

A Decade of ATP Tuning: Machine Learning, Strategy Invention, and AI in the Future^{*}

Jan Jakubův

Czech Technical University in Prague, Prague, Czechia

This talk summarizes our progress and experience with fine-tuning Automated Theorem Provers (ATP) and Satisfiability Modulo Theories (SMT) solvers over the past ten years. Both ATP and SMT solvers are used in a wide range of applications and are utilized by various Interactive Theorem Provers (ITP) for automated discharging of trivial (or less trivial) proofs.

The majority of available ATP and SMT solvers implement a large set of various techniques, which are parametrized by numerous options. Correctly setting these options is quite often the key to good performance, and the right combination also depends on the specific problems in question. We have approached the problem from three complementary directions: we can *learn inside* the solver and guide its individual decisions, we can *search outside* the solver and invent whole strategies offline, and we can *choose per problem* which strategy to run. All three rely on data from previous solver runs, and they can be naturally combined.

Our first line of work targets the given clause loop in saturation-based provers, where the prover repeatedly selects the next clause to process. After hand-crafted heuristics based on similarity to the conjecture [16], ENIGMA [18] replaced guessing by learning: clauses from previous proofs of the E prover serve as positive examples, while the other unnecessarily processed clauses serve as negatives. The trained model then guides future given clause selection. ENIGMA was later extended with more sophisticated features [19], gradient-boosted trees and neural networks [3], symbol-independent graph neural networks [9,2], and parental guidance [4]. Since the model is queried inside the innermost loop, its speed matters as much as its accuracy. ENIGMA improved E by 70% on the Mizar library in a single-strategy setting [21,8], and by 25% on Isabelle Sledgehammer problems, outperforming all previous ATP and SMT systems [5]. Complementary gains were obtained by Deepire [23] in Vampire. Our most recent work, presented in the main track, extends both to native higher-order guidance for Isabelle/HOL.

Our second line of work invents whole strategies without any internal modifications of the solver. Classical algorithm configuration, like ParamILS [7], looks for a single best configuration, but for ATPs a portfolio of complementary strategies is far more valuable. Following BliStr [24], which repeatedly specializes strategies on the problems they are already good at, we developed BliStr-Tune [17,20] for hierarchical E strategies, EmpireTune [15] for both E and Vampire, which also invents term orderings [14], and finally Grackle [6], their generic,

^{*} Supported by Renaissance Philanthropy, project DEEPER, and the Czech Science Foundation, project no. 24-12759S.

solver-independent successor. On the Bitwuzla SMT solver, Grackle raised the number of solved SMT-COMP problems from 1345 to 1500. On Prover9 [1], it more than doubled the number of solved AIM problems (from 41 to 91), outperforming both E and Vampire there. Invented strategies are often surprising and hard to explain, but they consistently beat the hand-made ones.

In SMT, we focus on the theory of uninterpreted functions (UF), where ground reasoning is decidable and fast, and the difficulty lies in quantifier instantiation. After first modeling the choice of quantifiers as a multi-armed bandit problem [12], we returned to the ENIGMA-style approach and guided the instantiation loop of *cvc5* by gradient-boosted trees [10,11]. This improves *cvc5* on Mizar by more than 20%, and our best learned strategy alone outperforms the CASC portfolio of *cvc5* by more than 10%. A neural variant [22] is more accurate but again pays in speed. Combining *cvc5*, invented *cvc5* strategies, and classification variants, we solved 3021 of 14163 hard Mizar problems, raising the ATP coverage of Mizar from 75% to over 80% [13].

LLM coding agents can now write code that compiles, runs, and is not obviously wrong. Rather than asking whether an LLM can prove theorems, we asked whether it can build the tool that proves them. Primo is an SMT solver for linear real arithmetic (QF_LRA) written entirely by a coding agent, with the human choosing the target, supplying the literature, and insisting on proper evaluation. The agent also rewrote ParamILS in Rust as RamParILS and extracted the parameter space of Primo from its option parser. Tuned Primo solves 1599 of 1753 SMT-LIB QF_LRA instances in 30 seconds, more than OpenSMT and Yices. Agents excel at implementing known algorithms and tireless optimization, but they are poor judges of results that are too good to be true: the scarce resource has moved from writing code to judging results.

Looking back at the ten years, we see that default settings are only a local optimum: every solver we touched improved substantially by tuning alone. Complementarity beats individual strength, iteration pays off, and the best results always combine several techniques. Less comfortably, we rarely know why an invented strategy works, and we do not yet fully understand why some benchmarks respond to learning better than others. For example, methods that work well on Mizar give much smaller gains on TPTP, SMT-COMP, or Isabelle. We conclude with an outlook on native higher-order reasoning, configuration on the fly, and agents taking part in the experimental loop.

