Curating Cold Start RL Data We constructed our initial training data through the following process:

1. We crawled problems from Art of Problem Solving (AoPS) contests [2], prioritizing math olympiads, team selection tests, and post-2010 problems explicitly requiring proofs, totaling 17,503 problems. This problem set is denoted as $\mathcal{D}_p$.
2. We generated candidate proofs using a variant of DeepSeek-V3.2-Exp-Thinking. As this model was not optimized for theorem proving and tended to produce concise but error-prone outputs, we prompted it to iteratively refine its proofs over multiple rounds to improve comprehensiveness and rigor.
3. We randomly sampled proofs across diverse problem types (e.g., algebra and number theory) and had mathematical experts score each proof according to the evaluation rubrics described above.

This process yielded an initial RL dataset $\mathcal{D}_v = \{(X_i, Y_i, s_i)\}$, where each item consists of a problem X_i, a proof Y_i, and an overall proof score $s_i \in \{0, 0.5, 1\}$.

RL Objective. Building on a version of DeepSeek-V3.2-Exp-SFT which was supervised fine-tuned on reasoning data related to mathematics and code, we trained the model with reinforcement learning to produce proof analyses using two reward components:

- **Format reward** R_{format}: An indicator function that enforces the model to generate both a summary of identified issues and a proof score, by checking whether the final response contains the key phrase "Here is my evaluation of the solution:" as well as a score within `\boxed{}` following "Based on my evaluation, the final overall score should be:".
- **Score reward** R_{score}: Rewards based on proximity between predicted score s_i' and annotated score s_i:

$$R_{\text{score}}(s_i', s_i) = 1 - |s_i' - s_i| \tag{1}$$

The RL objective for training the verifier is:

$$\max_{\pi_\varphi} \mathbb{E}_{(X_i, Y_i, s_i) \sim \mathcal{D}_v, (V_i', s_i') \sim \pi_\varphi(\cdot | X_i, Y_i)} \left[R_{\text{format}}(V_i') \cdot R_{\text{score}}(s_i', s_i) \right] \tag{2}$$

where V_i' denotes the verifier's final response and s_i' is the proof score extracted from it.

2.1.2. Introducing Meta-Verification to Review Proof Analyses

The approach described in Section 2.1.1 trains proof verification through RL to align predicted proof scores with expert annotations, but provides no direct supervision on the identified issues themselves. This creates a critical vulnerability: when evaluating flawed proofs (where $s_i < 1$) during training, the verifier can receive full reward by predicting the correct scores while hallucinating non-existent issues, undermining its trustworthiness.

To address this problem, we introduce **meta-verification**: a secondary evaluation process that assesses whether issues identified by the verifier indeed exist and whether these issues logically justify the predicted proof score according to the evaluation rubrics $\mathcal{I}_v$. The complete meta-verification rubrics $\mathcal{I}_{mv}$ are detailed in Appendix A.3.

[2] `https://artofproblemsolving.com/community/c13_contest_collections`