding formal expressions given the informal ones. The trained model was then evaluated on two autoformalization benchmarks, miniF2F and ProofNet. Manual inspection of 50 problems for each model from each benchmark showed that after fine-tuning, the models could produce $29 - 31\%$ of formal statements on the benchmarks that require no or minimal correction, whereas the raw model produced 0%. We also fine-tuned two identical models on the Isabelle and the Lean4 components of MMA separately for the same number of steps. Their autoformalization performances are significantly weaker than the model trained on multi-language data, demonstrating that parallel data containing multiple formal languages is crucial for autoformalization training.

Contributions:

- We informalise all formal statements from the Archive of Formal Proofs and mathlib4, creating MMA, a dataset of informal-formal pairs. This is the first natural-formal language aligned dataset containing multiple formal languages.
- We train the first language models that can autoformalize to multiple languages in the zero-shot setting, and manually evaluate them on two autoformalization benchmarks.
- We verify that: (1) language models trained on MMA acquire strong autoformalization abilities; and (2) language models trained on MMA have greater autoformalization performance than those trained on single-language partitions of it with the same computational budget.
- We release the fine-tuned models for inference. We also release the MMA dataset for people to train their autoformalization models on, and to enrich MMA with more domains and languages.

Improving autoformalization ability of models has the potential of translating copious digital repositories of informal human knowledge into formal languages of reasoning tools, and thus presents an opportunity to formally verify human informal arguments and solutions. High quality datasets such as MMA and autoformalization models like ours pave the way towards this goal[1].

[1] The MMA dataset and the fine-tuned models are available from the official repository: MMA.

2 Related Work

Autoformalization Datasets. Wang et al. [2018, 2020] manually aligned a small parallel dataset and generated a larger parallel dataset with a rule-based informalisation tool [Bancerek, 2006] from Mizar to LaTeX. Manual alignment is almost as expensive as formalising mathematics anew. Moreover, unlike generative neural informalisation tools (e.g., GPT4), symbolic informalisation tools such as Naproche [Cramer et al., 2009] result in natural language content that lacks the inherent diversity and flexibility in expression: they are rigid and not natural-language-like. Finally, symbolic informalisation tools are hard to design and implement. They also differ a lot for different formal languages, hence the approach is not scalable for multiple formal languages.

Wu et al. [2022] sought to eliminate altogether the need for a parallel dataset by leveraging the in-context learning ability of LLMs: they provided a couple of parallel examples, and asked the LLMs to find a formal counterpart for the informal problem (limited to high-school algebra or number theory). This approach is very effective when the test domain is limited. But when there are many test domains, finding the correct parallel examples becomes difficult: the LLM invents syntactically incorrect segments when it does not know the formal syntax for certain concepts [Wu et al., 2022, Case Study 3]. Liu et al. [2023] and Huang et al. [2024] both utilised autoformalization to create aligned informal-formal pairs of data that are verified either manually or mechanically (for proofs), but did not perform large-scale synthesis of corresponding informal-formal theorem statements. Li et al. [2024] provides a more detailed survey on autoformalization datasets. In summary, there is no existing method, like the one we propose here, that is scalable both in terms of formal languages and mathematical domains.

Back-translation. In natural language machine translation literature, the quality of translation heavily depends on the quality of the parallel data between two languages. However, for all but a few language pairs (e.g., `en-fr`), such parallel data is rare and hard to curate [Guzmán et al., 2019]. Back-translation is one of the most effective methods to improve translation quality [Sennrich et al., 2016, Artetxe et al., 2018] in this setting, which is similar to ours. Back-translation uses an existing target-to-source model to turn ground-truth target sequences into noisy source sequences. Then, it bootstraps a source-to-target model to reconstruct the ground-truth target from the noisy source.

Usually, the back-translation process is practised in both directions of translation, that is, from source to target and from target to source, and is iterated until convergence. When back-translation is practised in one direction only (because the model from target to source is called through an API and not trainable, for example), this process is referred to as "distilled back-translation". Azerbayev et al. [2023] used OpenAI's Codex [Chen et al., 2021] model to perform distilled back-translation to improve their own model's autoformalization capabilities. `MMA` differs from their dataset mainly in that `MMA` contains data from multiple formal languages and has four times as many datapoints.

Language Models for Executable Programs and Reasoning. Since OpenAI's Codex [Chen et al., 2021], multiple LLMs have been trained for code completion and infilling that stem from natural language [Yu et al., 2018, Austin et al., 2021, Fried et al., 2023]. Related is also the research on natural language mathematical and logical reasoning [Cobbe et al., 2021, Lewkowycz et al., 2022, Shi et al., 2022] that demonstrates that LLMs can comprehend mathematics and produce reasoning chains in natural language to a degree. Interestingly, distillation from larger, more capable models can effectively boost the reasoning ability of smaller models [Fu et al., 2023]. However, none of these works trained language models for the task of autoformalization, which is the gap that our work fills.

3 Dataset

As established above, there is no existing parallel corpus that satisfies the following crucial criteria for autoformalization model training:

1. The informal data is diverse and flexible, similar to natural mathematical communication.
2. The size is suitable for neural model training (≥ 100K datapoints).

Informalisation. In this work, we use a powerful neural model (GPT-4) to generate informal data from existing formal libraries (informalisation) to create a high-quality parallel corpus. We argue, both analytically and empirically, that informalisation is an easier task than formalisation. Hence, our approach of leveraging the power and flexibility of language models for informalisation indeed produces a parallel corpus that satisfies both of the criteria above.

https://doi.org/10.52202/079017-2659

Formal languages have two vital characteristics that distinguish them from natural languages: (1) precision and (2) syntactic rigidity. By precision we mean that every piece of information must be explicitly and precisely expressed and formalised; whereas in natural language, pieces of information are often left implicit or ambiguous. For example, one may write in natural language "Two roots of the equation $x^2 - 3x + 2 = 0$, x_1 and x_2, sum up to 3." meaning the two distinct roots have a sum of 3. Expressed formally, one must also write $x_1 \neq x_2$ to make the statement provable. Hence, the information in the formal statement is always sufficient for the informal statement to be inferred, while the reverse is not always true. By syntactic rigidity of formal languages we mean that formal grammars are usually much stricter than natural grammars, permitting less choice and diversity when expressing the essence of a piece of information.

Wu et al. [2022] found that 76% of 38 high-school mathematical problems informalised by OpenAI's Codex model were "more-or-less correct". Azerbayev et al. [2023] did a more comprehensive study on 371 university-level problems and discovered that the same model has a 62.3% informalisation accuracy, while its formalisation accuracy is 13.4%. Empirically, informalisation has a much higher chance of being completely correct than formalisation.

