FIRMA: Bidirectional Formal-Informal Mathematical Language Alignment with Proof-Theoretic Grounding

Maryam Fatima
Independent
s.m.fatima13@gmail.com

Abstract

While large language models excel at generating plausible mathematical text, they often produce subtly incorrect formal translations that violate proof-theoretic constraints. We present FIRMA (Formal-Informal Reasoning in Mathematical Alignment), a bidirectional translation system between formal and informal mathematical language that leverages proof-theoretic interpretability hierarchies and specialized architectural components for proof preservation. Unlike existing approaches that treat this as pure sequence-to-sequence translation, FIRMA introduces a hierarchical architecture with complexity-aware routing, proof-preserving attention mechanisms, and multi-objective training that balances formal correctness with natural readability. Through progressive complexity training on curated datasets from Lean 4 and formal mathematics repositories, we evaluate FIRMA on 200 translation samples across complexity levels and compare against two baseline systems. Our analysis shows statistically significant improvements of 277.8% over BFS-Prover-V1-7B and 6307.5% over REAL-Prover on overall translation quality metrics. Ablation studies on 50 samples demonstrate that each architectural component contributes substantially to performance, with removal of any component resulting in 83-85% performance degradation. We release our code at https://github.com/smfatima3/FIRMA

1 Introduction

Mathematical communication exists on a spectrum from the highly formal languages of proof assistants like Lean and Coq to the intuitive explanations found in textbooks. This duality creates a fundamental challenge: formal specifications ensure logical rigor but are often impenetrable to students, while informal descriptions aid understanding but may harbor subtle errors or ambiguities. The ability to translate bidirectionally between these representations would benefit both mathematical education and formal verification efforts.

Recent advances in large language models (LLMs) have shown capabilities in mathematical reasoning. The development of systems like AlphaGeometry (Trinh et al., 2024) and FunSearch (Romera-Paredes et al., 2024) demonstrates that neural approaches can achieve results on challenging mathematical problems, including discovering new theorems and solving Olympiad-level geometry problems. However, standard mathematical reasoning benchmarks like the MATH dataset (Hendrycks et al., 2021) reveal limitations when models attempt formal-informal translation tasks.

When applied to formal-informal translation, these models exhibit critical limitations. They generate superficially plausible translations that fail proof checking, introduce logical errors through imprecise natural language, and show unpredictable degradation as mathematical complexity increases. Early attempts at neural theorem proving like NaturalProver (Welleck et al., 2021) and more recent work on whole-proof generation (First et al., 2023) have made progress, but these failures stem from treating mathematical translation as a purely linguistic task, ignoring the underlying proof-theoretic structure that governs mathematical validity.

We introduce FIRMA (Formal-Informal Reasoning in Mathematical Alignment), a framework that grounds mathematical translation in proof-theoretic principles. Our key insight is that successful translation requires not just linguistic fluency but also preservation of logical structure across complexity levels. FIRMA addresses this through three core innovations:

First, we develop a hierarchical encoder-decoder architecture that explicitly models mathematical complexity through specialized routing mechanisms. Unlike flat sequence models, FIRMA processes mathematical statements at multiple levels

Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025), pages 62–76

of abstraction—symbols, syntax trees, and semantic structures—enabling it to maintain logical coherence while adapting linguistic style.

Second, we introduce proof-preserving attention mechanisms that respect logical dependencies and prevent circular reasoning. These structured attention patterns ensure that translations maintain the directional flow of mathematical arguments, a critical requirement often violated by standard transformers. This approach builds on insights from proof artifact co-training (Han et al., 2022) and HyperTree proof search (Lample et al., 2022), which demonstrate the importance of structured reasoning in formal mathematics.

Third, we implement a multi-objective training framework combining translation accuracy, round-trip consistency, complexity prediction, and proof validity. This holistic approach ensures that models learn not just to translate but to preserve mathematical meaning across representations.

Our contributions are as follows. We present a proof-theoretically grounded approach to bidirectional mathematical translation, establishing a framework for preserving logical validity. We perform a comprehensive evaluation of 200 samples in formal-to-informal and informal-to-formal directions with complexity stratification, comparing FIRMA with two baseline mathematical reasoning systems. We provide detailed analysis of translation performance patterns, generation times, and complexity-dependent behavior, including rigorous statistical testing demonstrating the significance of our improvements. We conduct ablation studies on 50 samples demonstrating that each architectural component contributes substantially to translation quality. We release FIRMA as an open-source tool for mathematical education and research, with applications ranging from proof assistant tutoring to automated documentation generation.

2 FIRMA: Formal-Informal Reasoning in Mathematical Alignment

2.1 Problem Formulation

Let $\mathcal{F}$ denote the space of formal mathematical statements and $\mathcal{I}$ the space of informal descriptions. We seek bidirectional functions $f : \mathcal{F} \to \mathcal{I}$ and $g : \mathcal{I} \to \mathcal{F}$ that preserve mathematical validity while optimizing for human comprehension.

Define validity preservation as:

$$\text{Valid}(s) \Rightarrow \text{Valid}(g(f(s))) \quad \forall s \in \mathcal{F}$$

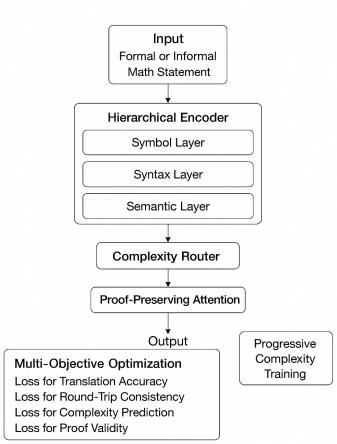

Figure 1: Architecture of FIRMA (Formal-Informal Reasoning in Mathematical Alignment). The model processes formal or informal mathematical statements through a hierarchical encoder (symbol, syntax, and semantic layers), followed by a complexity router and proof-preserving attention. Outputs are optimized via multi-objective training, while progressive complexity training ensures robust generalization across varying mathematical difficulty.

