Into The Limits of Logic: Alignment Methods for Formal Logical Reasoning

Francisco F. López-Ponce[1,2], Gemma Bel-Enguix [2]
[1]Posgrado en Ciencias e Ingeniería de la Computación - UNAM
[2]Grupo de Ingeniería Lingüística - UNAM
francisco.lopez.ponce@ciencias.unam.mx, gbele@iingen.unam.mx

Abstract

We implement Large Language Model Alignment algorithms to formal logic reasoning tasks involving natural-language (NL) to first-order logic (FOL) translation, formal logic inference, and premise retranslation. These methodologies were implemented using task-specific preference datasets created based on the FOLIO datasets and LLM generations. Alignment was based on DPO, this algorithm was implemented and tested on off-the-shelf and pre-aligned models, showing promising results for higher quality NL-FOL parsing, as well as general alignment strategies[1]. In addition, we introduce a new similarity metric ($LogicSim$) between LLM-generated responses and gold standard values, that measures logic-relevant information such as premise count and overlap between answers and expands evaluation of NL-FOL translation pipelines[2]. Our results show that LLMs still struggle with logical inference, however alignment benefits semantic parsing and retranslation of results from formal logic to natural language.

1 Introduction

Reasoning using a formal logic language is one of the basis of mathematical thinking. Being able to abstract a problem in natural language and express the distinct variables and relationships between them in logical terms helps mitigating ambiguity and unclear relationships. Using these formal representations, a step-by-step inference procedure can be carried out in order to obtain a logically valid conclusion of the presented premises. This type of reasoning is crucial for, not only mathematics but, any discipline in need of an explainable decision making process.

State-of-the-art Large Language Models (LLMs) exhibit human-like reasoning capabilities for various tasks such as coding, general academic examination, and reading comprehension (OpenAI, 2023; Anthropic, 2024). However, formal logic and mathematical reasoning has proven to be an area of expertise where LLMs consistently underperform: the Claude 3 series of models barely reach a 42% Accuracy in the AMC 12 (Mathematical Association of America, 2025), and a 61% on the MATH dataset (Hendrycks et al., 2021), even with such scores Claude outperforms models like GPT 4. Given the connection between logic, mathematics, and human cognitive processes (Yang et al., 2024b), improving an LLM's performance in this area is an open research problem. Not only is this interesting as a stand-alone problem, it has very positive implications for the explainability of LLMs. Having a model that can correctly infer and explain step-by-step said process would benefit most current uses of LLMs.

In this article we focus on an end-to-end inference process divided into three main steps. Given a set of premises in natural language (NL) the first step is a translation of the premises from natural language to first-order logic (FOL). The second step is an LLM-based inference procedure based on the FOL representations. The final step corresponds to a retranslation of the conclusion from FOL to natural language. For each step we implement an LLM Alignment methodology focused on the corresponding task in order to improve a model's performance, as well as a corresponding evaluation.

LLM Alignment methods are post-training strategies that modify a model's internal weights in order to generate text that caters with human selected responses for a wide range of downstream tasks. Alignment strategies, based on reinforcement learning with human feedback (Christiano et al., 2017), were originally implemented for auto-

[1]Datasets and models can be found freely on HuggingFace: https://huggingface.co/Kurosawama

[2]Code can be accessed via this paper's GitHub: https://github.com/Kurocaguama/Into-The-Limits-of-Logic

Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025), pages 112–123

matic summarization (Stiennon et al., 2020). However, recent research uses these strategies to adjust an LLM's behavior to generate safe and useful responses (Ouyang et al., 2022; Bai et al., 2022; Ji et al., 2025). Interestingly, alignment methods have been shown to improve an LLM's performance in tasks outside safety and harmfulness, making them an integral part of post training for newer releases of models (DeepSeek-AI et al., 2025; OpenAI, 2025).

Our approach differs from similar methodologies that work with the same NL-FOL workflow by omitting the dependency on prompting, the use of an external solver, and self-verification. By integrating translation and inference capabilities into a model via alignment, we obtain a robust model capable of solving our problem on its own without the need of external tools. Additionally, we present a novelty metric that measures logical similarity between two sets of premises based on information concerning predicates and logical connectors. This metric allows us to finely evaluate LLMs during the workflow, expanding our focus from the single truth value that NL-FOL workflows are usually evaluated by.