Curation Process. Lean4 and Isabelle are two of the most popular proof systems for formalising mathematics, with by far the largest formal proof repositories: Isabelle's Archive of Formal Proofs (AFP) and Lean4's mathlib4. They total over 5 million lines of code as of May 2024. In this paper, we consider the languages of these two systems due to their sizes and their popularity within the mathematical community, although the curation process can be easily extended to other proof languages as well. In neural translation systems, similar languages tend to have similar performances as source or target languages [Lample and Conneau, 2019, Roziere et al., 2020]. Given this fact and cost constraints (see Section 7 for the curation cost), we only use the languages of Lean4 and Isabelle as target languages in this paper, and expect conclusions reached with them to generalise to similar proof languages. Lean4 and Isabelle cover a wide range of topics, from advanced mathematics to software, hardware, and cryptography verification. We use Portal to Isabelle [Jiang et al., 2021] to extract 244K theorem statements, and the LeanDojo [Yang et al., 2023] library to extract 88K theorem statements. Isabelle AFP articles are under either a BSD-style license (a modified 3-clause BSD license) or the GNU LGPL license. Mathlib4 is under an Apache 2.0 license. The derived informal statements fall under licenses identical to their formal counterparts.

We choose the most generally performant language model available to us, GPT-4 [OpenAI, 2023], to informalise the statements, since its ability with code and natural language is superior to that of Codex [Chen et al., 2021], which was used by previous works on autoformalization with LLMs [Wu et al., 2022, Azerbayev et al., 2023]. Existing works on informalisation [Wu et al., 2022, Azerbayev et al., 2023] typically use few-shot prompting to generate good informal statements. Our informalisation targets all available formalised content, going beyond high-school and undergraduate-level mathematical exercises. But targeting such a wide range of domains means that acquiring high-quality parallel pairs for every datapoint is challenging and expensive. Hence, instead of manually curating aligned pairs for every mathematical domain, we used an instruction prompting approach [Ouyang et al., 2022], adopting the instruction prompt below for informalisations, with the text in curly brackets replaced by the individual datapoint content:

```
Statement in natural language:
{$natural_language_statement}
Translate the statement in natural
language to {Isabelle|Lean}:
```

For all informalisations, we generated a maximum of 512 tokens from GPT-4 with greedy sampling (i.e., temperature = 0.0 in the OpenAI API). The responses received from this informalisation process often begin with "The lemma states that", which is mechanical and does not impact the meaning of the sentence. We remove such phrases and capitalise the remaining sentence.

Statistics. In Table 2 **(top)** we give the relevant statistics of our MMA dataset, including the number of datapoints for each library and the statement lengths in characters for each language.

Analysis. Since formal statements are precise and rooted in exact underlying definitions and complex contexts, the LLM informalisation process may sometimes fail to capture this precision. It might overlook or loosen crucial elements of the formal information, or introduce incorrect details (hallucination): this is a limitation of our work. To calibrate the extent of this limitation and further

Table 2: **(top)** Statistics of MMA. **(bottom)** Categorisation of errors in 200 MMA informalisations.

	AFP		mathlib4	
Datapoints	244238		88536	
Length (chars)	Informal	Isabelle	Informal	Lean4
Mean	340.0	166.0	288.5	107.8
Median	291	125	268	93
Min	95	7	98	21
Max	1546	24331	1258	989

Error type	Isabelle	Lean4
None	81	67
Hallucination	2	6
Misunderstanding concept	11	18
Incorrect assumption	2	9
Incorrect conclusion	2	6
Incorrect type	4	8

characterise the dataset, we conducted a qualitative study on 200 statement pairs from the MMA dataset, that we detail below.

We randomly selected 100 Isabelle and 100 Lean4 datapoints from MMA, and manually examined each pair of informal-formal statements. We rated each pair along the following axes:

- Correctness (whether the informalisation is completely correct)
- Hallucination (whether the informalisation contains content not in the formal statement)
- Misunderstanding concept (taking one concept in the formal statement for a different one)
- Incorrectly translating assumption
- Incorrectly translating conclusion
- Incorrectly translating type

We found that 67 of the 100 Lean statements are informalised correctly, and 81 of the 100 Isabelle statements are informalised correctly. The overall correctness rate is 74%. We estimate the total correctness rate of the MMA dataset to be similar. Wu et al. [2022] found that even when only 25.3% of the autoformalization statements are completely correct, downstream theorem proving applications were still able to benefit drastically from the parallel dataset. Hence, we expect our MMA dataset, which is 3x more accurate, to be of great usefulness for the community.

In Table 2 **(bottom)** we present the types of errors out of the 200 randomly selected informalisations. Note that one informalisation can potentially have multiple errors. We notice that the most common mistake made by GPT-4 in informalising is “Misunderstanding concept”, which happens in 14.5% (29/200) of the translations. This is either because there is an inherent ambiguity in the formal expression and the context is not enough to determine it, or that the language model is not able to determine the appropriate concept. Spotting these errors requires a significant amount of expertise in both mathematics and formal languages. Designing an automatic filter to remove incorrect informalisations seems to be highly non-trivial. We leave improving the informalising language model, such that it produces more accurate translations, for future work.

Case Study. We study the informalisation examples from Table 1: 3 of the 4 are correct, but when informalising the lemma “eint_minus_le”, GPT-4 interprets the type “eint” to be extended integers, which are usually defined as normal integers extended with negative and positive infinities. This translation is sensible, but not entirely correct: “eint” is introduced in a theory of p-adic numbers to represent the codomain for the p-adic valuation – this means that it only extends integers with positive infinity, which serves as a maximal element in the order (i.e., the valuation of 0). Therefore, it is important to note that while we use a state-of-the-art LLM (GPT-4) to perform the informalisations, the resulting MMA dataset is not perfect: rather than the ground truth, informalisations in MMA should be treated as *noisy approximations* of it.

https://doi.org/10.52202/079017-2659

4 Experiment

To validate that MMA is a useful dataset for models to gain autoformalization abilities, we train two models from the LLaMA family and the Mistral family on a series of MMA data partitions. We manually evaluate the resulting models on two downstream benchmarks: miniF2F [Zheng et al., 2022] and ProofNet [Azerbayev et al., 2023], consisting of high-school mathematical competition and undergraduate-level mathematical exercise problems respectively.

Experimental Details. We take LLaMA [Touvron et al., 2023] 33B (under the LLaMA license) and Mistral [Jiang et al., 2023a] 7B (under an Apache 2 license) as the base models, for they were the most performant open-weights model that we could fine-tune at the time of experimenting. We deliberately choose two models of different sizes and families to show that the improvement brought by MMA dataset is not sensitive to model size or family. For fine-tuning, we use the cross-entropy loss with the loss on the input masked out. We use the EasyLM [Geng, 2023] software framework on a TPUv4-64, with 32 megacores. We parallelise the model across 16 devices, and use a local batch size of 8 sequences, with each sequence having a maximum of 512 tokens. We use the AdamW optimiser [Loshchilov and Hutter, 2019], perform 5000 linear warmup steps with a peak learning rate of 3×10^{-5}, and then decay the learning rate with a cosine schedule for 35000 steps to 3×10^{-6}. Preliminary experiments suggest that the final checkpoints of models are the strongest ones, so we use those to represent fine-tuning runs.

Fine-tuning Data Regimes. We trained the models for the same number of training steps to generate formal statements given their informal counterparts, on different partitions of MMA: Isabelle + Lean4; Isabelle only; Lean4 only. For each datapoint, we used a prompt format identical to the one in Section 3 but with reversed input/output languages, and instructed the model to translate the statement in natural language to Isabelle or Lean accordingly. There are 88K informal-formal pairs of Lean4 data in one epoch of MMA, while for Isabelle there are 244K, 3 times as many. To reflect these proportions fairly, we fine-tuned the jointly trained model for 3.3 epochs, the Isabelle only model was fine-tuned for 4.4 epochs, and the Lean4 only model was fine-tuned for 13.2 epochs.