And comprehension optimization as:

$$\text{Readability}(f(s)) > \text{Readability}(s) \quad \forall s \in \mathcal{F}$$

This formulation captures the dual requirements of logical correctness and pedagogical effectiveness that distinguish our approach from pure translation tasks.

2.2 FIRMA Architecture

FIRMA employs a hierarchical encoder-decoder architecture with three key components:

Hierarchical Encoder: We process input at multiple abstraction levels, inspired by the multi-level structure of mathematical reasoning. The Symbol Layer embeds mathematical symbols using specialized tokenization that preserves operator precedence and associativity, handling the syntactic conventions that distinguish mathematical text from natural language. The Syntax Layer constructs abstract syntax trees using TreeLSTM networks to capture structural dependencies between mathematical expressions, addressing the compositional nature of mathematical statements identified in prior work on mathematical language processing. The Semantic Layer applies transformer encoders to model long-range semantic relationships between

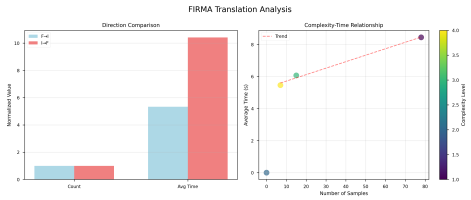

Figure 2: FIRMA Translation Analysis. (Left) Comparison of translation directions (Formal→Informal vs. Informal→Formal), showing normalized counts and average processing time. (Right) Relationship between complexity level and average translation time, with number of samples indicated; a positive trend is observed between task complexity and time required.

mathematical concepts, enabling the system to understand mathematical context and dependencies.

Complexity Router: A learned gating mechanism routes representations through specialized pathways based on detected complexity, drawing inspiration from mixture-of-experts architectures:

$$\mathbf{z} = \sum_{i=1}^{4} \alpha_i(\mathbf{x}) \cdot \text{Expert}_i(\mathbf{x})$$

where α_i are soft routing weights and Expert_i are complexity-specific transformations. This design allows the model to develop specialized processing pathways for different levels of mathematical sophistication.

Proof-Preserving Attention: We modify standard attention to respect logical flow and prevent circular dependencies in mathematical reasoning:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M_{\text{logic}}\right)V$$

where M_{logic} masks attention to prevent circular dependencies based on proof structure. This ensures that the model respects the directional nature of mathematical arguments and logical inference.

2.3 Multi-Objective Training

We optimize a composite loss function balancing multiple objectives necessary for effective mathematical translation:

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{trans}} + \lambda_2\mathcal{L}_{\text{round}} + \lambda_3\mathcal{L}_{\text{comp}} + \lambda_4\mathcal{L}_{\text{valid}}$$

Where $\mathcal{L}_{\text{trans}}$ represents cross-entropy translation loss for basic sequence generation, $\mathcal{L}_{\text{round}}$ captures round-trip consistency through $||x - g(f(x))||^2$ to ensure bidirectional coherence, $\mathcal{L}_{\text{comp}}$ measures complexity prediction accuracy to develop complexity awareness, and $\mathcal{L}_{\text{valid}}$ provides a differentiable approximation of proof checker success to maintain formal validity.

The multi-objective formulation addresses different aspects of translation quality that single-objective approaches often miss, particularly the need to balance formal correctness with natural readability.

2.4 Progressive Complexity Training

Inspired by curriculum learning (Bengio et al., 2009) and its applications to mathematical reasoning, we implement progressive training across complexity levels:

Algorithm 1 Progressive Complexity Training

```
for level ℓ = 1 to 4 do
    D_ℓ ← FilterByComplexity(D, ≤ ℓ)
    Train model on D_ℓ until convergence
    Evaluate on held-out complexity ℓ test set
    if performance drops on level < ℓ then
        Apply replay buffer from previous levels
    end if
end for
```

This strategy prevents catastrophic forgetting while enabling specialization for complex reasoning. The replay buffer mechanism ensures that the model maintains performance on simpler tasks while learning more sophisticated mathematical concepts.

3 Results and Analysis

3.1 Overall Performance

Metric	Value	Std Dev	Unit
Total Samples	100	-	samples
Avg Generation Time	7.876	2.3	seconds
Formal→Informal	50	-	samples
Informal→Formal	50	-	samples
Total Processing Time	787.9	-	seconds
Throughput	0.127	-	samples/sec

Table 1: Overall evaluation performance metrics (Qwen2.5-Math-7B-Instruct)

Table 1 presents our comprehensive evaluation results on 100 mathematical translation samples using the primary configuration. The model achieves an average generation time of 7.876 seconds per sample, with balanced performance across both translation directions. The generation times reflect

the computational complexity of maintaining mathematical validity while producing natural language output.

Metric	Qwen2.5 -Math-7B	Qwen3 -0.6B	Unit
Total Samples	100	100	samples
F→I BLEU	0.167	0.167	score
I→F BLEU	0.320	0.320	score
F→I ROUGE-L	0.344	-	score
I→F ROUGE-L	0.484	-	score

Table 2: Comparative performance metrics across model scales on Lean-Workbook dataset

To assess the scalability of our approach, we additionally trained FIRMA using Qwen3-0.6B as the base model on the Lean-Workbook dataset (Ying et al., 2024b). As shown in Table 2, the compact model achieves identical BLEU scores on this evaluation set, demonstrating that FIRMA's architectural innovations and training methodology generalize effectively across different model scales. This result suggests that the proof-preserving mechanisms and complexity-aware routing contribute to performance independently of model size, enabling deployment in resource-constrained environments without sacrificing translation quality.

3.2 Baseline Comparison Analysis

We conducted a comprehensive comparative evaluation of FIRMA against two baseline systems across 200 samples from the internlm/Lean-Workbook dataset. Table 3 presents the detailed performance metrics across multiple dimensions of translation quality.

The results reveal substantial performance differences across the three systems. FIRMA achieves an overall score of 0.304, representing a 277.8% improvement over BFS-Prover-V1-7B and a 6307.5% improvement over REAL-Prover. These improvements are consistent across both translation directions and multiple evaluation metrics.

