Make E Smart Again *

Zarathustra Amadeus Goertzel, Jan Jakubův, and Josef Urban

Czech Technical University in Prague, Prague

Making E Stupid and Then Smart Again The ENIGMA [4, 5, 6, 3] system with the XGBoost [1] implementation of gradient boosted decision trees has recently shown high capability to learn guiding the E [8] prover's inferences in real time. In particular, after several proving and learning iterations, its performance on the 57897 problems from the Mizar40 [7] benchmark improved by 70% (= 25397/14933) [6] over the good E strategy used for the initial proving phase. This good strategy uses many sophisticated *clause evaluation functions*, the Knuth-Bendix ordering (KBO6), and other E heuristics.

In this work we strip E to the bare bones: KBO6 is replaced with an identity relation as the minimal possible ordering (an addition to E), the strategy is replaced with the simple combination of the clause weight and FIFO (first in first out) evaluation functions, and literal selections are disabled. Literal selection is important because by limiting the literals used in inference, E can generate far fewer clauses and avoid redundant inferences. E is thus practically reduced to a basic resolution prover with some rewriting capabilities. We call this strategy E0:

```
--definitional-cnf=24 --prefer-initial-clauses -tIDEN --restrict-literal-comparisons
 -WNoSelection -H'(5*Clauseweight(ConstPrio,1,1,1),1*FIFOWeight(ConstPrio))'
```

E0 solves only 3872 of the Mizar40 problems in 5 seconds. The task for ENIGMA with this basic prover is to learn the ATP guidance completely on its own, i.e., we explore how smart E can become using machine learning in this *zero-strategy* setting. The more general related question is to what extent can machine learning replace the sophisticated human-invented theorem-proving body of wisdom used in today's ATPs for restricting advanced proof calculi.

Learning Experiments We evaluate ENIGMA with the basic strategy, E0, in several scenarios and over two datasets of different size. All experiments are run with 5 seconds per problem.[1] ENIGMA has so far been used in two ways: *coop* combines the learned advice with some standard E strategy equally (50 : 50) [2] while *solo* only uses the learned ENIGMA model for choosing the given clauses. The best results have been achieved by looping: that is, an ENIGMA model loop 0 is trained and run with E (loop 0), then the resulting data are added to the initial training data and a new ENIGMA model is trained (loop 1).

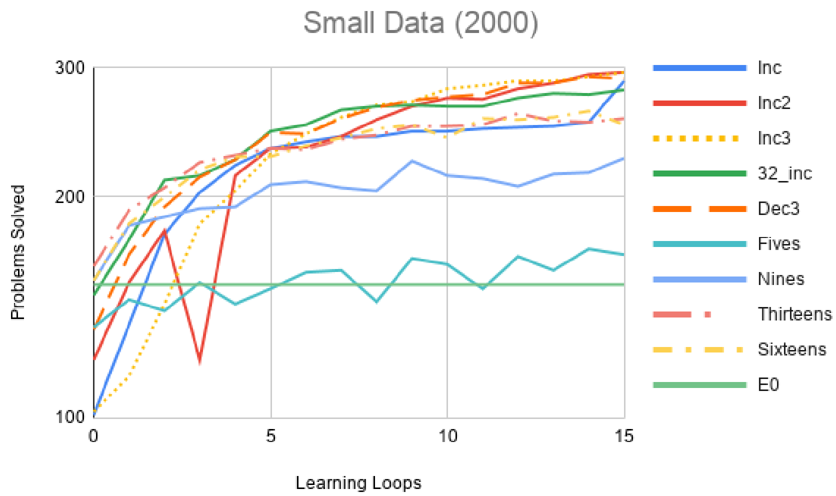
In this work we train with both *solo* and *coop*, and only present results from *solo* runs because they represent the most minimal setting.

Small Data (2000 problems): The E evaluations and XGBoost training can take a long time on the full Mizar40 dataset, so we randomly sampled 2000 problems to test hyper-parameters on. Each XGBoost model consists of T decision trees of depth D, the most important training meta-parameters. In previous work T and D were fixed for all loops of learning. Here we try vary the values during 15 loops. Let $S_{D,T}$ denote the experiment with specific T and D. The following results are included in the plot of solved problems (above right): *Fives* ($S_{5,100}$), *Nines* ($S_{9,100}$), *Thirteens* ($S_{13,200}$), *Sixteens* ($S_{16,100}$).

*Supported by the ERC Consolidator grant no. 649043 AI4REASON and by the Czech project AI&Reasoning CZ.02.1.01/0.0/0.0/15_003/ 0000466 and the European Regional Development Fund.

[1] Almost all the experiments are run on the same hardware: Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz with 188GB RAM.

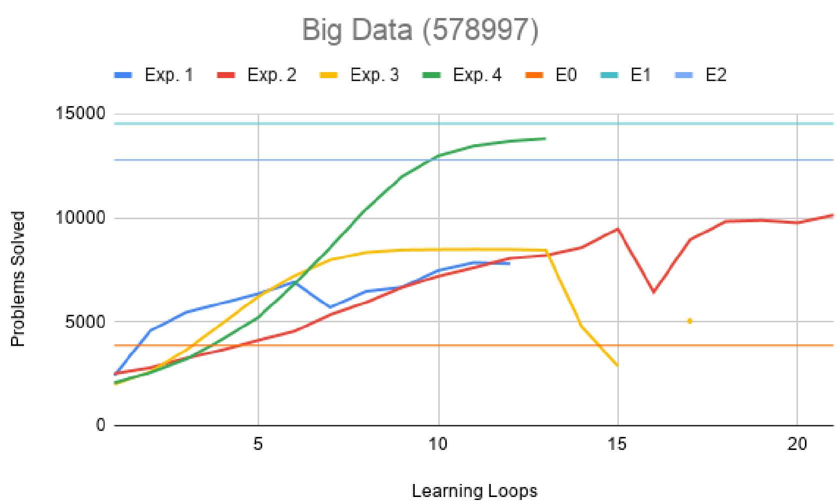
[2] This means that half of the given clauses are selected by the E strategy and half by ENIGMA's model.



