We trained a dedicated meta-verifier using RL to perform this evaluation. By incorporating the meta-verifier's feedback into verifier training, we can improve the faithfulness of the verifier's issue identification.

Meta-Verifier Training Process

1. We obtained an initial verifier π_φ following Section 2.1.1.
2. Mathematical experts scored the quality of verifier responses according to $\mathcal{I}_{mv}$, creating dataset $\mathcal{D}_{mv} = \{(X_i, Y_i, V_i, ms_i)\}$, where V_i is the analysis of proof Y_i and $ms_i \in \{0, 0.5, 1\}$ is the expert-annotated quality score.
3. We trained a meta-verifier $\pi_\eta(\cdot|X, Y, V, \mathcal{I}_{mv})$ to analyze the verifier's proof analysis V. The meta-verifier produces a summary of issues found in the analysis itself, followed by a quality score measuring how accurate and justified the verifier's analysis is. The RL objective follows the same structure as the verifier training, with format and score rewards.

Using the trained meta-verifier π_η, we enhanced the verifier training by integrating meta-verification feedback into the reward function:

$$R_V = R_{\text{format}} \cdot R_{\text{score}} \cdot R_{\text{meta}} \tag{3}$$

where R_{meta} is the quality score from the meta-verifier.

We trained the enhanced verifier on both the verification dataset $\mathcal{D}_v$ and the meta-verification dataset $\mathcal{D}_{mv}$, using the same reward mechanism on $\mathcal{D}_{mv}$ as used for training the meta-verifier. The resulting model can perform both proof verification and meta-verification tasks.

On a validation split of $\mathcal{D}_v$, the average quality score of the verifier's proof analyses – as evaluated by the meta-verifier – improved from 0.85 to 0.96, while maintaining the same accuracy in proof score prediction.

2.2. Proof Generation

2.2.1. Training a Generator for Theorem Proving

With verifier π_φ serving as a generative reward model, we train a proof generator $\pi_\theta(\cdot|X)$ with the RL objective:

$$\max_{\pi_\theta} \mathbb{E}_{X_i \sim \mathcal{D}_p, Y_i \sim \pi_\theta(\cdot|X_i)} [R_Y] \tag{4}$$

where R_Y is the proof score produced by $\pi_\varphi(\cdot|X_i, Y_i, \mathcal{I}_v)$.

2.2.2. Enhancing Reasoning via Self-Verification

When a proof generator fails to produce a completely correct proof in one shot – common for challenging problems from competitions like IMO and CMO – iterative verification and refinement can improve results. This involves analyzing the proof with an external verifier and prompting the generator to address identified issues.

However, we observed a critical limitation: when prompted to both generate and analyze its own proof in one shot, the generator tends to claim correctness even when the external verifier