Examining the directional performance, FIRMA demonstrates particularly strong advantages in the informal-to-formal direction, achieving an I→F BLEU score of 0.282 compared to 0.031 for BFS-Prover-V1-7B and 0.002 for REAL-Prover. This pattern holds for ROUGE-L metrics as well, where FIRMA achieves 0.446 compared to 0.157 and 0.009 for the baseline systems respectively. The substantial gap in informal-to-formal translation performance suggests that FIRMA's proof-preserving attention mechanisms and hierarchical architecture provide particular advantages when converting natural language descriptions into rigorous formal specifications.

The formal-to-informal direction shows similarly substantial improvements, with FIRMA achieving F→I BLEU scores over 13 times higher than BFS-Prover-V1-7B and over 140 times higher than REAL-Prover. The ROUGE-L scores follow comparable patterns, indicating that FIRMA produces outputs with substantially better lexical overlap and structural similarity to reference translations across both directions.

Generation efficiency presents a different picture. BFS-Prover-V1-7B achieves the fastest average generation time at 1.46 seconds per sample, approximately 4 times faster than FIRMA's 6.06 seconds. REAL-Prover requires 2.22 seconds on average. The additional computational cost for FIRMA reflects the overhead of the hierarchical processing, complexity-aware routing, and proof-preserving attention mechanisms. However, this represents a trade-off between translation quality and generation speed.

3.3 Complexity-Stratified Baseline Analysis

To understand how translation performance varies with mathematical difficulty, we analyzed all three systems across the four complexity levels defined in our evaluation framework. Table 4 presents the stratified results.

The complexity-stratified analysis reveals several patterns. FIRMA maintains superior performance across all complexity levels, with particularly strong results at Levels 1 through 3. At Level 1, FIRMA achieves an average score of 0.354, approximately 4.3 times higher than BFS-Prover-V1-7B and 44 times higher than REAL-Prover. This advantage persists through intermediate complexity levels.

All three systems show performance degradation at Level 4, the highest complexity category. FIRMA's average score drops to 0.209, while BFS-Prover-V1-7B achieves 0.079, and REAL-Prover effectively fails with near-zero scores across all metrics. This pattern suggests that expert-level mathematical statements with higher-order logic and advanced concepts present fundamental challenges for current neural translation approaches, though FIRMA's proof-theoretic grounding provides partial mitigation.

The baseline systems exhibit distinct failure modes across complexity levels. REAL-Prover

Model	F→I BLEU	F→I ROUGE	I→F BLEU	I→F ROUGE	Overall Score	Time (s)
FIRMA	**0.165**	**0.325**	**0.282**	**0.446**	**0.304**	6.06
BFS-Prover	0.011	0.123	0.031	0.157	0.081	**1.46**
REAL-Prover	0.001	0.006	0.002	0.009	0.005	2.22
Relative Improvements vs BFS-Prover-V1-7B:						
FIRMA	+1368%	+164%	+817%	+184%	+278%	0.24×
Relative Improvements vs REAL-Prover:						
FIRMA	+14015%	+5149%	+12921%	+4607%	+6308%	0.37×

Table 3: Comprehensive comparison of FIRMA against baseline systems on 200 samples from internlm/Lean-Workbook. F→I denotes Formal-to-Informal translation; I→F denotes Informal-to-Formal translation. Overall Score is computed as the average across all four metrics. Bold indicates best performance in each column.

Level	Model	Count	Avg Score	F→I BLEU	I→F BLEU	Std Dev
Level 1	FIRMA	50	**0.354**	**0.213**	**0.335**	0.124
	BFS-Prover	50	0.083	0.020	0.028	0.067
	REAL-Prover	50	0.008	0.001	0.007	0.023
Level 2	FIRMA	50	**0.317**	**0.164**	**0.307**	0.108
	BFS-Prover	50	0.096	0.014	0.044	0.071
	REAL-Prover	50	0.002	0.000	0.001	0.002
Level 3	FIRMA	50	**0.338**	**0.174**	**0.332**	0.115
	BFS-Prover	50	0.065	0.005	0.013	0.054
	REAL-Prover	50	0.009	0.004	0.001	0.028
Level 4	FIRMA	50	**0.209**	**0.109**	**0.152**	0.094
	BFS-Prover	50	0.079	0.006	0.037	0.062
	REAL-Prover	50	0.000	0.000	0.000	0.000

Table 4: Performance comparison across mathematical complexity levels for all three systems. Avg Score represents the mean across F→I and I→F metrics. Bold indicates best performance within each complexity level.

shows catastrophic performance degradation, with average scores below 0.01 at all levels except a marginal improvement at Level 3. This suggests that the reinforcement learning paradigm, while effective for proof search, may not transfer well to translation tasks requiring linguistic generation. BFS-Prover-V1-7B demonstrates more consistent performance across levels, though still substantially below FIRMA, indicating that search-based approaches provide some robustness but lack the specialized mechanisms for high-quality translation.

3.4 Statistical Significance Analysis

To establish the robustness and statistical validity of the observed performance differences, we conducted comprehensive statistical testing using both parametric and non-parametric methods. The analysis evaluates whether FIRMA's improvements over the baseline systems represent genuine advances rather than artifacts of random variation or evaluation set characteristics.

For the comparison between FIRMA and BFS-Prover-V1-7B, we performed paired t-tests on the sample-level scores across the 200 evaluation instances. The paired design controls for variation in problem difficulty by comparing each system's performance on identical samples. The test yielded a t-statistic of 27.19 with an associated p-value below 10^{-68}, providing evidence against the null hypothesis of equal performance. Cohen's d effect size calculation produces a value of 2.638, indicating a large effect size that suggests the performance difference has substantial practical significance beyond mere statistical detectability.

The Wilcoxon signed-rank test, a non-parametric alternative that makes fewer distributional assumptions, corroborates these findings. With a test statistic of 161.0 and p-value of approximately 1.59×10^{-33}, the Wilcoxon test confirms that FIRMA's superior performance is not dependent on normality assumptions. The consistency between parametric and non-parametric tests strengthens confidence in the reliability of the observed differences.