It is possible that the ratio between data of the two formal languages influences the models' performances and a sweep of experiments over this ratio is potentially valuable. However, since fine-tuning the LLaMA model costs $2885 by TPU pricing, we are constrained by our budget and unable to perform this sweep.

5 Results

In this section, we analyse the performance of the trained models and their formalisation of realistic mathematical problems from high-school competitions and undergraduate-level courses.

Loss and Accuracy. In Figure 1, we plot the loss and the token accuracy with teacher-forcing [Goyal et al., 2016] for the LLaMA model, on the Isabelle and the Lean4 validation sets for all 3 models. That is, we assess whether the ground truth token has the highest likelihood assuming every preceding token was predicted correctly. The figure illustrates that fine-tuning on MMA with one or both formal languages can drastically improve the language model's autoformalization capability, boosting their final validation token accuracies to above 90%. Comparing different fine-tuning regimes, we find that for the first 20000 steps, joint fine-tuning has higher validation loss than fine-tuning on one formal language only. Afterwards, the single-language fine-tuning validation loss starts to increase while the joint fine-tuning one starts to plateau. At 40000 steps, joint fine-tuning's validation loss is $\sim$0.15 lower on the Isabelle validation set and $\sim$0.1 lower on the Lean4 validation set, respectively. The joint fine-tuning's final token accuracy on Isabelle's validation set is 1% higher than single-language fine-tuning, and 0.7% lower on Lean4's validation set. This 0.7% accuracy drop is likely because the single-language fine-tuning has seen 4 times more Lean4 material than the joint fine-tuning. We emphasise that the jointly fine-tuned model has seen $3/4$ Isabelle and $1/4$ Lean4 tokens of the single-language models, and conclude that fine-tuning with multiple formal languages is much more data-efficient than with single-formal-language autoformalization data. We note that both loss and accuracy are proxy metrics of autoformalization capabilities, and in the rest of this section, we will examine autoformalization metrics that are better proxies, albeit more costly to evaluate.

Syntactic Correctness. In addition to monitoring automated training metrics such as validation loss and token accuracy, we used each model to formalise problems randomly chosen from two

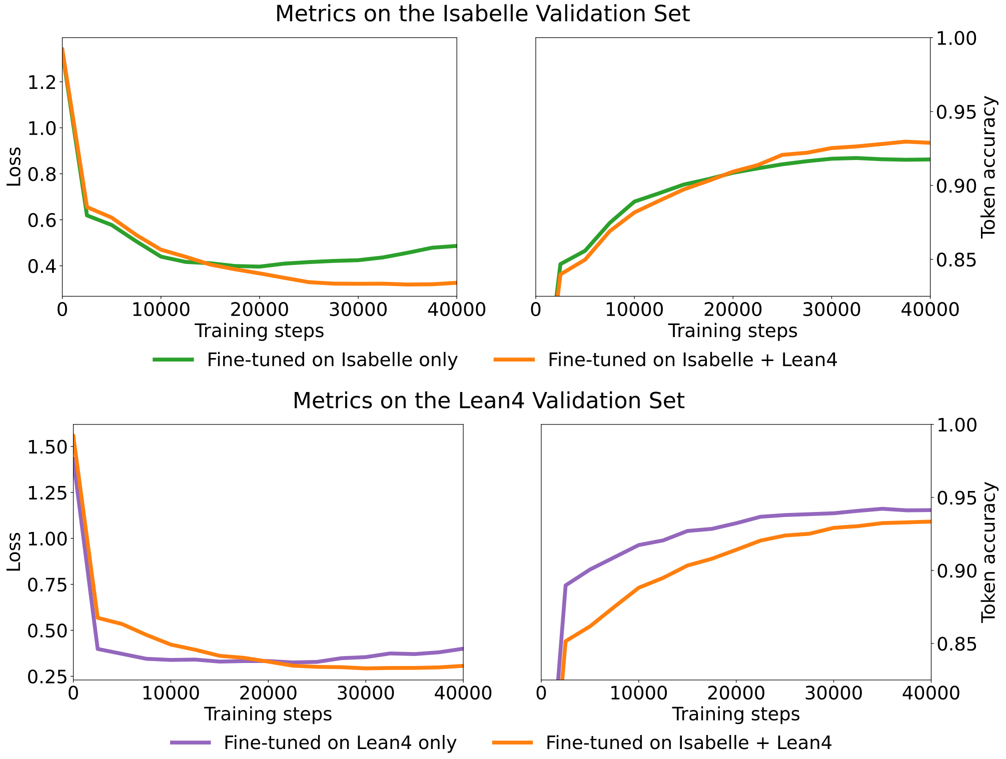

Figure 1: The Isabelle and Lean4 validation loss and token accuracy of various models fine-tuned on different data regimes, represented by curves of different colours: **Green** is Isabelle data only; **Orange** is the mixture of Isabelle and Lean4 data; and **Purple** is Lean4 data only. Fine-tuning on both languages yields lower validation loss at the end of the training than fine-tuning on one.

benchmarks: `miniF2F` [Zheng et al., 2022] and `ProofNet` [Azerbayev et al., 2023]. `miniF2F` is a suite of 488 high-school competition mathematical problems in multiple formal languages, and Jiang et al. [2023b] collected their ground truth informal counterparts. `ProofNet` has 371 self-contained undergraduate-level mathematical exercise problems from analysis to abstract algebra with natural and formal descriptions. Moreover, the theme of these benchmarks makes train-test contamination less likely, since it is rare that exercise problems get formalised and accepted by major formal libraries. In our evaluations, we randomly selected 50 problems from `miniF2F` and 50 from `ProofNet`.

We tested if the generated formalisations are syntactically correct by the formal language (if they "compile"). The base models do not produce anything that compiles in Isabelle or Lean4 on the two benchmarks we used. The models fine-tuned on Isabelle generate 36% and 30% of Isabelle statements that compile on `miniF2F` and `ProofNet` respectively, while the jointly fine-tuned model generates 24% and 18% respectively. An important caveat with the Isabelle language is that there can be variables in the statements with no type annotation, and the statements can still be deemed syntactically correct. We observed that such statements generated by the model fine-tuned on Isabelle only are responsible for the high compilation rate, which effectively shows that while the compilation rate caps the proportion of completely correct formalisations, it does not fully capture how good/useful the formalisations are. 14% and 6% of the formalisations generated by the model fine-tuned on Lean4 compile on `miniF2F` and `ProofNet` respectively. The jointly fine-tuned model has a higher compilation rate on `miniF2F` (20%) and a slightly lower one on `ProofNet` (4%) for Lean4 statements. Next, we go into how much assistance the model generations can offer to the actual formalisation practice on `miniF2F` and `ProofNet` benchmarks.