We tested our methodology around the FOLIO dataset (Han et al., 2024), a human created and annotated dataset focused on natural language to first-order logic translation and inference, that is easily adaptable to our three step pipeline. Evaluation was in accordance to FOLIO and step-specific metrics. Our results show that off-the-shelf LLMs can be easily adjusted to become efficient semantic parsers with a limited amount of information, even being able to follow particular prompt-structures aside from the logical benefit. Pre-aligned models perform decently without any logic-based alignment, yet their performance falters after our alignment strategy is applied, suggesting that our methodology needs to be polished in order to work as an additional layer of post-training.

2 Related Work

2.1 LLMs for Logic-Based Reasoning

Early work in this area, such as ProofWriter (Tafjord et al., 2021), tested how well LLMs could reproduce proof generation based on given facts and rules, opting for a generative strategy over classification systems based on pretrained models such as PRover (Saha et al., 2020).

Recent work has covered a wide variety of generative strategies similar to ProofWriter. Pan et al. (2023)'s Logic-LM and Olausson et al. (2023)'s LINC are an example of tool-augmented systems that test an identical end-to-end problem as ourselves. A key difference is that they incorporate the use of an external solver for the inference and retranslation part of the problem, an addition that enables both systems to outperform classic prompting strategies like Chain-of-Thought (Wei et al., 2023) reasoning. Further research has improved tool-augmented systems (Raza and Milic-Frayling, 2025; Kirtania et al., 2024), even surpassing LogicLM's and LINC's results. However, the addition of the external solver reduces the impact of the LLM within the logic pipeline, limiting the influence to the symbolic formalization of the pipeline. Yao et al. (2023) endows an LLM with a search function over possible answers in order to emulate a tree search. This removes the need of the external solver but still requires certain calculations to be carried out independently of the LLM.

Most LLM-only approaches rely on prompting and in-context learning, often modifying the structure of the prompt to carry out procedures like abstracting, formalizing, and explaining before answering (Ranaldi et al., 2025), formulating the solution as a Python function (Lyu et al., 2023), or giving the prompt a task-specific solution structure (Zhou et al., 2024). An extension of these approaches that is highly similar to our work, is the use of LLM Alignment methods in order to optimize a task-oriented Supervised Fine-Tuning (Ranaldi and Freitas, 2024). A slight difference however, is that this work focuses on general reasoning rather than formal logic.

2.2 Alignment Methodologies

Initial works in LLM Alignment taught models how to summarize texts (Stiennon et al., 2020) and how to behave in terms of safety and usefulness (Ouyang et al., 2022) using Supervised Fine-Tuning (SFT) to obtain a reference model for sampling, and PPO (Schulman et al., 2017) as a training strategy with another LLM. However, PPO is an online algorithm that uses two LLMs during training, meaning that it's a resource heavy option that now serves more as a baseline comparison rather than a widely used strategy. Alternative algorithms have been developed that solve this problem, particularly Direct Preference Optimization (DPO) (Rafailov et al., 2023). This algorithm avoids fine-tuning a reference model as well as the sampling from

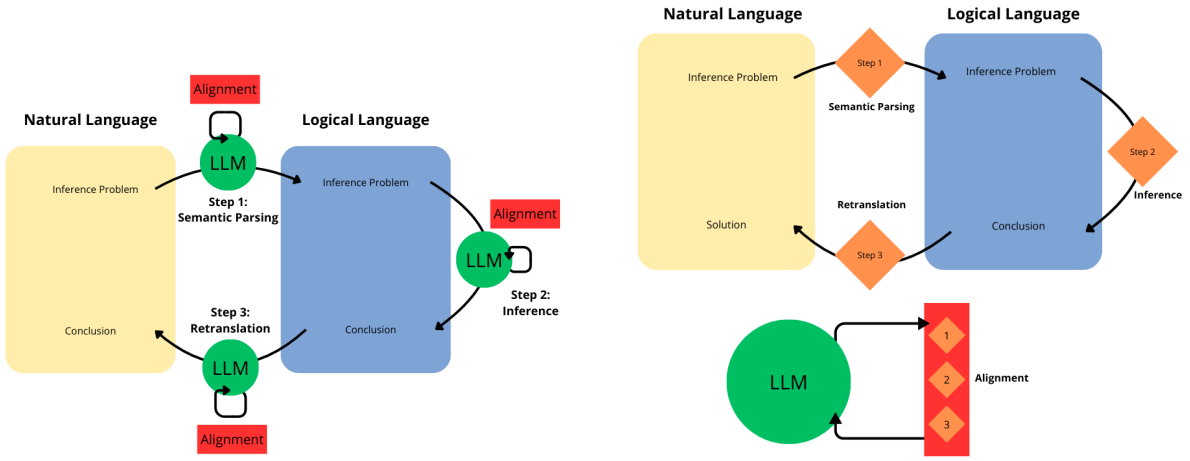

Figure 1: Step-Independent Alignment (left) and Mixture-of-Steps Alignment (right).

said model, it does so by working on a preference dataset that compares pairs of responses to a given prompt. DPO is a widely used algorithm for alignment in state-of-the-art LLMs (Grattafiori et al., 2024).