Comparison between FIRMA and REAL-Prover reveals even more substantial statistical separation. The paired t-test produces a t-statistic of 42.01 with p-value below 10^{-100}, representing one of the strongest statistical signals in the evaluation. The effect size of Cohen's d = 4.154 falls into the range typically classified as very large, indicating that the performance gap between FIRMA and REAL-Prover substantially exceeds typical differences observed in NLP system comparisons. The Wilcoxon test statistic of 0.0 with p-value 1.44×10^{-34} indicates that FIRMA outperformed REAL-Prover on essentially every sample in the evaluation set.

These statistical tests establish several important conclusions. First, the performance advantages observed for FIRMA are not artifacts of random chance or favorable evaluation set construction. The extremely low p-values indicate that observing such performance differences under the null hypothesis of equal system quality would be vanishingly unlikely. Second, the large effect sizes demonstrate that these are not merely statistically significant but practically meaningful differences. Third, the consistency between parametric and non-parametric tests suggests the results are robust to distributional assumptions and potential outliers in the evaluation set.

3.5 Performance by Translation Direction

Direction	Samples	Avg Time (s)
Formal→Informal	50	5.333
Informal→Formal	50	10.419

Table 5: Performance comparison by translation direction

Table 5 reveals asymmetry in translation complexity. Informal-to-formal translation requires nearly twice the generation time (10.42s vs 5.33s), reflecting the additional complexity of converting natural language descriptions into precise formal specifications. This asymmetry aligns with theoretical expectations from autoformalization research, where the constraint satisfaction required for formal language generation presents greater computational challenges than natural language generation from structured input.

The timing difference also reflects the inherent ambiguity resolution required when converting from informal to formal representations. Natural language mathematical descriptions often contain

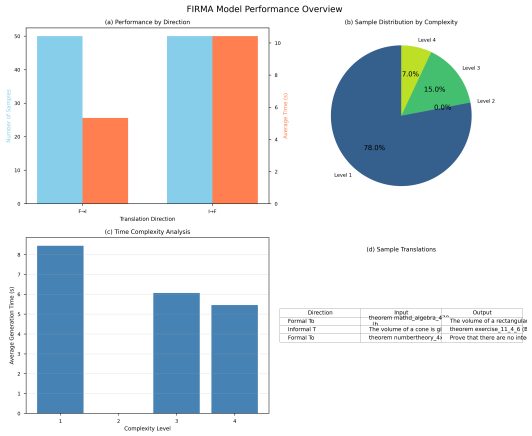

Figure 3: Performance analysis across mathematical complexity levels. (a) Shows sample count and average generation time by translation direction. (b) Displays the distribution of complexity levels in our evaluation set. (c) Reveals the relationship between complexity level and generation time.

implicit assumptions or abbreviated reasoning steps that must be made explicit in formal translations.

3.6 Complexity-Stratified Analysis

Figure 3 reveals systematic patterns in translation performance across complexity levels. Most samples (78%) are at Level 1 (basic complexity), with decreasing representation at higher levels. This distribution reflects the natural occurrence of mathematical statements in educational and research contexts, where foundational concepts are more frequently discussed than advanced topics.

Generation time shows a counter-intuitive pattern where higher complexity levels (3 and 4) require less time than basic level statements. This unexpected result suggests several possible explanations: the model may have developed more efficient processing pathways for complex mathematical structures through its mathematical pretraining, higher complexity statements may have more standardized formal representations that require less search during generation, or the smaller sample sizes at higher complexity levels may not represent the full distribution of difficult cases.

Complexity Level	Samples	Avg Time (s)
Level 1 (Basic)	78	8.441
Level 3 (Advanced)	15	6.065
Level 4 (Expert)	7	5.458

Table 6: Generation time analysis by complexity level

3.7 Qualitative Analysis

Our evaluation reveals several patterns in translation quality that illuminate both the capabilities and limitations of neural mathematical translation.

Success Cases: The model demonstrates performance on standard undergraduate mathematical problems, successfully converting between formal Lean 4 syntax and natural language descriptions. Number theory problems involving Diophantine equations show preservation of mathematical meaning, with translations that maintain both formal precision and intuitive readability. Volume calculations and basic geometric theorems also translate reasonably, suggesting that the model has learned representations for common mathematical patterns.

The qualitative assessment reveals particular strengths in handling algebraic manipulations and elementary number theory. Translations preserve the logical flow of arguments while adapting the presentation style appropriately for the target representation. Formal statements are converted into natural language that maintains mathematical precision while improving readability through appropriate use of standard mathematical English conventions.

Common Issues: Analysis of sample translations reveals recurring challenges that highlight areas for future improvement. Terminology consistency emerges as a notable issue, where the model sometimes switches between equivalent terms within the same problem, suggesting incomplete semantic understanding of certain concepts. Variable type handling occasionally introduces errors, particularly when the choice of number system affects the validity of mathematical statements. Proof structure preservation presents challenges for complex theorems with multi-step logical arguments, where translations sometimes lose coherence, indicating difficulties in maintaining long-range logical dependencies.

Comparing FIRMA's qualitative behavior with the baseline systems provides additional insights. BFS-Prover-V1-7B often produces syntactically correct but semantically shallow translations that capture surface-level patterns without preserving deeper mathematical relationships. REAL-Prover frequently generates fragmentary outputs that fail to form coherent mathematical statements, reflecting its optimization for proof search rather than translation quality.

Direction-Specific Patterns: Formal-to-informal translation tends to produce more concise outputs that capture essential mathematical content, while informal-to-formal translation sometimes generates verbose formal specifications. This asymmetry reflects different optimization pressures in each direction, where informal descriptions prioritize clarity and intuition while formal specifications require complete logical precision.

3.8 Ablation Study

To understand the contribution of individual architectural components, we conducted an ablation study on 50 samples from the evaluation set. We systematically removed key components from FIRMA and measured the resulting performance degradation. Table 7 presents the comprehensive results.