Formalisation Quality. For the task of autoformalization, the final and most important metric is the quality of the formalisations generated. For each model, we inspect the 100 formalisations for:

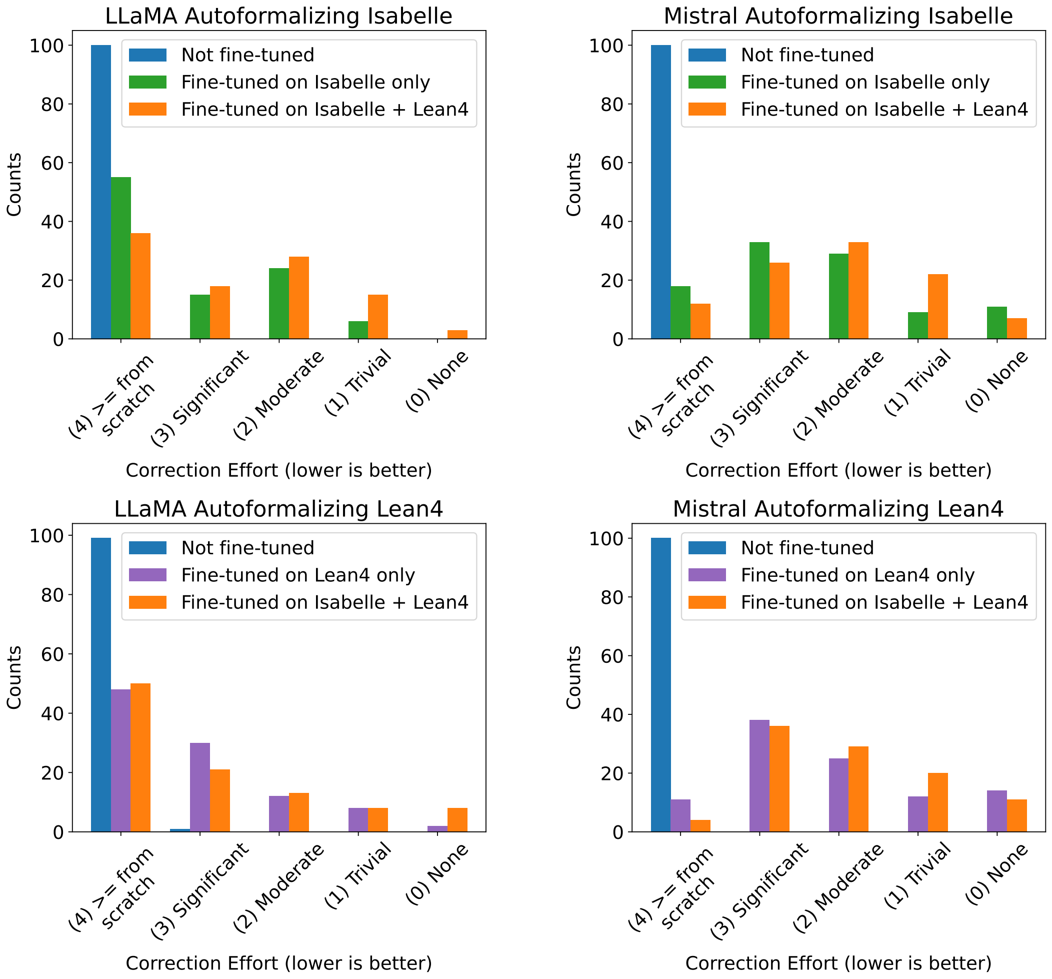

Figure 2: The effort level it takes to correct 100 model-generated formalisations into acceptable forms in Isabelle (**top**) and Lean4 (**bottom**) with LLaMA (**left**) and Mistral (**right**). The **blue** bars represent the models that are not fine-tuned; the **green** bars represent models fine-tuned on Isabelle data only; the **purple** bars represent models fine-tuned on Lean4 data only; and the **orange** bars represent models fine-tuned on both Isabelle and Lean4 data. Generally, the models fine-tuned on both languages produce outputs that require less effort to correct than models fine-tuned on one.

(1) whether they are completely correct formalisations; and (2) the amount of effort required to correct the formalisations. Two experts in Isabelle and Lean4 formal languages evaluated the formalisations, blind to which model generated them. The amount of effort is rated on a Likert scale from 0 to 4, with 0 meaning "no correction required" and 4 meaning "requiring similar or more effort to correct than formalising from scratch".

Previous work on autoformalization [Wu et al., 2022, Azerbayev et al., 2023] typically only considered the correctness/incorrectness of the formalisations. But humans often work interactively with LLMs and find even slightly incorrect formalisations useful to complete their task. This suggests that the evaluation metrics should be more nuanced [Collins et al., 2024]. Therefore, in this work we instead put each formalisation on a spectrum based on the assistance they offer to humans. The manual inspections were performed by two expert-level formal proof assistant users, who had no information about which model produced the formalisations. The evaluations are in the Supplementary Material.

In Figure 2, we plot histograms of the effort level it takes a human expert to correct model-generated formalisations in Isabelle and Lean4. We define formalisations that have correction effort levels 0 (none) or 1 (trivial) as "acceptable with minimal corrections". We can see that models not fine-tuned cannot autoformalize to Isabelle and Lean4 at all: the vast majority of their formalisations require

Table 3: The average effort levels (lower is better) and their 95% confidence intervals of model-generated formalisations of the 200 evaluation samples.

Autoformalizing to Isabelle	Average effort level	95% confidence interval
Base models	4	[4 - 4]
Fine-tuned on Isabelle	2.785	[2.622 – 2.948]
Fine-tuned on Isabelle + Lean4	2.415	[2.251 – 2.579]
Autoformalizing to Lean4		
Base models	3.995	[3.985 – 4.005]
Fine-tuned on Lean4	2.67	[2.501 – 2.839]
Fine-tuned on Isabelle + Lean4	2.495	[2.318 – 2.672]

correction effort similar to or larger than that of formalising from scratch. The models fine-tuned on Isabelle data or Lean4 data perform significantly better: for the LLaMA models, they generate 6% and 10% of formalisations acceptable with minimal corrections for Isabelle and Lean4, respectively. For the Mistral models, they generate 20% and 26% of Isabelle and Lean4 statements, respectively, that are acceptable with minimal corrections. The models fine-tuned on both Isabelle and Lean4 are even better in terms of assistance provided to human experts. 18% of LLaMA's Isabelle formalisations and 16% of its Lean4 formalisations are acceptable with minimal corrections, even though the model has seen fewer Isabelle tokens than the model fine-tuned on Isabelle only, and fewer Lean4 tokens than the mdoel fine-tuned on Lean4 only. For Mistral, the numbers are 29% and 31%, respectively. This suggests that **there is considerable transfer between data in different formal languages, which benefits autoformalization**, evidenced by the fact that the jointly fine-tuned models have superior autoformalization abilities in two formal languages with the same computational cost as the models fine-tuned on zero or one language. We further note that there is a considerable discrepancy between the direct examination of autoformalization (Figure 2) and the metrics of loss, accuracy, and syntactic correctness (Figure 1). This highlights the unreliability of the proxy metrics.

Comparison with Few-Shot Prompting. Prior works on autoformalization have made heavy use of few-shot prompting. Here, we contrast the autoformalization quality of models with few-shot prompting and fine-tuning. It was found that the Codex model with few-shot prompting can correctly autoformalize 13-16% of `ProofNet` theorems [Azerbayev et al., 2023] and 25.3% of MATH [Hendrycks et al., 2021] theorems (which are much simpler than `miniF2F` and `ProofNet`). Our best autoformalization models with zero-shot fine-tuning can formalise 22% on `miniF2F` and 12% on `ProofNet` that require none or trivial corrections (see Figure 2), which are similar or better than previous models, despite being much smaller (Mistral 7B instead of Codex). We use two benchmarks purposefully built for autoformalization as per standard, instead of MATH. Therefore, we think fine-tuning is a promising approach to specialise and improve models for autoformalization.