Further research regarding alignment algorithms has been carried out. The Group Relative Policy Optimization algorithm (GRPO) (Shao et al., 2024) is a memory-optimized version of PPO that avoids the reference model, and instead shows explicitly which responses are preferred by giving a prompt and a "correct" completion of said prompt.

Mathematically speaking, each strategy has its strengths and weaknesses that make it an adequate candidate for alignment (Xu et al., 2024). However we can't just rely solely on mathematical intuition given that we're working with text data as well as a concrete problem that can't be solely encapsulated with loss functions. For said reason we opt for DPO due to the nature of preference datasets used during training. This type of datasets enables a model to compare different answers, analyze differences between them, and consider the preference score in order to adequately adjust the model's response. In logical and mathematical reasoning there isn't always a single correct way to derive a proof, or to semantically parse a set of premises. We believe that by having a dynamic set of scores that measure correctness, a model should be able to efficiently adapt to varying sets of premises, inference steps, and lexical themes.

2.3 Logic and Mathematics Evaluation

LLM logical evaluation has a plethora of datasets to work with. However, not every dataset centered on logical reasoning actually evaluates formal logical language reasoning. Datasets like LogiQA (Liu et al., 2020) and LogicNLI (Tian et al., 2021) are good resources for natural language reasoning, however there's no logical formalization of the premises and answers within those datasets.

Newer benchmarks such as FOLIO (Han et al., 2024), LogicBench (Parmar et al., 2024), or MALLS (Yang et al., 2024a), deal with the formalization of the premises in first-order logic. In particular, FOLIO is an expert-written and human-reviewed dataset (contrary to the other two), that covers each step of an end-to-end pipeline for logic-inference.

3 Experiments

In this section we describe our end-to-end workflow with a working example, the creation of the preference datasets used for alignment, selected LLM checkpoints, and experimental setup.

3.1 Problem Definition

As previously mentioned, our inference procedure is divided into three separate steps, each evaluated independently to determine weak points throughout the end-to-end workflow.

The first step of our three step inference process is the **translation** of a set of premises from natural language to first order logic. This task is also referred to as semantic parsing, and is shared across many logic-based reasoning systems (Pan et al., 2023; Olausson et al., 2023; Raza and Milic-Frayling, 2025). The second step is the **inference** procedure based solely on the premises in their FOL syntax. The LLM should use logically valid

inference steps[3] in order to obtain a unique conclusion, expressed in FOL syntax, to the problem formulation. The third and final step corresponds to a **retranslation** of the conclusion to natural language. An example of the whole process can be seen in table 1. The example is extracted from the validation set of the FOLIO dataset.

3.2 Alignment

In order to train an LLM in each task an alignment strategy based on the corresponding step is implemented. Two variations of the alignment procedure are possible: *Step-Independent Alignment* (training with only a single step of the pipeline) and *Mixture-of-Steps Alignment* (training using the full pipeline). Step-Independent Alignment generates three distinctly aligned models, one for each corresponding step. On the other hand, Mixture-of-Steps returns a single model aligned to all of the steps. Figure 1 shows a diagram of the end-to-end inference process and the steps where alignment is performed. Due to the sparse amount of data available for training (1000 data instances at most per step), Step-Independent Alignment is not carried out in our experiments. The implementations talked about in the remainder of the paper describe training and performance using the Mixture-of-Steps methodology.

We implement alignment based on DPO (Rafailov et al., 2023) due to the advantages this algorithm presents over classic SFT + RLHF approaches (Ouyang et al., 2022), particularly in regards with the datasets needed implement the algorithm. DPO uses a preference dataset for training, each entry is comprised of four columns, two of which contain chosen and rejected generations, and two that contain a score that measures a numerical preference over each pair of responses. The chosen and rejected columns share the same input prompt, it's the LLM response that varies.