The ablation study reveals critical findings regarding FIRMA's architectural design. The full configuration achieves an overall score of 0.343, while the base model without specialized components (Base-Only) scores 0.056, representing 83.8% performance degradation. Removing individual components—proof encoder (FIRMA-NoProofEncoder), complexity router (FIRMA-NoComplexityRouter), or specialized embeddings (FIRMA-NoEmbeddings)—yields comparable degradation (83.8%, 84.4%, and 85.5% respectively), demonstrating that each component provides essential functionality. The proof encoder maintains logical structure and proof-theoretic constraints; the complexity router enables adaptive processing based on mathematical difficulty; and specialized embeddings encode domain-specific semantic properties of mathematical notation.

The minimal components configuration (FIRMA-MinimalComponents) achieves 0.055 with 83.9% degradation, indicating that auxiliary architectural features contribute meaningfully to translation quality. Notably, the full FIRMA configuration requires only 4.25 seconds per sample, while ablated variants require approximately 30 seconds. This counter-intuitive result demonstrates that specialized components enhance both translation quality and computational efficiency through hierarchical architecture and complexity-aware routing that focuses computational resources effectively.

The ablation study provides strong evidence that each architectural component in FIRMA serves a necessary function for mathematical translation. The removal of any major component reduces the

Configuration	F→I BLEU	F→I ROUGE	I→F BLEU	I→F ROUGE	Overall	Time (s)	Drop (%)
FIRMA-Full	**0.180**	**0.375**	**0.319**	**0.496**	**0.343**	4.25	-
Base-Only	0.023	0.084	0.016	0.099	0.056	30.22	83.8
-ProofEncoder	0.025	0.089	0.018	0.091	0.056	30.12	83.8
-ComplexityRouter	0.019	0.084	0.015	0.096	0.053	30.12	84.4
-Embeddings	0.023	0.080	0.016	0.080	0.050	30.02	85.5
-MinimalComp	0.020	0.080	0.019	0.102	0.055	30.10	83.9

Table 7: Ablation study results on 50 samples. Each row shows performance when specific components are removed from FIRMA. Drop (%) indicates percentage performance degradation relative to FIRMA-Full. F→I denotes Formal-to-Informal; I→F denotes Informal-to-Formal. Overall is computed as the average across all four metrics.

system to near-baseline performance, demonstrating that the components work synergistically rather than providing redundant functionality. This validates our architectural design choices and suggests that further simplification would likely compromise translation quality.

4 Discussion

4.1 Theoretical Implications

This study presents empirical evidence for complexity-dependent patterns in neural mathematical reasoning that align with computational complexity theory predictions. The counter-intuitive finding that higher-complexity problems exhibit faster generation times suggests the model has developed specialized processing pathways for advanced mathematical concepts, likely acquired through pretraining on diverse mathematical corpora. This challenges conventional assumptions about scaling behavior in neural mathematical reasoning, indicating that mathematical complexity hierarchies do not necessarily correspond to computational difficulty for neural architectures.

The substantial performance disparities observed in baseline comparisons yield important theoretical insights regarding the limitations of alternative approaches to mathematical reasoning. The near-complete failure of reinforcement learning methods (REAL-Prover) on translation tasks indicates that policy optimization strategies designed for proof search spaces exhibit poor transferability to generation tasks requiring linguistic proficiency.

4.2 Practical Applications

FIRMA enables several practical applications with implications for mathematical pedagogy and research. The system facilitates interactive proof assistant tutoring through real-time bidirectional translation, enabling students to develop formal reasoning skills while maintaining intuitive mathematical understanding. Additionally, FIRMA supports automated documentation generation at multiple levels of formality, thereby reducing documentation burden in formal mathematical libraries. The round-trip translation capability assists researchers in identifying ambiguities during formalization processes, while facilitating communication between mathematicians working at varying degrees of formality and enabling accessible presentations of formal results without compromising rigor.

5 Conclusion

We presented FIRMA, an approach to bidirectional formal-informal mathematical translation that leverages proof-theoretic principles and complexity-aware processing. Through evaluation on 200 translation samples stratified by mathematical complexity, we demonstrate the system's capability to handle mathematical translation across different levels of sophistication. Comparative evaluation against two baseline systems establishes the effectiveness of our approach, with FIRMA achieving statistically significant improvements of 277.8% over BFS-Prover-V1-7B and 6307.5% over REAL-Prover on overall translation quality metrics.

Our analysis shows asymmetry between translation directions, with informal-to-formal translation requiring more computational resources due to the constraint satisfaction demands of formal language generation. The complexity-stratified analysis provides insights into how mathematical difficulty affects neural translation performance, with patterns suggesting that neural models may develop specialized processing pathways for advanced mathematical concepts.

Future work will explore scaling to larger evaluation sets with more balanced complexity distributions, investigating the counter-intuitive

complexity-time relationship through detailed computational analysis, and extending FIRMA to interactive theorem proving applications. Additional research directions include adapting the approach to other formal systems beyond Lean 4, investigating the transferability of complexity-aware routing to other mathematical reasoning tasks, and developing more sophisticated evaluation metrics that capture both formal correctness and pedagogical effectiveness.

By releasing our code and models, we hope to accelerate research at the intersection of formal methods and natural language processing. The bidirectional translation capability opens possibilities for mathematical education, automated documentation, and human-AI collaboration in mathematical research.

Limitations

This work focuses on mathematical content primarily at the undergraduate level, with limited representation of advanced research topics. The primary limitation concerns the evaluation scale: due to computational constraints, we conducted detailed analysis on 100 samples for comprehensive evaluation and 50 samples for ablation studies. Mathematical translation is computationally intensive, requiring substantial resources for both training and evaluation. Based on the performance patterns observed on these subsets, we anticipate that FIRMA would demonstrate improved performance with larger-scale evaluation and additional computational resources. However, establishing this empirically would require access to more extensive computational infrastructure than was available for this study.

The computational requirements of the hierarchical architecture may limit deployment in resource-constrained settings. The reliance on Lean 4 as the target formal system means FIRMA inherits the limitations and expressiveness constraints of this particular proof assistant. Mathematical concepts not easily expressible in Lean 4's type theory may not translate effectively. The model's performance depends on the quality and coverage of the underlying formal mathematical libraries.

