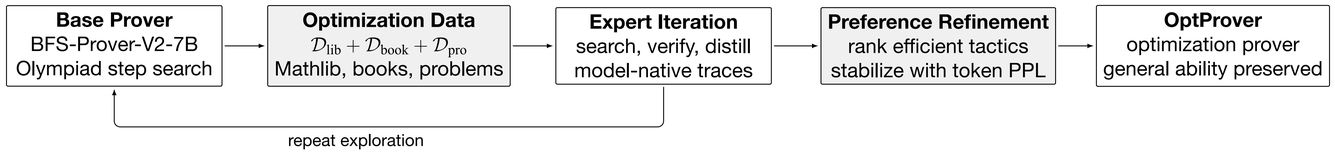
OptProver: Bridging Olympiad and Optimization through Continual Training in Formal Theorem Proving

Chenyi Li[1,*] Yanchen Nie[1,*] Zhenyu Ming[2] Gong Zhang[2] Kun Yuan[3] Zaiwen Wen[4]

[1]School of Mathematical Sciences, Peking University [2]Huawei Technologies
[3]Center for Machine Learning Research, Peking University [4]Beijing International Center for Mathematical Research, Peking University
*Equal contribution

Continual Training Pipeline

Takeaway: OptProver adapts an Olympiad-level step prover to undergraduate optimization by learning from verifier-confirmed search traces, then uses preference optimization to suppress valid but unhelpful tactics.



Why not naive fine-tuning?
Optimization proofs rely on domain-local definitions, convexity syntax, and algorithmic reasoning patterns that shift the tactic prior away from Olympiad proof search.

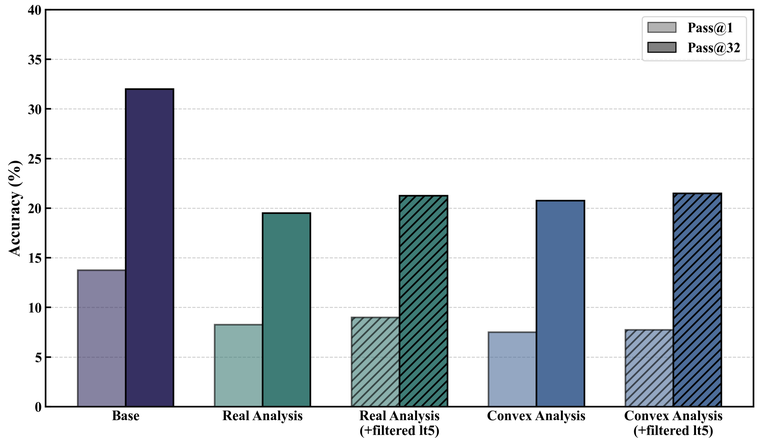
Why step-level proving?
Search observes intermediate proof states and Lean diagnostics, so local feedback can improve lemma choice, unfolding, rewriting, and tactic sequencing.

Why preference learning?
SFT learns valid tactics, but validity is not the same as utility. Preference data ranks verified proof-path actions above invalid or stagnant sibling branches.

Problem: Interactive Distribution Shift

Goal. Build a Lean 4 theorem prover for undergraduate optimization while preserving general Olympiad-level proving ability.

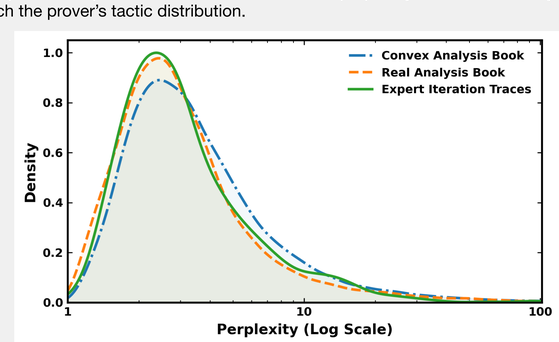
- Existing provers perform well on competition-style mathematics, but optimization requires specialized notions: convexity, subgradients, proximal operators, optimality, and algorithmic analysis.
- Direct continual training can reduce static perplexity while hurting interactive search, because high-probability tactics may be syntactically valid yet fail to advance a proof.
- We call this failure mode an **interactive distribution shift**: the learned tactic prior changes branch ordering and wastes verifier budget on dead-end subtrees.



Observation. Naive SFT on textbook-style optimization data degrades OptBench performance relative to the base prover, even after sequence-perplexity filtering.

Data Curation and Benchmark

OptBench: 400 undergraduate optimization problems spanning basic facts, convex analysis, and algorithmic proofs. Expert-iteration traces have lower perplexity than textbook-style proofs and better match the prover's tactic distribution.



