

---

# tnreason FOR LLM REASONING

---

RESEARCH NOTES IN THE ENEXA PROJECT

February 14, 2025

We here sketch, how the usage of tnreason can be enhanced based on LLMs, and how tnreason could be exploited in the retraining of LLMs.

## 1 Copilots for the usage of tnreason

We envision users to provide prompts describing their reasoning objectives, which the LLM interprets to generate precise tnreason function calls, ensuring a natural and efficient workflow. By leveraging LLMs' natural language understanding, users can describe complex reasoning tasks intuitively, allowing the copilot to translate them into structured calls to tnreason .

### 1.1 Model explainability

Having a tnreason model described in the formal script language, a LLM can be prompted to explain that model to an user. Besides datapoint specific inquiries, such a copilot can be used to explain entire parts of the model, for example based on a collection of rules triggered when having a specific input feature. The envisioned explainability therefore goes beyond the typical "post-hoc" explainability towards an intrinsic "ante-hoc" explainability bringing guarantees for a models behavior.

### 1.2 Prompt structure

The systems prompt will describe the basic functionality of tnreason and the users prompt consists in the requirements of the use case, formulated in natural language. Here the user does not have to be informed about the formal grammar of the tnreason script language, nor the structure and expressivity of propositional logics. This approach enhances accessibility of tnreason , enabling seamless interaction between human intuition and formal logical computation.

Preliminary experiments with the Gemini model on Colab can be found here: <https://colab.research.google.com/drive/1ESzJeZp7O022I1z31NeA8VExALUVfUf7?usp=sharing>

### 1.3 Multi-agent Orchestration

Another interesting idea is a multi-agent orchestration, where the ability of LLMs and formal proof languages are balanced with respect to each other. Here one might exploit the LLMs as a translator between human understandable text and formal statements such as in the tnreason script language.

- **Communication agent (LLM):** Interacts with the user, to create and maintain a tnreason model. Further, when asked about the models behavior at a specific data instance or in a more generic situation, it interprets the model in human understandable language. At best realized by a LLM to handle the flexible user-specific prompts.
- **Execution agent (tnreason ):** Maintains a model about a use case in cooperation with the communication agent, and executes the inference calls from the data agent. At best realized in an efficient and explainable manner and therefore in a formal, declarative language such as the tnreason script language.
- **Data agent (Classical Software):** Handles the runtime of a productive model, by transforming the input data into an inference instance, passing it to the execution agent, and preparing the output in a specific format.

## 2 Retraining LLMs based on a reinforcement environment created by `tnreason`

Retraining LLMs in a reinforcement learning environment incorporating `tnreason` workloads enables adaptive improvement in logical reasoning and decision-making. The model receives feedback based on the correctness and efficiency of its generated `tnreason` calls, refining its ability to solve complex reasoning tasks over time. This iterative training approach enhances the LLM’s alignment with formal reasoning principles, leading to more reliable and interpretable outputs.

One can pose `tnreason` inference models, such as probabilistic queries of formulas in a hybrid knowledge base generated from random, as a prompt and calculate the reward as the absolute difference of the exact calculation by `tnreason` and the response from the LLM.

## 3 Ideas to Monte Carlo Tree Searches

Monte Carlo Tree Search (MCTS) is a classical method in AI to solve adversarial games. Recent work [Besta et al. \(2025\)](#); [Xin et al. \(2024\)](#) integrate them into the architecture of LLMs to enhance reasoning capabilities.

### 3.1 Parametrizing a Search Tree by a Formula Selecting Network

Search Trees in combination with Monte Carlo tree searches MCTS received recent popularity as a mechanism to advanced reasoning functionalities of LLMs.

In the `tnreason` framework, several formula selecting networks (and other selection architectures such as slice selection) follow a similar idea. Instead of branches, we typically depict choices by assignments to selection variables. Thus, a formula selecting network parametrizes formulas, which are in the expressivity of formula search trees, where the branches along each layer of the tree have a common structure (such as selection of a logical connective or selection of a categorical variable).

### 3.2 Solving mode search problems by MCTS

Mode search problems have been a central class of problems in the main documentation, ranging from MAP inference of Markov Logic Networks to the grafting heuristic in greedy learning. So far, we have investigated methods from variational inference as well as reformulations into known optimization formats (HUBO) to solve mode search problems. MCTS provide alternative ways to solve such search problems, which follow a sampling based approach and make use of the search tree parametrization in the section above. Let us note, that while MCTS could operate on any tree representing a search space, we here have specific trees with constant branching factors to the nodes in each layer, and with each leaf corresponding with a coordinate.

The decisions, which have to be done iteratively, are assignments to the variables  $X_{[d]}$  on the axis of a tensor  $E$ . Given assignments  $x_{[k]}$  to the first  $k$  variables  $X_{[k]}$ , we could choose an assignment on  $X_k$  based on

$$\operatorname{argmax}_{x_k \in [m_k]} \langle E[X_{[d]}] \rangle [X_{[k]} = x_{[k]}, X_k = x_k] .$$

For efficiency reasons, MCTS optimizes an approximative problem instead, where the above contraction is estimated by a particle-based approximation of the tensor.

Alternatives here are further sampling approaches considered in `tnreason`, such as Gibbs sampling strategies resampling previous parameters in combination with simulated annealing.

When annealing the underlying distribution by scaling the energy tensor, the search objective in each decision step is more sensitive to the maximum in the respective slice, but harder to estimate (since the maximum is typically not found during estimation).

## 4 Literature

Small LLMs suffice to develop reasoning capability:

- [Guan et al. \(2025\)](#)

Reinforcement learning to enhance the reasoning capability of LLMs:

- [DeepSeek-AI et al. \(2025\)](#)

Combination with the LEAN language:

- Xin et al. (2024)

Monte Carlo Tree Search:

- Besta et al. (2025): Integration of MCTS into Reasoning Language Models (RLM)
- Xin et al. (2024): RMaxTS algorithm

## References

- Maciej Besta, Julia Barth, Eric Schreiber, Ales Kubicek, Afonso Catarino, Robert Gerstenberger, Piotr Nyczyk, Patrick Iff, Yueling Li, Sam Houlston, Tomasz Sternal, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Łukasz Flis, Hannes Eberhard, Hubert Niewiadomski, and Torsten Hoefler. Reasoning Language Models: A Blueprint, January 2025. URL <http://arxiv.org/abs/2501.11223>. arXiv:2501.11223 [cs].
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaoqun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025. URL <http://arxiv.org/abs/2501.12948>. arXiv:2501.12948 [cs].
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking, January 2025. URL <http://arxiv.org/abs/2501.04519>. arXiv:2501.04519 [cs].
- Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanbiao Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search, August 2024. URL <http://arxiv.org/abs/2408.08152>. arXiv:2408.08152 [cs].