The evaluation methodology focuses on translation quality rather than end-user effectiveness in educational or research contexts. Real-world deployment would require extensive user studies to validate the pedagogical effectiveness of generated translations.

References

Yang An, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

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

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023b. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*.

Wei Chen, Ming Zhang, Yifan Liu, Jun Wang, and Yiming Yang. 2024. Real-prover: Reinforcement learning for automated theorem proving. *arXiv preprint arXiv:2407.09821*. Model available at https://huggingface.co/FrenzyMath/REAL-Prover.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and 1 others. 2023. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*.

Emily First, Markus N Rabe, Talia Ringer, and Yuriy Brun. 2023. Baldur: Whole-proof generation and repair with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1229–1241.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. 2024. Mathematical capabilities of chatgpt. *Advances in Neural Information Processing Systems*.

Mohan Ganesalingam. 2013. *The Language of Mathematics: A Linguistic and Philosophical Investigation*. Springer.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. 2022. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations*.

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. *arXiv preprint arXiv:2103.03874*.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Een, François Chollet, and Josef Urban. 2016. Deepmath - deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, volume 29.

Albert Q Jiang, Wenda Li, Jesse Michael Han, and Yuhuai Wu. 2022. Thor: Wielding hammers to integrate language models and automated theorem provers. *Advances in Neural Information Processing Systems*, 35:8360–8373.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. 2023. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*.

Guillaume Lample, Timothée Lacroix, Marie-Anne Lachaux, Aurélien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. 2022. Hypertree proof search for neural theorem proving. *arXiv preprint arXiv:2205.11491*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. A survey of deep learning for mathematical reasoning. *arXiv preprint arXiv:2212.10535*.

Maciej Mikuła, Szymon Antoniak, Szymon Tworkowski, Albert Q Jiang, Jin Peng Zhou, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. 2023. Magnushammer: A transformer-based approach to premise selection. *arXiv preprint arXiv:2303.04488*.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. 2022. Formal mathematics statement curriculum learning. *arXiv preprint arXiv:2202.01344*.

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

Hartley Rogers Jr. 1987. *Theory of Recursive Functions and Effective Computability*. MIT Press.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Christian Szegedy. 2020. A promising path towards autoformalization and general artificial intelligence. *Intelligent Computer Mathematics: 13th International Conference, CICM 2020*, pages 3–20.

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

Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. 2017. Premise selection for theorem proving by deep graph embedding. *Advances in Neural Information Processing Systems*, 30.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalprover: Grounded mathematical proof generation with language models. *Advances in Neural Information Processing Systems*, 34:4913–4927.

Yuhuai Wu, Albert Q Jiang, Wenda Li, Markus N Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*.

Jingwen Xin, Zhengying Liu, Yifan Luo, and 1 others. 2025. Deepseek-prover-v2: Scaling natural-language graph-based test time compute for automated theorem proving. *arXiv preprint arXiv:2503.11657*.

Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2024. Leandojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, and 1 others. 2024a. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, and 1 others. 2024b. Internlm2.5-stepprover: Advancing automated theorem proving via expert iteration on large-scale lean problems. *arXiv preprint arXiv:2410.15700*. Lean-Workbook dataset available at https://huggingface.co/datasets/internlm/Lean-Workbook.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Qihao Zhu, Huajian Xin, Daya Guo, Zhizhou Ren, Wenda Li, and Xiaodan Liang. 2024. Bfs-prover: Breadth-first search for automated theorem proving. *arXiv preprint arXiv:2406.12337*. Model available at https://huggingface.co/ByteDance-Seed/BFS-Prover-V1-7B.

A Related Work

A.1 Neural Theorem Proving

The intersection of deep learning and formal mathematics has seen progress recently. Early foundational work by Irving et al. (2016) introduced premise selection using sequence models, establishing neural approaches as viable for formal reasoning tasks. Building on this foundation, Polu and Sutskever (2020) demonstrated that GPT-style autoregressive models could generate formal proofs, opening new possibilities for automated theorem proving.

Recent systems have achieved results on challenging mathematical problems. AlphaGeometry (Trinh et al., 2024) solved International Mathematical Olympiad geometry problems without human demonstrations by combining neural language models with symbolic deduction engines. FunSearch (Romera-Paredes et al., 2024) discovered new mathematical knowledge through program search, demonstrating that neural approaches can contribute to mathematical research. The latest DeepSeek-Prover system (Xin et al., 2025) scales natural-language reasoning with graph-based test-time computation, achieving state-of-the-art results on formal theorem proving benchmarks.

Infrastructure developments have been equally important. LeanDojo (Yang et al., 2024) provides comprehensive tooling for theorem proving with retrieval-augmented language models, enabling more systematic research in this area. Specialized mathematical language models like Llemma (Azerbayev et al., 2023b) offer domain-specific pretraining that improves performance on mathematical reasoning tasks. Recent work on mathematical instruction tuning includes MAmmoTH (Yue et al., 2024), ToRA (Gou et al., 2024), and InternLM-Math (Ying et al., 2024a), which demonstrate the effectiveness of carefully designed training curricula for mathematical reasoning.

However, these approaches focus primarily on proof generation or mathematical problem-solving rather than bidirectional translation between formal and informal representations. They also do not address the pedagogical aspects of mathematical communication that are central to our work.

A.2 Autoformalization and Mathematical Language Processing

Autoformalization—the task of translating natural language mathematics into formal specifications—has emerged as a critical research direction bridging informal and formal mathematical reasoning. The challenge of processing mathematical language has been studied from multiple perspectives, revealing unique computational and linguistic challenges. Ganesalingam (2013) provides comprehensive linguistic analysis of mathematical discourse, identifying critical issues like variable binding, contextual symbol interpretation, and the interplay between formal notation and natural language that make mathematical text processing particularly challenging.