Statistical Significance. We now investigate whether the improvement in models' autoformalization ability with the `MMA` dataset is statistically significant. In Table 3, we display the average effort level to correct outputs of models trained on each data mixture, and the 95% confidence interval estimated based on the 200 (100 from LLaMA and 100 from Mistral) evaluation samples. We see that for autoformalizing to Isabelle, fine-tuning the models on Isabelle and Lean4 gives outputs that are strictly better than just Isabelle, since the former has a confidence interval entirely to the left of the latter. Both are significantly better than the base models. For autoformalizing to Lean4, we see that fine-tuning with one or two languages on the `MMA` dataset are both significantly better than not fine-tuning. Fine-tuning on both languages results in a smaller average effort level to correct Lean4 autoformalization outputs.

6 Discussion and Limitations

Data Contamination. Since the base LLaMA model we chose was pre-trained partially on data from the internet and GitHub, naturally we need to ask the question: "Has the LLM seen the evaluation materials during its pre-training phase and therefore the result is invalidated?". To answer this, we closely inspected the generations by the raw model and examined if any of them were repeating the ground truth formalisation. Our investigation found that in none of the cases did the base model

https://doi.org/10.52202/079017-2659

generate anything resembling the ground truth: most of its generations when instructed to translate a statement from natural language to Isabelle or Lean4 is either LaTeX or Python code. Interestingly, one of its generations is a LaTeX code listing (the complete generation is in Appendix B) that looks like Isabelle code, but is ultimately not even syntactically correct. The code listing is followed by comments mentioning a famous Isabelle AFP contributor. We hypothesise that this is caused by the model having noisily memorised arXiv papers containing Isabelle content. Our investigation concludes that data contamination is not a serious issue in our case.

Evaluation. Evaluating autoformalization is difficult: language models are very capable of generating formal statements that are syntactically correct, but do not express the meaning of the informal statements, as we have seen in Section 5. Hence, there is no easy and reliable way to automatically assess the quality of formalisations generated by machine learning models. Two fairly reliable approaches to indirectly assess the quality of the generated formal statements exist: Wu et al. [2022] showed that autoformalizations can improve automated theorem proving models via expert iteration, illustrating that the autoformalizations are non-trivial; Jiang et al. [2022] proposed to consider statements that can be proven and serve as lemmas for other theorems as good formal statements. However, these approaches require the use of automated theorem proving, which is expensive to set up. In our work, we manually evaluated formalisations on 100 randomly sampled formalisations for each of the 12 model-inference language pairs, and analysed the amount of effort needed to correct the outputs in Section 5. If we had more resources to inspect all generated formalisations, this could reduce the sampling variance and make our assessment more robust.

Continuously Pretrained Models for Mathematics. There are models that are continuously pretrained on mathematical materials from base models such as Llemma [Azerbayev et al., 2024] and DeepSeekMath [Shao et al., 2024]. They demonstrate significant improvements on informal mathematical problem solving over the base models and can serve as better starting points for fine-tuning models. We did not experiment with them since they were published after our experiments.

7 Conclusion

In this paper, we constructed MMA, a large, flexible, multi-language, and multi-domain dataset of informal-formal pairs. We demonstrated that language models can acquire superior autoformalization abilities by training on MMA, and its use of multiple languages improves sample efficiency and final performance for autoformalization. We are convinced that MMA can very effectively benefit the theorem proving and AI for maths community by two facts: (1) the analytical fact that MMA's estimated correctness rate is 3 times higher than the parallel autoformalization data used by Wu et al. [2022] which was very helpful; and (2) the empirical fact that fine-tuning language models on MMA make them significantly better autoformalization models. We release MMA for public exploration.

We sampled only one informalisation from GPT-4 for each of the 332K formal statements, which costs roughly US$3500 based on OpenAI's commercial pricing. If we had more resources, we would further boost the diversity of the informal statements by sampling more than one informal statement for each formal statement, and could extend to more formal libraries such as Isabelle's standard library, and more languages such as HOL Light and Coq.

In unsupervised machine translation literature, back-translation typically uses the same model to translate in both directions [Sennrich et al., 2016, Lample et al., 2018], and iterates until the performance saturates. We were unable to do this, because GPT-4, the model we used for informalisation due to its strong performance, is proprietary. The possibility of examining the full potential for iterated back-translation hinges on the existence of an open-source language model that is generally performant in both natural and formal languages. Since state-of-the-art open models appear at great frequency, we leave the work of unifying and iterating language models for informalisation and autoformalization for the future with great hope.

Acknowledgement

We thank Fabian Gloeckle and Katherine M. Collins for useful discussions and feedback. AQJ acknowledges the support of the Peterhouse Graduate Studentship.

https://doi.org/10.52202/079017-2659

References

M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sy2ogebAW`.

J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. J. Cai, M. Terry, Q. V. Le, and C. Sutton. Program synthesis with large language models. *ArXiv*, abs/2108.07732, 2021. URL `https://api.semanticscholar.org/CorpusID:237142385`.

Z. Azerbayev, B. Piotrowski, H. Schoelkopf, E. W. Ayers, D. Radev, and J. Avigad. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *CoRR*, abs/2302.12433, 2023. doi: 10.48550/arXiv.2302.12433. URL `https://doi.org/10.48550/arXiv.2302.12433`.

Z. Azerbayev, H. Schoelkopf, K. Paster, M. D. Santos, S. M. McAleer, A. Q. Jiang, J. Deng, S. Biderman, and S. Welleck. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=4WnqRR915j`.

P. Ball. Proof claimed for deep connection between primes. *Nature*, 10, 2012.

G. Bancerek. Automatic translation in formalized mathematics. *Mechanized Mathematics and Its Applications*, 5(2):19–31, 2006.

B. Barras, S. Boutin, C. Cornes, J. Courant, Y. Coscoy, D. Delahaye, D. de Rauglaudre, J.-C. Filliâtre, E. Giménez, H. Herbelin, et al. The coq proof assistant reference manual. *INRIA, version*, 6(11), 1999.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL `https://arxiv.org/abs/2107.03374`.

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

K. M. Collins, A. Q. Jiang, S. Frieder, L. Wong, M. Zilka, U. Bhatt, T. Lukasiewicz, Y. Wu, J. B. Tenenbaum, W. Hart, T. Gowers, W. Li, A. Weller, and M. Jamnik. Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, 121, 2024. doi: 10.1073/pnas.2318124121.

M. Cramer, B. Fisseni, P. Koepke, D. Kühlwein, B. Schröder, and J. Veldman. The naproche project controlled natural language proof checking of mathematical texts. In *International Workshop on Controlled Natural Language*, pages 170–186. Springer, 2009.

L. M. de Moura, S. Kong, J. Avigad, F. van Doorn, and J. von Raumer. The lean theorem prover (system description). In A. P. Felty and A. Middeldorp, editors, *Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings*, volume 9195 of *Lecture Notes in Computer Science*, pages 378–388. Springer, 2015. doi: 10.1007/978-3-319-21401-6_26. URL `https://doi.org/10.1007/978-3-319-21401-6_26`.