We created a preference dataset for each step of the end-to-end procedure. Each dataset is model specific meaning that the responses of one model's checkpoint don't affect the behavior of other models. This allows us to evaluate various aspects of LLM behavior such as how susceptible each individual model is to our methodology, how much of an improvement is shown between aligned and vanilla models, parameter size dependency and more. This segmentation between steps generates three datasets per LLM checkpoint used, however combining them into a unique dataset is a straight forward procedure.

3.3 Preference Dataset Creation

To create a single entry of the dataset we consider a prompt x, based on which we need to obtain two answers, y_1, y_2 (chosen and rejected), and two scores, s_1, s_2, that serve as preferences for the LLM. The chosen column contains the pair (x, y_1), while the rejected column switches the response, containing the pair (x, y_2). Our datasets' chosen inputs are always extracted directly from the FOLIO dataset meaning they're human generated, on the other hand, rejected inputs are always LLM-generated. However, not every instance of FOLIO is considered, particularly for the inference and retranslation datasets. FOLIO is comprised of sets of premises and conclusions, and a corresponding truth value (True, False, Uncertain) for any conclusion. Those tagged as False or Uncertain are not taken into consideration.

Preference scores are balanced depending on similarity between both texts. Given the high quality annotations used in FOLIO, we believe considering said answer as gold standards gives us better aligned models with an objective ground truth.

Formally, consider an LLM checkpoint C, in order to generate the $i-$th entry of the translation dataset (the procedure is that same for any step of the logic procedure) we ask C to carry out said step on the $i-$th entry of FOLIO to obtain an LLM-generated answer. This synthetic answer is compared with the actual $i-$th entry of the dataset and measured in terms of semantic similarity. Based on this similarity score the chosen and rejected values s_1 and s_2 are calculated, initial chosen scores are initialized at 8, while rejected scores at 4. The semantic similarity measure modifies these scores allowing a max range of 8.5 and 3.5, and a minimum similarity score of 7.5 and 4.5.

We opted for a one-shot prompting approach for each step of the inference procedure. Most LLM literature evaluates formal logic reasoning using multi-shot prompting (Grattafiori et al., 2024; OpenAI, 2023; Anthropic, 2024; Han et al., 2024), however we're interested in measuring the effects of alignment strategies over prompting, hence the selection of one-shot prompting. The full prompts can be accessed via this paper's repository.

[3]The same ones used for the FOLIO dataset.

NL Premises	FOL Premises	FOL Conclusion	NL Conclusion
·Robert Lewandowski is a striker. ·Strikers are soccer players. ·Robert Lewandowski left Bayern Munchen. ·If a player leaves a team they no longer play for that team.	Striker($robertLewadowski$) $\forall x$(Striker(x) →SoccerPlayer(x)) Left($robertLewandowski, bayernMunchen$) $\forall x \forall y$(Left(x, y) → ¬PlaysFor(x, y))	SoccerPlayer($robertLewandowski$)	Robert Lewandowski is a soccer player.

Table 1: Extracted example from FOLIO.

3.4 Evaluation Methods

We compare an LLM-generated answer with a gold standard extracted from the FOLIO dataset, based on certain similarities we determine if the model's response is adequate. This methodology measures quality across each step and not only analyzes the final logical value of each sample, thus expanding NL-FOL translation metrics and evaluation systems.

For each step we measure a similarity value that considers consistency of individual premises, functions, variables, logical connectors and operations, as well as total amount of predicates. If there's a sufficient similarity based on these values, we tag the pair as adequate and carry out a manual revision to further analyze results. Reported scores are based on these tags. Our metric is defined as follows:

Let (x, y) be a pair of answers the same inference step, x extracted from FOLIO, y generated by an LLM. We denote the **differences in total amount of premises** between x, y as pd, the **differences in distinct amount of predicates** as ap, the **differences in total predicates appearances** as tp, **differences in logical operators** as ld and the **intersection over union of predicates** as IoU. Our metric is defined as:

$$LogicSim(x, y) = pd + ap + tp + ld + IoU \quad (1)$$

This metric operates over atomic sentences (sentences in the form of Predicate($v_1, ..., v_n$)), each separated using regular expressions. Final scores are normalized. Figure 2 shows an example of the metric, FOLIO's correct text is the same as the FOL Premise in table 1, the LLM answer is generated by LLAMA-3.1-8B.

Since this metric operates over FOL, the final step of the pipeline is evaluated using dense semantic similarity.