Method I: Expert Iteration

Model-native supervision. OptProver lets the current policy search over formal optimization statements and keeps only verifier-accepted trajectories from $\mathcal{D}_{\text{lib}} \cup \mathcal{D}_{\text{book}} \cup \mathcal{D}_{\text{pro}}$.

- Search filters statements into tactics that are both correct and discoverable by the current model.
- EI creates a stable initialization before preference learning pushes the prover toward efficient proof search.

Method II: Utility-Aware and Perplexity-Weighted Preference Learning

Utility-aware preferences

- $u = 2$: tactic lies on a verified closed proof path.
- $u = 1$: tactic is valid but leads to a failed subtree.
- $u = 0$: tactic is invalid.

Proof-path tactics are ranked above stagnant or invalid sibling branches.

Perplexity-weighted DPO

$$w_t = \text{clip}\left(\left(\frac{\tau}{\text{PPL}_t + \epsilon}\right)^{\alpha}, \delta_{\min}, \delta_{\max}\right)$$

Token-level weighting suppresses unstable updates from high-perplexity regions while preserving reliable preference signals.

Combined objective. PW-UAPO merges both ideas: rank efficient verifier-confirmed actions above stagnant alternatives, and stabilize updates through reference-model perplexity.

Main Results on OptBench

55.25% best Avg. Pass@32 | **37.75%** best Avg. Pass@1 | **48.61%** algorithmic Pass@32

Step-level provers

Method	Pass@1	Pass@32
BFS-Prover-V2-7B	13.25	32.00
OptProver (EI)	28.75	50.00
OptProver (EI+DPO)	31.00	52.00
OptProver (EI+UAPO)	36.25	53.50
OptProver (EI+PW-DPO)	36.25	54.25
OptProver (EI+PW-UAPO)	**37.75**	**55.25**

Whole-proof baselines

Method	P@32	P@256
DeepSeek-Prover-V2-7B	14.25	18.75
Goedel-Prover-V2-8B	12.50	19.75
Kimina-Prover-7B	5.75	10.75

Reading. Step-level verifier feedback is crucial for optimization proofs; whole-proof generation remains brittle in this domain.

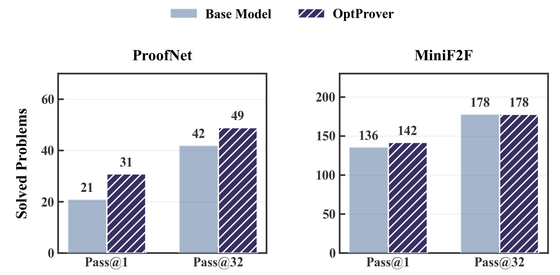
Interactive Search Reading

Mechanism. EI teaches model-native optimization tactics; PW-UAPO then uses verifier outcomes to rank branches by search utility, not just syntactic validity.

- **Correctness:** positives come from Lean-verified traces.
- **Usefulness:** proof-path actions beat stagnant valid siblings in the same search tree.
- **Stability:** PPL weights soften high-perplexity updates and protect general-domain behavior.

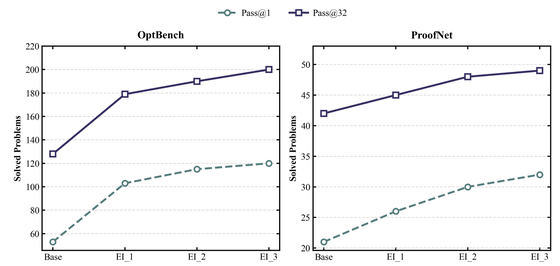
Robustness and Scaling

No catastrophic forgetting



MiniF2F remains stable and ProofNet improves under the same search budget.

Expert iteration saturates



EI saturates after the first rounds; preference learning provides the complementary jump to 55.25%.

Ablation and Category-Level Gains

Where the gain comes from

Method	Basic	Convex	Alg.
BFS P@32	36.36	44.44	16.67
EI P@32	49.59	60.00	40.97
PW-UAPO P@32	**55.37**	**62.22**	**48.61**
Gain over BFS	+19.01	+17.78	+31.94

- Algorithmic proofs benefit most because they contain long tactic chains where stagnant branches are costly.
- Convex proofs improve after EI aligns the model with domain-local definitions and rewriting conventions.

PW-DPO hyperparameter

Variant	Pass@1	Pass@32
$\tau = 1.00$	36.00	54.00
$\tau = 1.08$	**36.25**	**54.25**
$\tau = 2.42$	32.75	53.25

- The median token-level perplexity gives the best stability/learning trade-off.
- A large τ weakens high-PPL suppression and lets out-of-distribution tokens dominate updates.

Mechanism. Standard DPO penalizes rejected actions uniformly. OptProver instead separates invalid tactics from valid-but-stagnant tactics, then pushes probability mass toward verified, efficient proof-path actions under the same search budget.