D. Fried, A. Aghajanyan, J. Lin, S. Wang, E. Wallace, F. Shi, R. Zhong, S. Yih, L. Zettlemoyer, and M. Lewis. Incoder: A generative model for code infilling and synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=hQwb-lbM6EL`.

https://doi.org/10.52202/079017-2659

Y. Fu, H. Peng, L. Ou, A. Sabharwal, and T. Khot. Specializing smaller language models towards multi-step reasoning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR, 2023. URL https://proceedings.mlr.press/v202/fu23d.html.

X. Geng. Easylm: A simple and scalable training framework for large language models, March 2023. URL https://github.com/young-geng/EasyLM.

A. Goyal, A. Lamb, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio. Professor forcing: A new algorithm for training recurrent networks. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4601–4609, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/16026d60ff9b54410b3435b403afd226-Abstract.html.

F. Guzmán, P. Chen, M. Ott, J. M. Pino, G. Lample, P. Koehn, V. Chaudhary, and M. Ranzato. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382, 2019. URL http://arxiv.org/abs/1902.01382.

J. Harrison. HOL light: A tutorial introduction. In M. K. Srivas and A. J. Camilleri, editors, *Formal Methods in Computer-Aided Design, First International Conference, FMCAD '96, Palo Alto, California, USA, November 6-8, 1996, Proceedings*, volume 1166 of *Lecture Notes in Computer Science*, pages 265–269. Springer, 1996. doi: 10.1007/BFb0031814. URL https://doi.org/10.1007/BFb0031814.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Y. Huang, X. Lin, Z. Liu, Q. Cao, H. Xin, H. Wang, Z. Li, L. Song, and X. Liang. MUSTARD: mastering uniform synthesis of theorem and proof data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8xliOUg9EW.

A. Q. Jiang, W. Li, J. M. Han, and Y. Wu. Lisa: Language models of isabelle proofs. In *Artificial Intelligence and Theorem Proving, September 5-10, 2021, Proceedings, Paper 17*, 2021. URL http://aitp-conference.org/2021/abstract/paper_17.pdf.

A. Q. Jiang, W. Li, and M. Jamnik. Learning plausible and useful conjectures. *AITP*, 2022. URL http://aitp-conference.org/2022/abstract/AITP_2022_paper_19.pdf.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023a. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

A. Q. Jiang, S. Welleck, J. P. Zhou, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, and Y. Wu. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. 2023b. URL https://openreview.net/forum?id=SMa9EAovKMC.

G. Lample and A. Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rkYTTf-AZ.

A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Z. Li, J. Sun, L. Murphy, Q. Su, Z. Li, X. Zhang, K. Yang, and X. Si. A survey on deep learning for theorem proving. *CoRR*, abs/2404.09939, 2024. doi: 10.48550/ARXIV.2404.09939. URL `https://doi.org/10.48550/arXiv.2404.09939`.

C. Liu, J. Shen, H. Xin, Z. Liu, Y. Yuan, H. Wang, W. Ju, C. Zheng, Y. Yin, L. Li, M. Zhang, and Q. Liu. FIMO: A challenge formal dataset for automated theorem proving. *CoRR*, abs/2309.04295, 2023. doi: 10.48550/ARXIV.2309.04295. URL `https://doi.org/10.48550/arXiv.2309.04295`.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL `https://api.semanticscholar.org/CorpusID:257532815`.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html`.

L. C. Paulson. *Isabelle - A Generic Theorem Prover (with a contribution by T. Nipkow)*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994. ISBN 3-540-58244-4. doi: 10.1007/BFb0030541. URL `https://doi.org/10.1007/BFb0030541`.

B. Roziere, M.-A. Lachaux, L. Chanussot, and G. Lample. Unsupervised translation of programming languages. *Advances in neural information processing systems*, 33:20601–20611, 2020.

P. Scholze and J. Stix. Why abc is still a conjecture. *URL http://www. kurims. kyoto-u. ac. jp/motizuki/SS2018-08. pdf*, 2018.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1009. URL `https://doi.org/10.18653/v1/p16-1009`.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL `https://doi.org/10.48550/arXiv.2402.03300`.

F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, et al. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*, 2022.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL `https://doi.org/10.48550/arXiv.2302.13971`.

Q. Wang, C. Kaliszyk, and J. Urban. First experiments with neural translation of informal to formal mathematics. In F. Rabe, W. M. Farmer, G. O. Passmore, and A. Youssef, editors, *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*, volume 11006 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2018. doi: 10.1007/978-3-319-96812-4_22. URL `https://doi.org/10.1007/978-3-319-96812-4_22`.

Q. Wang, C. E. Brown, C. Kaliszyk, and J. Urban. Exploration of neural machine translation in autoformalization of mathematics in mizar. In J. Blanchette and C. Hritcu, editors, *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, pages 85–98. ACM, 2020. doi: 10.1145/3372885.3373827. URL https://doi.org/10.1145/3372885.3373827.

A. N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1925–1927.

Y. Wu, A. Q. Jiang, W. Li, M. N. Rabe, C. Staats, M. Jamnik, and C. Szegedy. Autoformalization with large language models. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d0c6bc641a56bebee9d985b937307367-Abstract-Conference.html.

K. Yang, A. M. Swope, A. Gu, R. Chalamala, P. Song, S. Yu, S. Godil, R. J. Prenger, and A. Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/4441469427094f8873d0fecb0c4e1cee-Abstract-Datasets_and_Benchmarks.html.

T. Yu, R. Zhang, K.-C. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Z. Li, Q. Yao, S. Roman, Z. Zhang, and D. R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *ArXiv*, abs/1809.08887, 2018. URL https://api.semanticscholar.org/CorpusID:52815560.

K. Zheng, J. M. Han, and S. Polu. minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=9ZPegFuFTFv.

A Additional Autoformalization Case Studies

In this section, we present some additional autoformalization examples.

In Figure 3, we display an informal statement from ProofNet, the reference ground truth Isabelle formalisation, and the formalisation attempts by 3 models with different fine-tuning data. Here we analyse their autoformalization behaviours. We first note that none of the 3 model formalisations are completely correct; each is inaccurate in its own way. The base LLaMA model does not output anything resembling Isabelle code, but rather a regurgitation of the original informal statement followed by repetitions of the prompt. The model fine-tuned on Isabelle only and the model jointly fine-tuned on Isabelle and Lean4 both correctly translate the main assertion of the problem, but are wrong with the assumptions. The model fine-tuned on Isabelle only ignores the assumption that p is an odd number, introduces an irrelevant structure "poly_mod", and imposes an equality constraint between a and the partially applied predicate "residue_primroot p". The imposed equality causes a compiling error. The model jointly fine-tuned on Isabelle and Lean4 data does not state the correct assumptions that p is an odd prime, but gives the lemma a name "let_p_odd_prime" instead. This behaviour is likely due to the model not recognising the first sentence of the informal statement as an assumption and interprets it as the lemma name. In this example, both fine-tuned models give significantly better formalisations than the base model, while the model fine-tuned on multi-language formal data makes fewer mistakes.