We also experiment with adaptively setting the hyper-parameters as the number of training examples increases. Protocol *Inc* ($S_{[3,33],100}$) increases D by 2 from 3 to 33 and keeps $T = 100$ fixed. Protocol *32_inc* ($S_{32,[50,250]}$) fixes $D = 32$ and gradually increases T from 50 to 250. Protocol *Inc2* ($S_{[3,33],*}$) gradually decreases T from 150 to 50, varying the value intuitively. Protocol *Inc3* ($S_{[3,33],[50,250]}$) aims to be more systematic and steps T from 50 to 250, and Protocol *Dec3*($S_{[3,33],[250,50]}$) decreases T from 250 to 50.

At the 15th loop *Inc* is best solving 299 problems, doubling the performance of E0 (152). *Inc2* and *Inc3* solve 298 problems for second, and *32_inc* takes third place at 291 problems. The conclusion is that simple protocols work well so long as T or D is incremented adaptively rather then fixed.

Big Data (57897 problems): The experiments are done on a large benchmark of 57897



Mizar40 [7] problems from the MPTP dataset [9]. E1 and E2 are two strong E strategies solving 14526 and 12788 problems.

- **Experiment 1** is done with $D = 9$ and $T = 200$ and uses our previously trained model, which allowed us to solve solve 25562 problems when cooperating with E1 in our previous experiments [6]. This strong model, which hashes the features into 65536 (2^{16}) buckets [2, Sec. 3.4], is used with E0 now.

- **Experiment 2**'s parameters were intuitively toggled during the looping as in *Inc3*. Exp. 2 uses training data from E1 and E2 for additional guidance up to the 4th loop (and then stops including them in the training data on the assumption they may confuse learning).

- **Experiment 3** sets T and S according to protocol *Inc3*. Exp. 3 only learns from E run with E0 and trains on the GPU, which requires the feature size to be reduced to 256.

- **Experiment 4** mimics Exp. 3 but uses E1 and E2 data for training (up to the 4th loop).

The strong model does not help much in guiding E without ordering or selection in Exp. 1. Exp. 2 learns gradually and catches up with Exp. 1, but seems to plateau around $10,000$. Surprisingly the pure Exp. 3 learns fast with the small feature size, but plateaus and drops in performance (perhaps due to overfitting). Exp. 4 indicates that guidance is useful and surpasses E2 with 13805 in round 13. This is a great improvement over the 3872 problems solved by E0.

Conclusion ENIGMA can learn to guide the E prover effectively even without smart strategies and term orderings. The models confer a 256% increase over the naive E0 after 13 rounds of the proving/learning loop, and even trained without guidance data, a 121% increase. The experiments indicate that machine learning can be used to fully control an ATP's guidance, learning to replace orderings, heuristic strategies, and deal with the increase in generated clauses without literal selection. Given the large symmetry-breaking impact of these methods in classical ATP, future work includes, e.g., training the guidance in such a way that redundant (symmetric) inferences are not done by the trained model once it has committed to a certain path. This probably means equipping the learning with more history in the saturation-style setting.

Running ENIGMA without term ordering and other restrictions is important because it allows us to combine training data from different strategies. We aim at combining several strategies into one with a performance comparable to their parallel execution. Increasing parameters S and T with training loops also seems promising and as it outperforms static values we plan to investigate it further.

References

[1] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[2] Karel Chvalovský, Jan Jakubuv, Martin Suda, and Josef Urban. ENIGMA-NG: efficient neural and gradient-boosted inference guidance for E. *CoRR*, abs/1903.03182, 2019.

[3] Zarathustra Goertzel, Jan Jakubuv, and Josef Urban. Enigmawatch: Proofwatch meets ENIGMA. In Serenella Cerrito and Andrei Popescu, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 28th International Conference, TABLEAUX 2019, London, UK, September 3-5, 2019, Proceedings*, volume 11714 of *Lecture Notes in Computer Science*, pages 374–388. Springer, 2019.

[4] Jan Jakubuv and Josef Urban. ENIGMA: efficient learning-based inference guiding machine. In Herman Geuvers, Matthew England, Osman Hasan, Florian Rabe, and Olaf Teschke, editors, *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*, volume 10383 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 2017.

[5] Jan Jakubuv and Josef Urban. Enhancing ENIGMA given clause guidance. In Florian Rabe, William M. Farmer, Grant O. Passmore, and Abdou Youssef, editors, *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*, volume 11006 of *Lecture Notes in Computer Science*, pages 118–124. Springer, 2018.

[6] Jan Jakubuv and Josef Urban. Hammering Mizar by learning clause guidance. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving, ITP 2019, September 9-12, 2019, Portland, OR, USA*, volume 141 of *LIPIcs*, pages 34:1–34:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[7] Cezary Kaliszyk and Josef Urban. MizAR 40 for Mizar 40. *J. Autom. Reasoning*, 55(3):245–256, 2015.

[8] Stephan Schulz. System description: E 1.8. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *LPAR*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.

[9] Josef Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reasoning*, 37(1-2):21–43, 2006.