Synthetic Proofs with Tool-Integrated Reasoning: Contrastive Alignment for LLM Mathematics with Lean

Mark Obozov[1] Michael Diskin[2] Aleksandr Beznosikov[1] Alexander Gasnikov[1] Serguei Barannikov[3]

[1]Innopolis University, Russia [2]HSE University, Russia [3]Skoltech, CNRS

Abstract

Modern mathematical reasoning benchmarks focus on answer finding rather than proof verification. We present:

- **Lean-based synthetic problem generation** with tree-based conjecture algorithm
- **Tool-Integrated Reasoning (TiR)** framework for partial proof validation
- **Contrastive preference optimization** to align model outputs
- **30,000 synthetic problems dataset** for benchmarking

Key Result: Up to **57% improvement** on MiniF2F benchmark across Qwen-2.5 models (0.5B-7B parameters)

Motivation & Challenges

Core Problem

- Vast search space of mathematical proofs
- Gap between answer-finding and formal proving
- Computational expense of exhaustive formal search

Our Approach

Combine LLM flexibility with structured verification:

- Generate diverse synthetic problems using formal tools
- Validate proofs without full formalization
- Align models through contrastive learning

Tree-Based Conjecture Generation

Algorithm 1 Synthetic Conjecture Generation

Require: Proof graph $T = (V, E)$, steps N, initial conjecture X_0, proof Y

$X_{\text{current}} \leftarrow X_0$
$T_{\text{trace}} \leftarrow \{\}$
for $i = 1$ to N **do**
  $X_{\text{next}} \leftarrow$ choose neighbor of X_{current}
  $T_{\text{trace}} \leftarrow T_{\text{trace}} \cup \{X_{\text{next}}\}$
  $X_{\text{current}} \leftarrow X_{\text{next}}$
end for
reverse T_{trace}
$T_{\text{trace}} \leftarrow Y \cup T_{\text{trace}}$

Neighbor selection strategies:

- Lean hints and Lean Copilot tooling
- Symbol overlap heuristics
- LLM-predicted difficulty

Tool-Integrated Reasoning (TiR)

Instead of full formal proofs, generate verification functions:

Algorithm 2 TiR-Based Validation

Require: Function $f : \{\dots\} \to \{0, 1\}$, trials N, environment $\mathcal{E}$

$success \leftarrow 0$
for $i = 1$ to N **do**
  $x \leftarrow \mathcal{E}$.generate_input()
  $output \leftarrow f(x)$
  if $output = 1$ **then**
    $success \leftarrow success + 1$
  end if
end for
return $\frac{success}{N}$

Example: For $a^2 + b^2 + c^2 \geq ab + ac + bc$, TiR generates a Python function that tests the inequality on random inputs.

Contrastive Alignment Training

SimPO Objective

$$\mathcal{L}_{\text{SimPO}} = -\mathbb{E}_{(x,y_w,y_l)} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right] \quad (1)$$

Where:

- y_w: correct proof (validated by TiR)
- y_l: incorrect proof (generated via rejection)
- $\beta = 2.0$: length normalization
- $\gamma = 0.5$: margin between scores

Generating Rejected Values

1. Use smaller/weaker LLM
2. Prompt model to introduce mistakes
3. Perturb proof trajectories

Experimental Results

Main Results on MiniF2F

Model	Size	Aligned	Baseline
Qwen-2.5	0.5B	**0.22** (+57%)	0.14
Qwen-2.5	1.5B	**0.37** (+28%)	0.29
Qwen-2.5	7B	**0.53** (+13%)	0.47

Key Findings

- **Consistent gains** across all model sizes
- **Largest relative improvement** for smallest models
- **TiR filtering** effectively removes invalid proofs
- Benefits persist as model size grows

Training Details

- 3 epochs on 2× A100 80GB GPUs
- Modified torchtune library
- Gradient checkpointing for memory efficiency
- Best checkpoint selected via validation

Example Generated Problems

Original Problem

For $a, b, c \geq 0$ and $a + b + c = 1$, prove:

$$1 + 12abc \geq 4(ab + bc + ac)$$

Generated Extension

After tree-based walk:

$$1 + 12ab(1 - a - b) \geq 4(ab + b(1 - a - b) + (1 - a - b)a)$$

TiR Validation Example

```
def f(a, b, c):
    lhs = a**2 + b**2 + c**2
    rhs = a*b + a*c + b*c
    return 1 if lhs >= rhs else 0
```

Contributions & Future Work

Main Contributions

- Tree-based algorithm for synthetic conjecture generation
- TiR framework for scalable proof validation
- 30,000 problem dataset (to be released)
- Demonstrated improvements on formal benchmarks

Future Directions

- Chain-of-thought prompting for generation
- Extension to geometry and other domains
- Stronger LLM-formal verification integration
- Scaling to larger models

Limitations

- **Sampling-based validation:** No absolute soundness guarantee
- **Domain coverage:** Skewed toward algebra/number theory
- **LLM judge bias:** Evaluation relies on automated verifier
- **Single model family:** Experiments on Qwen-2.5 only

Contact: obozovmark9@gmail.com, michael.s.diskin@gmail.com