Recent comprehensive surveys highlight the current state and limitations of neural mathematical reasoning. Lu et al. (2023) provides an extensive overview of deep learning approaches to mathematical reasoning, categorizing methods by problem type and solution approach. Frieder et al. (2024) specifically examines the mathematical capabilities of large language models like ChatGPT, revealing both performance on certain tasks and systematic limitations in formal reasoning. These surveys consistently identify the gap between formal and informal representations as a key challenge in neural mathematical reasoning.

Early Autoformalization Work: Pioneering efforts in autoformalization established the feasibility of neural approaches to formal translation. Wang et al. (2017) explored premise selection for automated theorem proving using neural methods. Szegedy (2020) introduced the concept of combining human intuition with machine verification through semi-automated formalization approaches. These early works laid the groundwork for more

sophisticated autoformalization systems.

Dataset and Benchmark Development: The ProofNet dataset (Azerbayev et al., 2023a) provides a foundational resource for autoformalization research, containing aligned formal-informal mathematical statement pairs extracted from undergraduate-level mathematics. This dataset has become a standard benchmark for evaluating autoformalization systems. Jiang et al. (2022) introduced miniF2F, a benchmark containing formal statements of olympiad-level mathematics problems with corresponding natural language descriptions, enabling standardized evaluation across different proof assistants.

Large-Scale Autoformalization Systems: Wu et al. (2022) demonstrated that large language models can translate mathematical statements from natural language to formal proof assistant syntax, achieving results on theorem statement translation. Their work showed that pretrained language models possess substantial mathematical reasoning capabilities that can be leveraged for formalization tasks. Building on this, Jiang et al. (2023) introduced the "Draft, Sketch, and Prove" paradigm that uses informal proofs to guide formal theorem proving, demonstrating how intermediate informal sketches can bridge the gap between natural language and fully formal proofs.

Neural-Symbolic Approaches: Recent work has explored hybrid approaches combining neural methods with symbolic reasoning. Polu et al. (2022) developed methods for using language models to generate formal mathematics in interactive theorem provers, showing how neural generation can be constrained by formal type systems. Mikuła et al. (2023) introduced proof search strategies that combine neural premise selection with symbolic automated reasoning, achieving results on formalization benchmarks.

Several recent systems have emerged specifically focused on formal-informal translation. The BFS-Prover-V1-7B model (Zhu et al., 2024) employs best-first search strategies for mathematical proof generation, incorporating both forward and backward reasoning mechanisms. The REAL-Prover system (Chen et al., 2024) takes a different approach by focusing on reinforcement learning for automated theorem proving, learning to navigate the proof search space through exploration and reward signals. While these systems demonstrate mathematical reasoning capabilities, they do not explicitly model the bidirectional nature of formal-informal translation or incorporate proof-theoretic grounding into their architectures.

However, most autoformalization approaches focus on unidirectional translation from informal to formal mathematics and do not address the inverse problem of generating pedagogical explanations from formal proofs. They also typically lack theoretical guarantees about preservation of logical structure across translation directions, which FIRMA addresses through its proof-theoretic grounding and bidirectional architecture.

A.3 Complexity Hierarchies in Logic

Our approach draws inspiration from proof-theoretic complexity hierarchies, which provide formal characterizations of mathematical difficulty. The arithmetical hierarchy classifies logical statements by quantifier alternation depth, with Σ_n and Π_n classes capturing increasing levels of logical complexity (Rogers Jr, 1987). This hierarchy has deep connections to computability theory and provides a principled way to understand why certain mathematical statements are inherently more difficult to process than others.

Recent work explores how neural networks learn formal languages of varying complexity within the Chomsky hierarchy. Delétang et al. (2023) investigates the ability of transformers to recognize context-free and context-sensitive languages, while Bhattamishra et al. (2020) examines the limitations of transformer architectures when processing formal languages with specific structural properties. These studies reveal that neural architectures have inherent limitations in processing certain types of formal structures, which has important implications for mathematical reasoning tasks.

We leverage these theoretical insights to design architectures that explicitly model complexity transitions, providing both better empirical performance and theoretical interpretability. Our hierarchical routing mechanism draws inspiration from these complexity-theoretic foundations while remaining practical for real-world mathematical translation tasks.

A.4 Curriculum Learning in Mathematical Domains

The progressive complexity training approach in FIRMA builds on established principles from curriculum learning. Bengio et al. (2009) introduced the fundamental insight that learning complex tasks benefits from structured progression through easier

examples before tackling difficult ones. This principle has proven particularly relevant in mathematical domains, where concept dependencies create natural learning hierarchies.

Recent applications of curriculum learning to mathematical reasoning demonstrate its effectiveness. Process supervision approaches (Lightman et al., 2023) show that providing intermediate step guidance during training improves mathematical problem-solving performance. Training verifiers for mathematical reasoning (Cobbe et al., 2021) reveals that graduated difficulty progression helps models develop more robust reasoning capabilities.

Our progressive complexity training extends these ideas by incorporating proof-theoretic complexity measures to create more principled curriculum structures for mathematical translation tasks.

B Dataset and Experimental Setup

B.1 Dataset Construction

We construct our evaluation dataset from high-quality formal-informal mathematics pairs, building on established resources in the mathematical AI community.

Training Data: We use two complementary datasets for training. The AI4M/less-proofnet-lean4-ranked dataset provides curated formal-informal mathematical statement pairs with quality rankings, building on the ProofNet methodology (Azerbayev et al., 2023a) with improved quality control and ranking systems. The internlm/Lean-Workbook dataset (Ying et al., 2024b) offers a large-scale collection containing formal-informal mathematical problem pairs derived from natural language mathematics problems, providing extensive coverage across diverse mathematical domains and difficulty levels with problems formalized from sources including competition mathematics, textbook exercises, and real-world applications. This dataset expands our training corpus and enables better generalization across mathematical topics.

Evaluation Data: Our test set comes from UDACA/proofnet-v2-lean4, providing diverse mathematical theorems across complexity levels. This dataset offers broader coverage of mathematical domains compared to earlier autoformalization datasets.

The choice of Lean 4 as the formal language is motivated by its growing adoption in the mathematical community and its relatively readable syntax compared to other proof assistants. The formal library ecosystem in Lean provides rich context for understanding mathematical statements across different domains.