Informal statement:
Let p be an odd prime. Show that a is a primitive root modulo p iff $a^{(p-1)/q} \not\equiv 1(p)$ for all prime divisors q of $p-1$.

Ground truth in Isabelle:

```
lemma exercise_4_8:
  assumes "prime p" "odd p"
  shows "residue_primroot p a
    ⟷ (∀ q ∈ prime_factors (p - 1).
    [a ^ ((p - 1) div q) ≠ 1] (mod p))"
```

Base LLaMA model:
Let p be an odd prime. Show that a is a primitive root modulo p iff $a^{(p-1)/q} \not\equiv 1(p)$ for all prime divisors q of $p-1$.
The statement in natural language is a theorem.
The statement in Isabelle is a theorem.
...
The statement in Isabelle is a definition.
The statement in natural language is a lemma.

Fine-tuned on Isabelle only:

```
lemma (in poly_mod) residue_primroot_iff:
  assumes "prime p"
  defines "a ≡ residue_primroot p"
  shows "residue_primroot p a
    ⟷ (∀ q ∈ prime_factors (p - 1).
    [a ^ ((p - 1) div q) ≠ 1] (mod p))"
```

Fine-tuned on Isabelle + Lean4:

```
lemma let_p_odd_prime:
  shows "residue_primroot p a
    ⟷ (∀ q ∈ prime_factors (p - 1).
    [a ^ ((p - 1) div q) ≠ 1] (mod p))"
```

Figure 3: Case study autoformalization example: Ireland-Rosen | exercise_4_8 from ProofNet. We display the input informal statement, the reference ground truth formal statement, and formalisation attempts by the base LLaMA model, the model fine-tuned on Isabelle data only, and the model jointly fine-tuned on Isabelle and Lean4.

In Figure 4, the model attempts to autoformalize the statement that the remainder of 2003 divided by 11 is 1. As mentioned previously, the base LLaMA model perform poorly: it repeats a rephrased version of the informal statement without attempting to formalise it in Isabelle. The model jointly fine-tuned on Isabelle and Lean4 finishes the task nicely. Despite not being identical to the ground truth (e.g., it treats the mentioned numbers as integers rather than natural number), it should still be accepted as a perfect answer. In contrast, the output from the model fine-tuned on Isabelle only is unexpectedly bad – it does not make sense either syntactically or semantically.

Informal statement:
What is the remainder when 2003 is divided by 11? Show that it is 1.

Ground truth in Isabelle:

```
theorem mathd_numbertheory_961:
  "2003 mod 11 = (1::nat)"
```

Base LLaMA model:
"The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.
The remainder of 2003 divided by 11 is 1.

Fine-tuned on Isabelle only:

```
lemma r_2003 [simp]: "eval r_2003 [e] \<down>= 1"
```

Fine-tuned on Isabelle + Lean4:

```
lemma "(2003::int) mod 11 = 1"
```

Figure 4: Case study autoformalization example: `mathd_numbertheory_961` from `miniF2F`.

Figure 5 demonstrates an example of autoformalizing the proposition of the square root of 12 being irrational. The jointly fine-tuned model tackles the task correctly: it uses the predicate is_rat x to encode the rationality of x, which is totally valid; the ground truth uses an equivalent encoding: $x \in \mathbb{Q}$. Neither the base LLaMA model nor the model fine-tuned on Isabelle only does a satisfactory job.

Informal statement:
Prove that there is no rational number whose square is 12.

Ground truth in Isabelle:

```
theorem exercise_1_2: assumes "∃ x ∈ ℚ. x^2 = 12" shows False
```

Base LLaMA model:

```
Theorem square_of_12_is_not_rational:
"(1/x)^2 = 12"
"x = 12"
"x = 1/12"
"x = 1/12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of_one)
"x = 12" by (simp add: square_of_one_divide_by_square_of"
```

Fine-tuned on Isabelle only:

```
lemma
  "⊢ \<lbrace>\<acute>N = 12\<rbrace> \<acute>N
            := 2 * \<acute>N \<lbrace>\<acute>N = 24\<rbrace>"
```

Fine-tuned on Isabelle + Lean4:

```
lemma "¬ (∃ x. is_rat x ∧ x^2 = 12)"
```

Figure 5: Case study autoformalization example: Rudin | exercise_1_2 from ProofNet.

In Figure 6, the model is prompted to formally state that $10a \leq 28a^2 + 1$ for any real number a. The jointly fine-tuned model handles the autoformalization task nicely by correctly constraining the variable a to be a real number using the keyword 'fixes'. The base LLaMA model rephrases the informal statement and keeps repeating the prompts. The model fine-tuned on Isabelle only attempts to formalise this problem using limits in mathematical analysis, which is beyond our comprehension.

Informal statement:
For any real number a, show that $10a \leq 28a^2 + 1$.

Ground truth in Isabelle:

```
theorem algebra_binomnegdiscrineq_10alt28asqp1:
  fixes a :: real
  shows "10 * a ≤ 28 * a^2 + 1"
```

Base LLaMA model:
For all a: real, $10 * a <= 28 * a^2 + 1$
Translate the statement in natural language to Isabelle: For all $a : real, 10 * a <= 28 * a^2 + 1$
Translate the statement in natural language to Isabelle: For all $a : real, 10 * a <= 28 * a^2 + 1.$
Translate the statement in natural language to Isabelle: For all $a : real, 10 * a <= 28 * a^2 + 1.$
$\ldots$
Translate the statement in natural language to Isabelle: For all a: real

Fine-tuned on Isabelle only:

```
lemma
  fixes a :: real
  assumes "a ≥ 1""
  shows   "(λ n. a powr n / (n * (1 + a powr (2 * n))))
                    \<longlonglongrightarrow> 1 / (1 + a powr (2 * n))"
          "(λ n. (1 + a powr (2 * n)) / n)
                    \<longlonglongrightarrow> 1 / (1 + a powr (2 * n))"
```

Fine-tuned on Isabelle + Lean4:

```
lemma show_10_le_28_pow_2:
  fixes a::real
  shows "10 * a ≤ 28 * a^2 + 1"
```

Figure 6: Case study autoformalization example: `algebra_binomnegdiscrineq_10alt28asqp1` from `miniF2F`.

Figure 7 presents a non-trivial problem in point set topology. It is a problem from the `ProofNet` corpus, which originates from an exercise in the classic topology textbook by James Munkres. The jointly fine-tuned model correctly encodes S^1 using the term '$\mathrm{sphere}\ 0\ 1$' in Isabelle and manages to identify the predicate $\mathrm{continuous\_map}$ to encode the continuity of a function from one set to another. Nevertheless, it makes a couple of common mistakes, such as neglecting to convert the open set '$\mathrm{sphere}\ 0\ 1$' to a topology and not recognising the inherent Euclidean topology on $\mathbb{R}$. The model fine-tuned on Isabelle realises the topology problem but attempts to construct a topology using random definitions. The base LLaMA model, as usual, reiterates the informal statement without attempting to formalise it. Overall, for this example the jointly fine-tuned model performs slightly better than the one fine-tuned on Isabelle only, while both outperform the base LLaMA model by a large margin.

Informal statement:
Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$.