3.5 LLMs

A total of 5 LLMs were tested, table 2 shows the full list. We test both -INSTRUCT (referred to as *pre-aligned models*) and vanilla models in order to measure the impact of previous general purpose alignment in our experiments. Due to insufficient computing power, larger checkpoints weren't tested. Checkpoints obtained using our methodology will be referred to as *logic-aligned models* or -LA models due to their checkpoint name.

Model Series	Checkpoint
Llama 3	`Llama-3.1-8B`
Llama 3	`Llama-3.1-8B-Instruct`
LLama 3	`Llama-3.2-3B`
Llama 3	`Llama-3.2-3B-Instruct`
Gemma 3	`google/gemma-3-1b-it`

Table 2: List of LLM checkpoints used during testing.

Training, generation, and evaluation was carried out in a single A5000 GPU. The creation of single step dataset averaged 2:10 hours, alignment averaged around 16 minutes using TRL's (von Werra et al., 2020) implementation of DPO. Details regarding hyperparameters can be found in Appendix B as well as this paper's repository.

4 Results and Evaluation

Table 3 shows the average $LogicSim$ score of each model during the first two steps. The third step isn't evaluated using this metric since the answers themselves are in NL, while our metric only evaluates FOL premises. Table 4 shows the average semantic similarity between gold standards and retranslated premises between checkpoints.

5 Discussion and Future Work

We first discuss model-agnostic results, afterwards vanilla models and their logic-aligned counterpart,

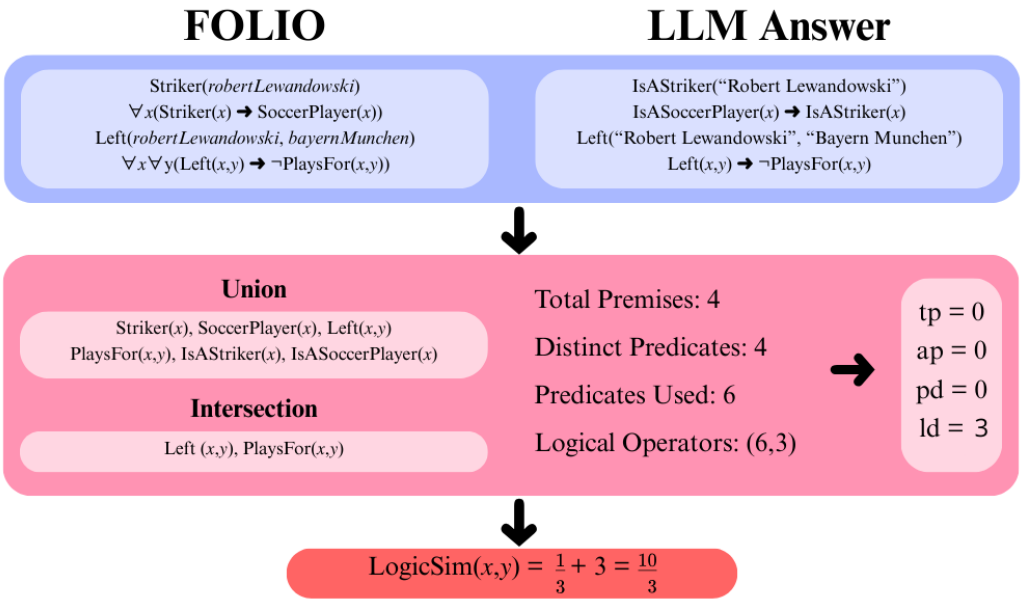

Figure 2: LogicSim(x, y) between a gold standard and an LLM answer.

Checkpoint	Translation	Inference
LLAMA-3.1	(20.4, 20.7)	(47, 45.9)
LLAMA-3.1-INST	(22.8, 22.9)	(**58.5**, **59.8**)
LLAMA-3.2	(24.4, 23.1)	(47.5, 48.8)
LLAMA-3.2-INST	(25.8, **25.6**)	(56.6, 53.6)
GEMMA-3-1B-IT	(**29.1**, 30.8)	(46.7, 47.9)

Table 3: The ordered pairs indicate scores for -LA aligned models, and base checkpoints, in said order. e.g. (20.4, 20.7) indicates that the -LA model scored 20.4, while the base checkpoint 20.7.