References

1. Aleksandrova, K., Jakubův, J., Kaliszyk, C.: Prover9 unleashed: Automated configuration for enhanced proof discovery. In: LPAR 2024. EPIc, vol. 100, pp. 360–369. EasyChair (2024). <https://doi.org/10.29007/SD6T>
2. Chvalovský, K., Jakubův, J., Olšák, M., Urban, J.: Learning theorem proving components. In: TABLEAUX 2021. LNCS, vol. 12842, pp. 266–278. Springer (2021). https://doi.org/10.1007/978-3-030-86059-2_16

3. Chvalovský, K., Jakubův, J., Suda, M., Urban, J.: ENIGMA-NG: efficient neural and gradient-boosted inference guidance for E. In: CADE 2019. LNCS, vol. 11716, pp. 197–215. Springer (2019). https://doi.org/10.1007/978-3-030-29436-6_12
4. Goertzel, Z.A., Chvalovský, K., Jakubův, J., Olšák, M., Urban, J.: Fast and slow Enigmas and parental guidance. In: FroCoS 2021. LNCS, vol. 12941, pp. 173–191. Springer (2021). https://doi.org/10.1007/978-3-030-86205-3_10
5. Goertzel, Z.A., Jakubův, J., Kaliszyk, C., Olšák, M., Piepenbrock, J., Urban, J.: The Isabelle ENIGMA. In: ITP 2022. LIPIcs, vol. 237, pp. 16:1–16:21. Schloss Dagstuhl (2022). <https://doi.org/10.4230/LIPICS.ITP.2022.16>
6. Hůla, J., Jakubův, J., Janota, M., Kubej, L.: Targeted configuration of an SMT solver. In: CICM 2022. LNCS, vol. 13467, pp. 256–271. Springer (2022). https://doi.org/10.1007/978-3-031-16681-5_18
7. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: An automatic algorithm configuration framework. *J. Artif. Intell. Res.* **36**, 267–306 (2009). <https://doi.org/10.1613/JAIR.2861>
8. Jakubův, J., Chvalovský, K., Goertzel, Z.A., Kaliszyk, C., Olšák, M., Piotrowski, B., Schulz, S., Suda, M., Urban, J.: MizAR 60 for Mizar 50. In: ITP 2023. LIPIcs, vol. 268, pp. 19:1–19:22. Schloss Dagstuhl (2023). <https://doi.org/10.4230/LIPICS.ITP.2023.19>
9. Jakubův, J., Chvalovský, K., Olšák, M., Piotrowski, B., Suda, M., Urban, J.: ENIGMA anonymous: Symbol-independent inference guiding machine (system description). In: IJCAR 2020. LNCS, vol. 12167, pp. 448–463. Springer (2020). https://doi.org/10.1007/978-3-030-51054-1_29
10. Jakubův, J., Janota, M., Piepenbrock, J., Urban, J.: Machine learning for quantifier selection in cvc5. In: ECAI 2024. FAIA, vol. 392, pp. 4336–4343. IOS Press (2024). <https://doi.org/10.3233/FAIA241009>
11. Jakubův, J., Janota, M., Piepenbrock, J., Urban, J.: Machine learning for quantifier selection in cvc5. *Int. J. Approx. Reason.* **189**, 109602 (2026). <https://doi.org/10.1016/J.IJAR.2025.109602>
12. Jakubův, J., Janota, M., Piotrowski, B., Piepenbrock, J., Reynolds, A.: Selecting quantifiers for instantiation in SMT. In: SMT 2023. CEUR, vol. 3429, pp. 71–77. CEUR-WS.org (2023)
13. Jakubův, J., Janota, M., Urban, J.: Solving hard Mizar problems with instantiation and strategy invention. In: CICM 2024. LNCS, vol. 14960, pp. 315–333. Springer (2024). https://doi.org/10.1007/978-3-031-66997-2_18
14. Jakubův, J., Kaliszyk, C.: Towards a unified ordering for superposition-based automated reasoning. In: ICMS 2018. LNCS, vol. 10931, pp. 245–254. Springer (2018). https://doi.org/10.1007/978-3-319-96418-8_29
15. Jakubův, J., Suda, M., Urban, J.: Automated invention of strategies and term orderings for Vampire. In: GCAI 2017. EPiC, vol. 50, pp. 121–133. EasyChair (2017). <https://doi.org/10.29007/XGHJ>
16. Jakubův, J., Urban, J.: Extending E prover with similarity based clause selection strategies. In: CICM 2016. LNCS, vol. 9791, pp. 151–156. Springer (2016). https://doi.org/10.1007/978-3-319-42547-4_11
17. Jakubův, J., Urban, J.: BliStrTune: hierarchical invention of theorem proving strategies. In: CPP 2017. pp. 43–52. ACM (2017). <https://doi.org/10.1145/3018610.3018619>
18. Jakubův, J., Urban, J.: ENIGMA: efficient learning-based inference guiding machine. In: CICM 2017. LNCS, vol. 10383, pp. 292–302. Springer (2017). https://doi.org/10.1007/978-3-319-62075-6_20

19. Jakubův, J., Urban, J.: Enhancing ENIGMA given clause guidance. In: CICM 2018. LNCS, vol. 11006, pp. 118–124. Springer (2018). https://doi.org/10.1007/978-3-319-96812-4_11
20. Jakubův, J., Urban, J.: Hierarchical invention of theorem proving strategies. *AI Commun.* **31**(3), 237–250 (2018). <https://doi.org/10.3233/AIC-180761>
21. Jakubův, J., Urban, J.: Hammering Mizar by learning clause guidance (short paper). In: ITP 2019. *LIPICs*, vol. 141, pp. 34:1–34:8. Schloss Dagstuhl (2019). <https://doi.org/10.4230/LIPICs.ITP.2019.34>
22. Piepenbrock, J., Janota, M., Urban, J., Jakubův, J.: First experiments with neural cvc5. In: LPAR 2024. *EPiC*, vol. 100, pp. 264–277. EasyChair (2024). <https://doi.org/10.29007/H5DR>
23. Suda, M.: Vampire with a brain is a good ITP hammer. In: FroCoS 2021. LNCS, vol. 12941, pp. 192–209. Springer (2021). https://doi.org/10.1007/978-3-030-86205-3_11
24. Urban, J.: BliStr: The Blind Strategymaker. In: GCAI 2015. pp. 312–319 (2015)