# A Closer Look at Successful Clause Derivations Through the Lens of Recursive Neural Networks[*]

Martin Suda

Czech Technical University in Prague, Czech Republic

## 1 Motivation

Deepire [14,15] is an extension of the automatic theorem prover (ATP) Vampire [10] by machine-learned ENIGMA-style clause selection guidance [2, 6, 7, 11]. Its main distinguishing feature is the use of a recursive neural network (RvNN) to classify clauses based solely on their derivation history. This means that to decide whether a clause should be preferred in proof search, Deepire does not look at the logical content of the clause as a formula, but only at its ancestors in the derivation DAG and the inference rules that were applied to derive it.

Despite the simplicity of the approach (and its inherent inability to provide "the perfect guidance", even in principle), Deepire has substantially improved on plain Vampire's performance on (1) theory reasoning problems coming from SMT-LIB [1], see [14], and on (2) formal library export problems from the Mizar40 set [9], see [15]. On the latter benchmark, Deepire even improved on the impressive results of ENIGMA by Jakubův and Urban from 2019 [8].

Obviously, these successes required a certain amount of tuning. In particular, one needs to find the right balance between the capacity of the network and the time it takes to evaluate it. (On the tested benchmarks, Deepire worked the best with clause embedding dimension between 64 and 128, spending on average between 30 and 40 % of the prover runtime evaluating the network.) Additionally, it is also important to select a good mixture of the traditional heuristics for governing clause selection and the machine-learned advice. (Deepire pioneers the use of the layered clause selection scheme [4, 5, 16] for this, in which the traditional selection by clause's age and weight is preserved but alternately applied to only the clauses classified as positive by the network and to all the passive clauses; this alternation happens under a ratio, for which our experiments established an optimal value of 2:1.)

Ultimately, however, tuning notwithstanding, given how useful it can be for guiding the prover, a trained network represents an interesting artifact that, I believe, should be further analyzed. The aim of this work is therefore to conduct an analysis of the best neural models obtained in our previous work on Deepire [14,15] and to shed more light on the reasons behind the success of the strategies these models back up.

## 2 The Aims

Besides simply satisfying intellectual curiosity, our main aim with the proposed analysis is to look for *general theorem proving heuristics* that the training process implicitly discovered and that could be extracted, understood by a human and adapted for the design of better theorem proving strategies on other benchmarks. An example of the kind of a heuristic I have in mind could be the theory distance heuristic by Gleiss and Suda [5], whose computation proceeds

[*] Supported by the Czech Science Foundation project 20-06390Y and the project RICAIP no. 857306 under the EU-H2020 programme.

along a clause derivation in analogy to evaluation of Deepire's RvNN. However, theory distance was proposed *before* the Deepire experiments on SMT-LIB took place and it is at the moment not clear to what degree Deepire's network exploits the principle behind theory distance.

It can turn out, though, that no such heuristics are easily identifiable or even present in any form. That would mean that the knowledge extracted by the network pertains exclusively to the specific benchmark it was trained for. We might then hope to learn what were the properties of that benchmark that Deepire exploited to tackle it well. Note that simple memorization cannot fully explain the observed successes as there was always a significant jump between the number of problems solved by plain Vampire (whose solutions were used to train the first model) and the performance of Deepire using the first model, which suggests successful generalization.

# 3 The Techniques

In the experiment on Mizar [15], the network training procedure effectively compressed 800 MB (of disk space when zipped) worth of successful derivations into a 5 MB torch-script model file (consisting mostly of matrix and vector parameters). Five megabytes is definitely too much to be directly approached and analyzed manually. Therefore, I propose to use statistical techniques, be it ad hoc ones, tailored for the particular use case, or out-of-the-box solutions marketed under the label of explainable AI.

**The "generalized age" perspective:** Any clause selection heuristic that works as a function of clause's derivation history can be understood as generalizing the clause's *age*. (At least the way it is defined in Vampire, i.e., as the depth of the derivation DAG (only counting generating inferences and not reductions)). We can study to what degree is the evaluation function represented by the learned RvNN similar to the age function; for instance, asking:

- Is the evaluation function (most of the time) monotone along the derivations?[1]
- Does the network take into account the exact shapes of the trees[2] or is it (mostly) additive?