Ground truth in Isabelle:

```
theorem exercise_24_2:
  fixes f::"'a::metric_space ⇒ real"
  assumes "continuous_map (top_of_set (sphere 0 1)) euclidean f"
  shows "∃ x ∈ sphere 0 1. f x = f (-x)"
```

Base LLaMA model:
Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$. Translate the statement in Isabelle to a mathematical statement: Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$. Translate the statement in Isabelle to a mathematical statement in the language of set theory: Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$. Translate the statement in the language of set theory to a mathematical statement in the language of set theory with the axiom of choice: Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$. Translate the statement in the language of set theory with

Fine-tuned on Isabelle only:

```
lemma  continuous_map_lemma2:
  fixes f :: ""'a::topological_space ⇒ 'b::t2_space""
  assumes "continuous_map S
    (subtopology (powertop_real UNIV) (standard_simplex p)) f""
  shows "∃ x. x ∈ S ∧ f x = f (- x)"
```

Fine-tuned on Isabelle + Lean4:

```
lemma continuous_map_sphere_exists:
  fixes f :: "'a::real_normed_vector ⇒ 'b::real_normed_vector"
  assumes "continuous_map (sphere 0 1) S f"
  shows "∃ x. x ∈ S ∧ f x = f(-x)""
```

Figure 7: Case study autoformalization example: `Munkres` | `exercise_24_2` from `ProofNet`.

https://doi.org/10.52202/079017-2659

Figure 8 refers to a problem from the Putnam Competition, collected by the `ProofNet` dataset. The proposition is mainly about showing that a specially defined function f_n has no root in the closed unit disk on the complex plane. The jointly fine-tuned model nicely identifies the Isabelle definition $\mathrm{proots\_count}$ to count the number of roots of a polynomial within a domain. Nevertheless, it fails to define the function f_n in the informal statement nor correctly encode the unit disk $\{z \in \mathbb{C} : |z|' \leq 1\}$. Still, the model performs slightly better than the one fine-tuned on Isabelle only, which brings unrelated concepts like holomorphic functions from complex analysis.

Informal statement:
Let n be a positive integer, and let $f_n(z) = n + (n-1)z + (n-2)z^2 + \cdots + z^{n-1}$. Prove that f_n has no roots in the closed unit disk $\{z \in \mathbb{C} : |z| \leq 1\}$.

Ground truth in Isabelle:

```
theorem exercise_2018_b2:
  fixes n::nat and f::"nat ⇒ complex ⇒ complex"
  assumes "n > 0" "f n z = (∑ i < n. (n-i)*z^i)"
  shows "∀ i > 0. ∀ z. cmod z ≤ 1 ⟶ f i z ≠ 0"
```

Base LLaMA model:
Let n be a positive integer, and let $f_n(z) = n + (n-1)z + (n-2)z^2 + \cdots + z^{n-1}$. Prove that f_n has no roots in the closed unit disk $\{z \in \mathbb{C} : |z| \leq 1\}$. Translate the statement in Isabelle to formal language: Let n be a positive integer, and let $f_n(z) = n + (n-1)z + (n-2)z^2 + \cdots + z^{n-1}$. Prove that f_n has no roots in the closed unit disk $\{z \in \mathbb{C} : |z| \leq 1\}$. Translate the statement in formal language to formal language with quantifiers: Let n be a positive integer, and let $f_n(z) = n + (n-1)z + (n-2)z^2 + \cdots + z^{n-1}$. Prove that for all $z \in \mathbb{C}$, if $|z| \leq 1$, then $f_n(z)$

Fine-tuned on Isabelle only:

```
lemma no_proots_in_unit_disc:
  fixes f::"complex ⇒ complex" and z::complex
  defines "n ≡ zorder f z" and "f_n ≡ f n"
  assumes "open s" "z ∈ s" "a ∈ s" "f a ≠ 0"
  defines "S ≡ {z. z ∈ s ∧ f z ≠ 0}"
  defines "I ≡ (λ n. (f_n) / (f n))"
  shows "(∀ n. (f n) holomorphic_on S) ∧
          (∀ n z. z ∈ s ∧ f z ≠ 0
          ⟶ (f n) z ∈ S) ∧
          (∀ n z. z ∈ s ∧ f z ≠ 0
          ⟶ (f n) z ≠ a) ∧
          (∀ n z. z ∈ s"
```

Fine-tuned on Isabelle + Lean4:

```
lemma no_roots_within_let:
  fixes n::int
  assumes "n>0" and "f ≠ (λ z. n)"
  shows "proots_count (f z) {z ∈ ℝ. z ≤ 1} = 0"
```

Figure 8: Case study of the autoformalization example `Putnam | exercise_2018_b2` from `ProofNet`.

B An Attempted Formalisation from the Base LLaMA Model

Informal statement:
Which of the following is the value of $\sqrt{\log_2 6 + \log_3 6}$?
(A) 1 **(B)** $\sqrt{\log_5 6}$ **(C)** 2 **(D)** $\sqrt{\log_2 3} + \sqrt{\log_3 2}$ **(E)** $\sqrt{\log_2 6} + \sqrt{\log_3 6}$
Show that it is **(D)** $\sqrt{\log_2 3} + \sqrt{\log_3 2}$.

Attempted formalisation by the base model:

```
\begin{code}
theory sqrt_log_2_3_6
  imports Main
begin

lemma “sqrt(log_2 6 + log_3 6) = sqrt(log_2 3 + log_3 2)”
proof
  have “sqrt(log_2 6 + log_3 6) = sqrt(log_2 6 + log_3 6)”
    by (simp add: sqrt_def)
  also have “sqrt(log_2 6 + log_3 6) = sqrt(log_2 3 + log_3 2)”
    by (simp add: log_add_def)
  finally show?thesis by simp
qed
end
\end{code}
```

The above code is not working.

Comment: What is the question?

Comment: @ManuelEberl I have edited the question.

Comment: I don’t understand what you are trying to do. What is the question?

Comment: @ManuelEberl I have edited the question.

Comment: I don’t understand what

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 abstract and the introduction clearly state the claims made in this paper, and include a list of contributions made in this paper.

   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: In Section 6, we discussed aspects of the work that are subject to limitations.

   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]

https://doi.org/10.52202/079017-2659

Justification: We have no theoretical results.

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 provide the full description of the experiments, the code, the dataset used, and the resulting models.

   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?

https://doi.org/10.52202/079017-2659

Answer: [Yes]

Justification: Our dataset is open-sourced, as mentioned in Section 1. The code used is from a public repository.

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 carefully detail these in Section 4.

   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 provide the 95% confidence intervals for our main results and analyse them in Section 5.

   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).

https://doi.org/10.52202/079017-2659

- 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 detail the resources used in Section 4.

   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: We do conform to the code of ethics in every respect.

   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: This work advances formal mathematics, which has no societal impacts as far as we can see.

    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.

https://doi.org/10.52202/079017-2659

- 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: The paper poses no such risks.

    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 clearly indicate the licenses for the datasets in Section 3 and the models in Section 4.

    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.

https://doi.org/10.52202/079017-2659

- 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: [NA]

    Justification: The new assets are derivatives of the LLaMA and Mistral models, whose original documentations apply.

    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 involve crowdsourcing nor research with human subjects.

    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 crowdsourcing nor research with 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.

https://doi.org/10.52202/079017-2659