Checkpoint	Retranslation
LLAMA-3.1	(0.38, 0.22)
LLAMA-3.1-INSTRUCT	(0.32, .33)
LLAMA-3.2	(0.27, 0.21)
LLAMA-3.2-INSTRUCT	(**0.43**, **0.39**)
GEMMA-3-1B-IT	(0.23, 0.27)

Table 4: Average semantic similarity between a checkpoint's retranslation step.

to finish with the pre-aligned models. Select tables can be found in the Appendix C.

5.1 Model-Agnostic

Models perform best in step 1, however they struggle particularly when having to use multiple predicates as well as functions that operate over various variables. As an example consider premise 4 from table 1, that stand alone premise is associated with the following FOL formulation:

$$\forall x \forall y(Left(x, y) \rightarrow \neg PlaysFor(x, y)) \quad (2)$$

This formulation uses two universal quantifies, over two variables x, y that correspond to a soccer player and a team. In general, vanilla, pre-aligned, and -LA models are unable to formulate a translation that takes into account the second variable (the team) in order to correctly generalize the formulation (see tables 7 and 8). At best, some translations are either vague enough that they make sense on their own, or are closely related to the premise but don't represent the same logical formulation. LLAMA-3.1-8B-INSTRUCT-LA translates this premise in a contrapositive manner (Table 7), indeed if a player x plays for a club y, it implies that x has **not** left club y, while that is logically equivalent it is an incorrect translation.

Additionally, -LA models generate longer responses in comparison with gold standards, table 9 shows that LLM-generations can be over 1000 characters longer than the answers extracted from the dataset. This is a notorious problem in steps 2 and 3 since the expected answer is a single logical conclusion, however, LLMs tend to regurgitate information presented in either the prompt or the alignment datasets.

While the semantic parsing task of steps 1 and 3 shows promising improvement, logical inference remains a challenge for all models. Most inference steps where poorly executed, vanilla models were

too susceptible to following the prompt structure even in cases when the answer was comprised of a single premise.

5.2 Vanilla Models

Logic-aligned models improve generation structure throughout the full procedure, noteworthy improvement can be seen in steps 1 and 3. However structure improvement doesn't translate to semantic parsing correctness.

Vanilla models are highly inconsistent when recreating any step of the pipeline, their generations barely have overlapping lexicon with the gold standard, premises are not separated in any manner, responses are quick to degenerate, and reasoning is conspicuous by its absence. In contrast, logic-aligned models are able to separate and parse premises, create complex predicates with adequate use of variables, and explain the reasoning behind such predicates.

This improvement in performance and parsing-quality is heavily tied to our specific problem, as well as prompt structure used during alignment. The prompts follow a one-shot structure (see Appendix A), in particular for steps 1 and 3 the single example contains context regarding logical symbols as well as a precise comment on each premise and predicate (marked by the three consecutive colons). This is the most notable pet phrase adapted by the logic-aligned models. Similarly, the prompt separates the problem, the predicates, and premises (in that order), making the model highly susceptible to generating answers in the same format disregarding logical-veracity.

However, even with general improvements in parsing structure and the use of logical connectors, -LA models struggle to remain consistent during parsing and to incorporate world-knowledge into this task. Consider the example shown in tables 7 and 8, NL premises are extracted from the dataset, the corresponding FOL premises were translated by the LLAMA-3.1-8B-LA checkpoint. The model fails to realize that the function $Left(x)$ can't be used as a variable, $RobertLewandowski$ should be a constant rather than a function, and that Bayern Munchen is not represented as a constant.

5.3 Pre-Aligned Models

Pre-aligned models have better baselines in terms of style, structure, and general problem solving capabilities. Even in with a one-shot style of prompting, these models surpass Vanilla + -LA checkpoints throughout the pipeline.

Problems like the one discussed in the previous subsection aren't as notorious with pre-aligned models. However, a different problem is encountered with these models: the variation of FOL expressiveness. As an example consider the predicate RankedHighlyBy(x, womensTennisAssociation), this is shortened to RankedHigh(x) by -INSTRUCT LLMs, and is specified (in natural language) that the institution doing the ranking is the Women's Tennis Association. Problems like these happen in particular with pre-aligned models and require manual revision of the experiments.

5.4 Future Work

The alignment datasets could improve substantially. Dataset-wise the prompts used for alignment varied only in the test example, the linguistic structure of the prompt, as well as the one-shot example remained the same. Adding variations in structure such as zero-shot and multi-shot examples, a varied lexicon and different training examples, as well as more diverse preference scores would improve the robustness of the system.