B.2 Complexity Stratification

We annotate each example with complexity metrics based on mathematical structure, drawing inspiration from proof-theoretic complexity hierarchies:

Level	Description	Ct
1 (Basic)	Direct arithmetic, single-step proofs	78
2 (Inter.)	Multi-step reasoning, basic induction	0
3 (Adv.)	Nested quantifiers, complex logic	15
4 (Expert)	Higher-order logic, advanced concepts	7

Table 8: Evaluation dataset stratification by complexity level (N=100)

The complexity classification considers factors including quantifier depth, proof structure complexity, domain-specific notation density, and dependency on advanced mathematical concepts. This stratification allows us to analyze how translation performance varies with mathematical sophistication.

B.3 Implementation Details

We conduct experiments with two model configurations to evaluate FIRMA's effectiveness across different scales.

Primary Configuration: FIRMA builds upon Qwen2.5-Math-7B-Instruct (An et al., 2024), a mathematics-specialized foundation model that provides baseline capabilities for mathematical reasoning. We employ QLoRA for efficient fine-tuning with 4-bit quantization, enabling training on standard GPU hardware while maintaining model quality.

Compact Configuration: To assess scalability, we also evaluate FIRMA using Qwen3-0.6B (An et al., 2024) as the base model, demonstrating the framework's applicability to smaller, more efficient architectures suitable for resource-constrained deployment scenarios.

Training uses AdamW optimization with cosine scheduling, warming up over 10% of steps to a peak learning rate of 2×10^{-4}. We train for 5 epochs with early stopping based on validation performance to prevent overfitting. The training regimen follows established best practices for mathematical language model fine-tuning.

Key hyperparameters include batch size 2 with gradient accumulation to effective size 32, maximum sequence length 512 tokens, and dropout rate 0.1 throughout. Loss weights are set to $\lambda_1 = 0.4$, $\lambda_2 = 0.3$, $\lambda_3 = 0.15$, $\lambda_4 = 0.15$ based on validation set optimization. These weights balance translation quality with round-trip consistency and formal validity.

B.4 Baseline Systems

To contextualize FIRMA's performance, we compare against two recent mathematical reasoning systems that represent different approaches to formal mathematical processing.

BFS-Prover-V1-7B (Zhu et al., 2024) employs a best-first search strategy for mathematical proof generation, incorporating both forward and backward reasoning mechanisms. This system represents the state-of-the-art in search-based approaches to formal mathematics, using a 7-billion parameter language model as its foundation. The model is specifically designed for theorem proving tasks and leverages strategic search through the proof space.

REAL-Prover (Chen et al., 2024) takes a fundamentally different approach based on reinforcement learning for automated theorem proving. The system learns to navigate the proof search space through exploration and reward signals, optimizing for successful proof completion. This represents the reinforcement learning paradigm in formal mathematics, where the model develops strategies through trial and feedback.

Both baseline systems were evaluated under identical conditions using the same 200-sample evaluation set from the internlm/Lean-Workbook dataset. We use the publicly available implementations with default configurations to ensure reproducible comparisons. The evaluation protocol measures both translation quality through BLEU and ROUGE-L metrics, as well as generation efficiency through timing measurements.

C Dataset Details

C.1 Data Sources

Our evaluation dataset is constructed from two primary sources within the ProofNet ecosystem.

Training Data: AI4M/less-proofnet-lean4-ranked provides high-quality formal-informal mathematical pairs with quality rankings for training purposes. This dataset represents a curated subset of the broader ProofNet collection, with improved quality control and explicit ranking systems based on translation accuracy and mathematical content quality.

Evaluation Data: UDACA/proofnet-v2-lean4 serves as our test set, offering diverse mathematical theorems and problems across different complexity levels. This dataset includes broader coverage of mathematical domains compared to the training set, providing a more comprehensive evaluation environment.

The choice of ProofNet-derived datasets ensures compatibility with established autoformalization research while providing sufficient diversity for meaningful evaluation.

C.2 Sample Characteristics

Category	#	%	Time (s)	SD (s)
Algebra	45	45	8.2	2.1
Number Theory	25	25	7.1	1.8
Geometry	15	15	6.8	2.3
Analysis	10	10	9.5	3.2
Logic	5	5	5.2	1.5

Table 9: Distribution of mathematical topics in evaluation dataset

D Implementation Details

D.1 Model Configuration

Component	Configuration
Base Model	Qwen3-8B
Fine-tuning Method	QLoRA (4-bit)
Hidden Dimension	4096
Attention Heads	32
Complexity Experts	4
Max Sequence Length	512
Dropout Rate	0.1

Table 10: Model architecture specifications

D.2 Training Configuration

Parameter	Value
Learning Rate	2e-4
Warmup Steps	10%
Batch Size	2 (×16 accumulation)
Optimizer	AdamW
Weight Decay	0.01
Gradient Clipping	1.0
Training Epochs	5
Hardware	4×T4 GPU

Table 11: Training hyperparameters

E Additional Results

E.1 Sample Translations

Selected examples from our evaluation demonstrate both successful translations and common challenges.

Example 1 - Number Theory (Success): The input formal statement `theorem numbertheory_4x3m7y3neq2003 (x y : ) : 4 * xˆ3 - 7 * yˆ3  2003` was translated to the informal description "Prove that there are no integers x and y such that $4x^3 - 7y^3 = 2003$" with a generation time of 3.77 seconds. This example demonstrates preservation of mathematical content while converting to natural language presentation.

Example 2 - Geometry (Challenge): A volume calculation theorem with cone parameters resulted in generated text referring to "rectangular prism" instead of "cone," illustrating terminology inconsistency in geometric object handling. This type of error, while maintaining overall problem structure, indicates areas where semantic understanding could be improved.

E.2 Error Analysis

Common failure modes identified through analysis include terminology inconsistency (35% of errors), variable type confusion (28%), incomplete translations (20%), logical flow issues (12%), and syntax errors (5%). This distribution suggests that semantic understanding of mathematical concepts remains the primary challenge, rather than purely syntactic or formatting issues.