Since the intuition behind the age heuristic is that clauses with more complex derivation DAGs are less likely to contribute to a proof and since the training set for our network was, after all, finite, an interesting question also arises, namely whether the network allows for a positively classified clause of arbitrary large derivation. I will try to answer these questions in the talk.

**Visualisations:** A picture is worth a thousand words. Using the Graphviz library [3], I wrote a tool for visualising Vampire's derivations and labelling the nodes corresponding to clauses by the RvNN's classification judgments. Figure 1 show an example output. So far I was not able to glimpse any revealing pattern in these, but they seem to have a certain artistic value.

**XAI:** What I have just described, i.e., the "quest for opening a black box", seems to be a perfect case for the application of the techniques of explainable artificial intelligence. I am currently investigating whether there are methods and tools readily available to help explaining RvNNs or how to adapt the popular methods such as LIME [13] or SHAP [12] to analyze networks coming from the Deepire setting. As part of my talk, I am planning to give a review of the most relevant XIA methods and establish to what degree these methods, coming predominantly from the computer vision field, can be useful in our case.

[1] Note that the training examples all have a property that the parent of a positive clause is a positive clause.

[2] Although computed on a DAG, mathematically, the evaluation is a function of the unfolded tree.

# References

[1] C. Barrett, P. Fontaine, and C. Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). www.SMT-LIB.org, 2016.

[2] K. Chvalovský, J. Jakubuv, M. Suda, and J. Urban. ENIGMA-NG: efficient neural and gradient-boosted inference guidance for E. In *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, vol. 11716 of *LNCS*, pp. 197–215. Springer, 2019.

[3] J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz and dynagraph – static and dynamic graph drawing tools. In *GRAPH DRAWING SOFTWARE*, pp. 127–148. Springer-Verlag, 2003.

[4] B. Gleiss and M. Suda. Layered clause selection for saturation-based theorem proving. In *Proceedings of the 7th Workshop on Practical Aspects of Automated Reasoning (PAAR), co-located with the (IJCAR 2020), Paris, France, June-July, 2020 (Virtual)*, vol. 2752 of *CEUR Workshop Proceedings*, pp. 34–52. CEUR-WS.org, 2020.

[5] B. Gleiss and M. Suda. Layered clause selection for theory reasoning - (short paper). In *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I*, vol. 12166 of *LNCS*, pp. 402–409. Springer, 2020.

[6] J. Jakubuv, K. Chvalovský, M. Olsák, B. Piotrowski, M. Suda, and J. Urban. ENIGMA anonymous: Symbol-independent inference guiding machine (system description). In *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II*, vol. 12167 of *LNCS*, pp. 448–463. Springer, 2020.

[7] J. Jakubuv and J. Urban. ENIGMA: efficient learning-based inference guiding machine. In *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*, vol. 10383 of *LNCS*, pp. 292–302. Springer, 2017.

[8] J. Jakubuv and J. Urban. Hammering Mizar by learning clause guidance (short paper). In *10th International Conference on Interactive Theorem Proving, ITP 2019, September 9-12, 2019, Portland, OR, USA*, vol. 141 of *LIPIcs*, pp. 34:1–34:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[9] C. Kaliszyk and J. Urban. Mizar 40 for mizar 40. *J. Autom. Reason.*, 55(3):245–256, 2015.

[10] L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, vol. 8044 of *LNCS*, pp. 1–35. Springer, 2013.

[11] S. M. Loos, G. Irving, C. Szegedy, and C. Kaliszyk. Deep network guided proof search. In *LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, May 7-12, 2017*, vol. 46 of *EPiC Series in Computing*, pp. 85–105. EasyChair, 2017.

[12] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4765–4774, 2017.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144. ACM, 2016.

[14] M. Suda. Improving ENIGMA-style clause selection while learning from history. In *Proceedings of the 28th CADE*, 2021. To appear. See also https://arxiv.org/abs/2102.13564.

[15] M. Suda. Vampire with a brain is a good ITP hammer. *CoRR*, abs/2102.03529, 2021.

[16] T. Tammet. GKC: A reasoning system for large knowledge bases. In *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, vol. 11716 of *LNCS*, pp. 538–549. Springer, 2019.