With regards to training data, increasing the size of the alignment dataset, either by combining our datasets with general purpose alignment, or by increasing the amount of formal logic reasoning examples is an avenue of research that might help improve performance in the end-to-end inference task. This could enable a more robust implementation of single-step alignment.

RL-wise, implementing different alignment algorithms like GRPO, as well as expanding this problem to a multi-objective optimization case could be beneficial for further experiments. The three step pipeline easily adapts into multi-objective scenarios like those proposed in Panacea (Zhong et al., 2024) and AMoPO (Liu et al., 2025), this approach reduces the amount of models needed to be evaluated and makes use of all of the previously created datasets. A parallel approach would be to incorporate multi-shot examples during training, this would harness the best of alignment and prompting strategies.

Limitations

LLM alignment drastically improves a model's generative capabilities for a given task, however the fundamental workings of the LLM remain the same.

Our methodology enables LLMs to mimic certain aspects of formal logic reasoning, however incorporating real world knowledge into their mimicking is a limiting aspect of the methodology.

Even if a model is capable of solving tasks like the one evaluated in this paper, it does not mean that the problem of mathematical thinking and abstraction are solved. LLMs are still stochastic by nature and leveraging the generation probabilities of formal logic tokens to mimic rational thinking and abstraction is not the same as actual rational thinking and abstraction.

Ethical Considerations

Our work aims at giving an LLM abstraction capabilities over natural language, however these models are still susceptible to biases inherent from their training data, adding a logical layer of processing to an LLM doesn't make this problem disappear. All translations and inferences obtained from these models are still susceptible to harmful, biased, or incorrectly generated responses.

Acknowledgments

This work is funded by SECIHTI, CVU number 2045472, and PAPIIT projects IG400325 and IN104424. Special thanks to Dr. Helena Gómez-Adorno for the use of her GPU cluster, and Dr. Carlos Hernández Castellanos for the insightful discussions about this project.

References

Anthropic. 2024. Introducing the next generation of claude.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. Folio: Natural language reasoning with first-order logic. *Preprint*, arXiv:2209.00840.

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

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, Sirui Han, Yike Guo, and Yaodong Yang. 2025. PKU-SafeRLHF: Towards multi-level safety alignment for LLMs with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31983–32016, Vienna, Austria. Association for Computational Linguistics.

Shashank Kirtania, Priyanshu Gupta, and Arjun Radhakrishna. 2024. LOGIC-LM++: Multi-step refinement for symbolic formulations. In *Proceedings of the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ACL 2024)*, pages 56–63, Bangkok, Thailand. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *Preprint*, arXiv:2007.08124.

Qi Liu, Jingqing Ruan, Hao Li, Haodong Zhao, Desheng Wang, Jiansong Chen, Wan Guanglu, Xunliang Cai, Zhi Zheng, and Tong Xu. 2025. AMoPO: Adaptive multi-objective preference optimization without reward models and reference models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8832–8866, Vienna, Austria. Association for Computational Linguistics.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and

Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

MAA Mathematical Association of America. 2025. America mathematics competition.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

OpenAI. 2025. Gpt-5 system card. `https://openai.com/index/gpt-5-system-card/`. Accessed August 8, 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Towards systematic evaluation of logical reasoning ability of large language models. *arXiv preprint arXiv:2404.15522*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Leonardo Ranaldi and Andre Freitas. 2024. Self-refine instruction-tuning for aligning reasoning in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2347, Miami, Florida, USA. Association for Computational Linguistics.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17222–17240, Vienna, Austria. Association for Computational Linguistics.

Mohammad Raza and Natasa Milic-Frayling. 2025. Instantiation-based formalization of logical reasoning tasks using language models and logical solvers. *CoRR*, abs/2501.16961.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136, Online. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *Preprint*, arXiv:2404.10719.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2024a. Harnessing the power of large language models for natural language to first-order logic translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6942–6959, Bangkok, Thailand. Association for Computational Linguistics.

Zonglin Yang, Xinya Du, Rui Mao, Jinjie Ni, and Erik Cambria. 2024b. Logical reasoning over natural language as knowledge representation: A survey. *Preprint*, arXiv:2303.12023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. Panacea: Pareto alignment via preference adaptation for llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 75522–75558. Curran Associates, Inc.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Preprint*, arXiv:2402.03620.

