BAIT: Benchmarking (Embedding) Architectures for Interactive Theorem-Proving

Sean Lamont[1,2], Michael Norrish[1], Amir Dezfouli[3], Christian Walder[4], Paul Montague[2]

[1]Australian National University, [2]Defence Science and Technology Group, [3]BIMLOGIQ, [4]Google DeepMind

Summary

We present BAIT, a platform to unify and accelerate research in AI for Interactive Theorem-Proving (AI-ITP). Using BAIT, we:

- Perform an in-depth comparison of modern embedding architectures over several ITP benchmarks
- Develop a novel End-to-End system for automated Interactive Theorem Proving (ITP), outperforming previous work

Motivation

- ITP systems are essential to formal verification
- Broad applications, from pure mathematics to critical software
- ITP applications typically require expert human guidance, limiting their scalability
- Recent work in AI-ITP has shown promise in applying AI to automate and assist ITP guidance
- AI-ITP results are fragmented, with many different approaches spread across several ITPs
- In particular, the Embedding architectures used across AI-ITP approaches have not been compared thoroughly

BAIT

- Implements the setup in Figure 1, which represents many AI-ITP approaches
- A modular design decouples the Search, Model, Environment and Data
- Shared checkpointing, logging and experiment management
- Streamlines the integration and comparison of components, with minimal boilerplate
- Facilitates reproducibility and transparency

AI-ITP Setup

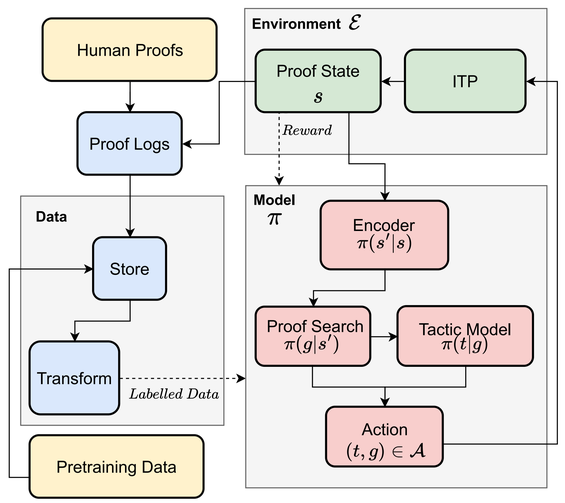

Figure 1:AI-ITP setup. A model π interacts with a proving environment $\mathcal{E}$, mapping a state s to an action (t, g), which defines tactic(s) t to apply to goal(s) $g \subseteq s$. π is trained with rewards or Data from processed Proof Logs, based on human proofs or agent-environment interactions. Implementing this modular framework, BAIT streamlines the integration and comparison of different approaches and benchmarks.

Experiments

Model	Cumulative	pass@1
GNN (Ours)	96.2%	**64.6%**
Transformer (Ours)	**96.8%**	63.3%
Original TacticZero	90.7%	43.0%

Table 1:Goals proven by TacticZero [1] in HOL4, after 1 attempt for validation and cumulatively for training. Experiments with the Embedding architecture result in large performance gains compared to the original.

Our experiments are over two categories:

- Supervised, with the task of predicting the tactic or premise used in a proof step. We compare Structure Aware Transformers (SAT) [2, 3], Graph Neural Networks (GNNs) [4], Transformer Encoders [5] and an Ensemble of GNN + Transformer over seven supervised ITP datasets
- End-to-End, where the agent interacts with a live environment and learns from self-play. We use the SoTA TacticZero [1] agent, with the associated HOL4 environment, and compare its original Encoder to GNN and Transformer Encoders

Embedding Comparison

Expression	GNN Encoder (Our Approach)	Original TacticZero
$\mathsf{diag}(A) = \mathsf{diag}(A^T)$	$R = (R^T)^T$	$\mathsf{FINITE}(\mathsf{POW}(s)) \Leftrightarrow \mathsf{FINITE}(s)$
$R\ x\ y \Rightarrow \mathsf{RC}(R)\ x\ y$	$\mathsf{RC}(\mathsf{RC}(R)) = \mathsf{RC}(R)$	$R\ x\ y \Rightarrow \mathsf{EQC}(R)\ x\ y$
$s \subseteq t \Leftrightarrow s \cup t = t$	$s\ \mathsf{DIFF}\ t = \emptyset \Leftrightarrow s \subseteq t$	$\mathsf{SURJ}\ f\ s\ t \Leftrightarrow \mathsf{IMAGE}\ f\ s = t$
$s \cup t = t \cup s$	$s \cup (t \cup u) = (s \cup t) \cup u$	$s \cap t = t \cap s$
$(s \cup t)x \Leftrightarrow x \in s \vee x \in t$	$x \in s \cup t \ \Leftrightarrow\ x \in s \vee x \in t$	$(s \cap t)x \Leftrightarrow x \in s \wedge x \in t$

Table 2:A selection of mathematical expressions (left) along with the nearest expression by cosine distance according to the TacticZero encoder (right) and our GNN encoder (center). Note the nearest neighbor as judged by the GNN model is generally far more semantically relevant to the original expression than the nearest neighbor as judged by the original TacticZero [1] Autoencoder.

Results

- For supervised benchmarks, SAT [2, 3], and Ensemble methods improve upon GNN [4] and Transformer Encoder [5] models, with SAT models performing the best overall.
- We reveal a significant improvement in the SoTA TacticZero [1] through experiments with the Embedding architecture (Table 1). We observed this was associated with more semantically relevant embeddings (Table 2)

References

[1] Minchao Wu, Michael Norrish, Christian Walder, and Amir Dezfouli. TacticZero: Learning to Prove Theorems from Scratch with Deep Reinforcement Learning. In *NeurIPS*, 2021.

[2] Dexiong Chen, Leslie O'Bray, and Karsten M. Borgwardt. Structure-aware transformer for graph representation learning. In *ICLR*, 2022.

[3] Yuankai Luo, Veronika Thost, and Lei Shi. Transformers over directed acyclic graphs. In *NeurIPS*, 2023.

[4] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise Selection for Theorem Proving by Deep Graph Embedding. In *NeurIPS*, 2017.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Links

- : sean-lamont.github.io/bait
- : github.com/sean-lamont/bait
- : sean.a.lamont@outlook.com

Acknowledgements

We would like to acknowledge Defence Science and Technology Group (DSTG) for their support in this project. We would also like to thank Minchao Wu for his help with the TacticZero source code.