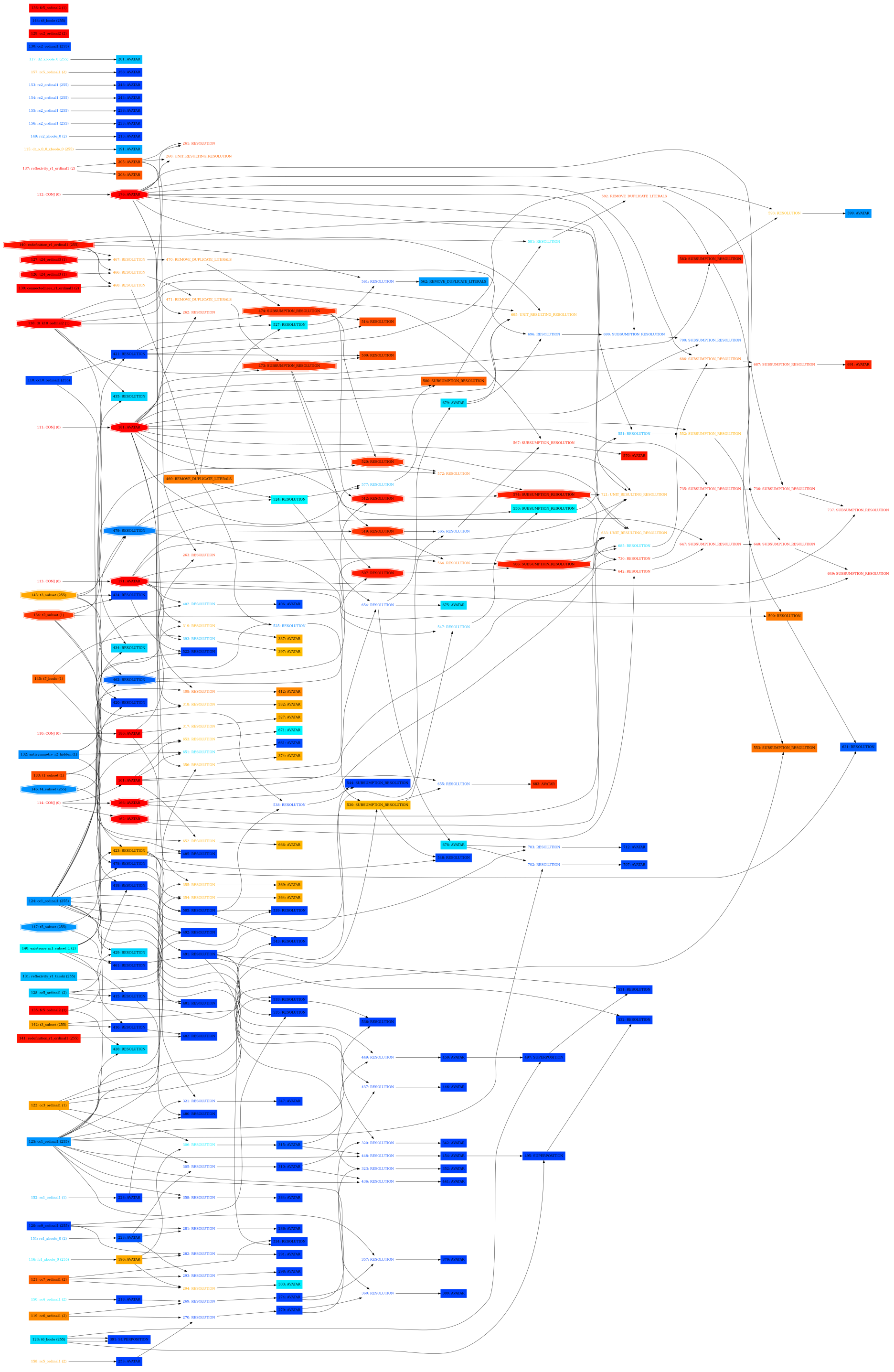

Figure 1: An example derivation from the Mizar benchmark (on the problem `t25_ordinal3`) evaluated by the network. Octagons mark clauses from the original proof, rectangles the remaining selected clauses, and labels without a box denote clauses that were never selected (the most common reason being immediate reductions). Hues of blue mark clauses classified as negative, and hues of orange and red those classified as positive. Since AVATAR was used in the proof, there is no final empty clause explicitly present (the ultimate contradiction was derived in the SAT solver). The derivation contains both false positives and false negatives.