A Prompts

The prompts used for training followed the same structure as Logic-LM. An example can be seen below:

Given a problem description and a question, the task is to parse the problem and the question into first-order logic formulars. The grammar of the first-order logic formular is defined as follows:

1) *logical disjunction of expr1 and expr2: expr1* $\vee$ *expr2*
2) *logical disjunction of expr1 and expr2: expr1* $\vee$ *expr2*
3) *logical exclusive disjunction of expr1 and expr2: expr1* $\oplus$ *expr2*
4) *logical negation of expr1:* $\neg$*expr1*
5) *expr1 implies expr2: expr1* $\rightarrow$ *expr2*
6) *expr1 if and only if expr2: expr1* $\leftrightarrow$ *expr2*
7) *logical universal quantification:* $\forall x$
8) *logical existential quantification:* $\exists x$

Problem:

All people who regularly drink coffee are dependent on caffeine. People either regularly drink coffee or joke about being addicted to caffeine. No one who jokes about being addicted to caffeine is unaware that caffeine is a drug. Rina is either a student and unaware that caffeine is a drug, or neither a student nor unaware that caffeine is a drug. If Rina is not a person dependent on caffeine and a student, then Rina is either a person dependent on caffeine and a student, or neither a person dependent on caffeine nor a student.

Predicates:

Dependent(x) ::: x is a person dependent on caffeine. Drinks(x) ::: x regularly drinks coffee. Jokes(x) ::: x jokes about being addicted to caffeine. Unaware(x) ::: x is unaware that caffeine is a drug. Student(x) ::: x is a student.

Premises:

- $\forall x$ *(Drinks(x)* $\rightarrow$ *Dependent(x)) ::: All people who regularly drink coffee are dependent on caffeine.*
- $\forall x$ *(Drinks(x)* $\oplus$ *Jokes(x)) ::: People either regularly drink coffee or joke about being addicted to caffeine.*

- $\forall x\,(Jokes(x) \rightarrow \neg Unaware(x))$ *::: No one who jokes about being addicted to caffeine is unaware that caffeine is a drug.*
- $(Student(rina) \land Unaware(rina)) \oplus \neg(Student(rina) \lor Unaware(rina))$ *::: Rina is either a student and unaware that caffeine is a drug, or neither a student nor unaware that caffeine is a drug.*
- $\neg(Dependent(rina) \land Student(rina)) \rightarrow (Dependent(rina) \land Student(rina)) \oplus \neg(Dependent(rina) \lor Student(rina))$ *::: If Rina is not a person dependent on caffeine and a student, then Rina is either a person dependent on caffeine and a student, or neither a person dependent on caffeine nor a student.*

Problem:
{}
Predicates:
For further details please review the paper's repository.

B Alignment Hyperparameters

Hyperparameter	Value
Generation Strategies	Greedy Search
Max_Tokens	150
Quantization	4Bit, torch.bfloat16 for every model

Table 5: LLM Generation Hyperparameters

Hyperparameter	Value
Lora r	8
α	0.32
Dropout	0.1
Epochs	3

Table 6: Alignment Hyperparameters

C Results

Tables mentioned in the article displayed in this section.

Llama-3.1-8B-LA	Llama-3.1-8B-Instruct-LA
$\forall x(Striker(x) \rightarrow Player(x))$ $\forall x(Left(x) \rightarrow \neg Player(x))$ $RobertLewandowski(Left)$	$Striker(robert\_lewandowski)$ $SoccerPlayer(robert\_lewandowski) \rightarrow Striker(robert\_lewandowski)$ $Left(robert\_lewandowski, bayern\_munchen)$ $PlayFor(x, y) \rightarrow \neg Left(x, y)$

Table 7: Translations by model (Part 1).

Llama-3.2-3B-LA	Gemma-3-1b-it-LA
$IsStriker(robert)$ $IsSoccerPlayer(robert)$ $IsLeft(bayern, robert)$ $IsPlayer(robert, bayern) \oplus \neg IsPlayer(robert, bayern)$	$Striker(RobertLewandowski)$ $SoccerPlayer(RobertLewandowski)$ $Left(RobertLewandowski)$

Table 8: Translations by model (Part 2).

Checkpoint	Translation	Inference	Retranslation
LLAMA 3.1-8B-LA	1080	815	977
LLAMA 3.1-8B-INSTRUCT-LA	1007	753	966
LLAMA 3.2-3B-LA	1016	820	1004
LLAMA 3.2-3B-INSTRUCT-LA	1092	870	1080
GEMMA-3-1B-IT-LA	1000	736	790
FOLIO	**280**	**28**	**32**

Table 9: Average answer length (in characters) for -LA models.