Making the Implicit Explicit

Extended Abstract (Work in Progress)

Jan Hula[1], Petr Hyner[2], and Miroslav Olšák[1]

[1] Czech Technical University in Prague
[2] University of Ostrava

1 Motivation

Humans can step back from a problem and examine how they are thinking about it [9, 15]. Apart from solving problems, we can also design procedures for solving them, reason about why those procedures work, and build confidence in their correctness. This reflective capacity which enables automation underlies much of mathematics and computer science, where progress often comes from turning informal problem-solving skills into explicit algorithms.

Recent advances in AI have shown that automated problem solving can also be achieved by processes that are largely opaque. When a process is not interpretable, it cannot be inspected, and systematically improved, even if the final answer can be checked. In human terms, producing good outcomes without access to the generative mechanism resembles what we often call creativity.

A recurring theme in math and CS is the "automation of creativity": taking what looks like inspired trial-and-error and converting it into a systematic procedure. Our goal is to bring this paradigm to large language models. Rather than relying on an LLM's reasoning to produce an answer, we want the model to externalize its problem solving into an explicit mechanistic procedure, which is much more efficient to run than the LLM itself.

2 The Proposed Approach

We propose a four-stage pipeline that extracts explicit, executable procedures from opaque LLM problem solving traces. In the *discovery* stage, the model is allowed to solve a target class of problems through its usual reasoning, producing solution traces that are otherwise discarded. These traces are then subjected to *extraction*, in which recurring sub-procedures are isolated as candidate algorithmic components [8, 3]. *Formalization* renders the candidates as code, and *testing* evaluates them on held-out instances to confirm that the explicit procedure reproduces, and where possible surpasses, the implicit behavior it was distilled from. Iterating this loop is intended to yield a growing library of reusable building blocks rather than ad hoc one-off scripts.

The pipeline is reflective in the sense that the LLM both generates the traces and participates in their analysis [17, 23, 1]. This dual role lets the model surface intermediate decisions that would otherwise remain internal, turning a successful but opaque problem-solving episode into an object of study. Although our initial experiments target mathematical reasoning, the methodology is not specific to mathematics; it applies to any domain in which the gap between informal expertise and a formal procedure is the dominant source of difficulty. A further benefit of producing explicit artifacts is that they can be composed with conventional symbolic tools, including proof assistants and decision procedures, providing a concrete interface between learned and verified components [22, 25, 5].

Once a reliable explicit procedure has been obtained, the question becomes how to integrate it back into the model so that the procedure is invoked when appropriate. We treat this as a problem

of sparse adaptation: the model's general reasoning ability should be preserved, while specific sub-tasks are delegated to external components. We consider two complementary mechanisms. In-context learning exposes the procedure through demonstrations or tool descriptions, leaving the weights untouched [4, 7]. Sparse fine-tuning tries to adjust the weights of the model in such a way that it rewires it only at the segments where the sub-procedure occured while leaving the rest of the behaviour intact [21, 20, 24]. We study and compare several ways of fine-tuning including fine-tuning external memory modules which can gate the model at specific points [12, 16, 2] and low-rank finetuning methods which update only a small subspace of the model's parameters [13, 6, 11].

Taken together, the extraction pipeline and the integration mechanism describe a hybrid architecture in which the LLM contributes flexible, open-ended reasoning, while a library of explicit procedures, derived from its own traces, handles the steps where correctness and efficiency are most consequential.

The experimental results evaluated on AIME problems [18, 19, 10] and the OpenR1-Math-220k dataset [14] will be presented at the workshop.

References

[1] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[2] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.

[3] Matthew Bowers, Theo X. Olausson, Catherine Wong, et al. Top-down synthesis for library learning. *Proceedings of the ACM on Programming Languages*, 7(POPL):1182–1213, 2023.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Artur d'Avila Garcez and Luís C. Lamb. Neurosymbolic AI: The 3rd wave. *arXiv preprint arXiv:2012.05876*, 2020.

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[7] Qingxiu Dong, Lei Li, Damai Dai, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[8] Kevin Ellis, Catherine Wong, Maxwell Nye, et al. DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, pages 835–850, 2021.

[9] John H. Flavell. Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10):906–911, 1979.

[10] Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[11] Zeyu Han, Chao Gao, Jinyang Liu, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[12] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[14] Hugging Face Open-R1 Team. OpenR1-Math-220k: A large-scale dataset for mathematical reasoning. https://huggingface.co/datasets/open-r1/OpenR1-Math-220k, 2025.

[15] Daniel Kahneman. *Thinking, Fast and Slow.* Farrar, Straus and Giroux, New York, 2011.

[16] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[17] Aman Madaan, Niket Tandon, Prakhar Gupta, et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[18] Mathematical Association of America. American Invitational Mathematics Examination (AIME) 2024. https://huggingface.co/datasets/HuggingFaceH4/aime_2024, 2024.

[19] Mathematical Association of America. American Invitational Mathematics Examination (AIME) 2025. https://huggingface.co/datasets/MathArena/aime_2025, 2025.

[20] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in pre-trained language models. In *International Conference on Machine Learning (ICML)*, 2023.

[22] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, et al. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[23] Noah Shinn, Federico Cassano, Edward Berman, et al. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[24] Yi-Lin Sung, Varun Nair, and Colin Raffel. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[25] Shunyu Yao, Jeffrey Zhao, Dian Yu, et al. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.