

DeepMind's FunSearch uses an LLM to find new cap set constructions

Source copy kept by wheresthe.ai for the story "DeepMind's FunSearch uses an LLM to find new cap set constructions" (<https://wheresthe.ai/d/gb-deepmind-funsearch-cap-set-2023-12/>).

Source: <https://deepmind.google/blog/funsearch-making-new-discoveries-in-mathematical-sciences-using-large-language-models/>

Text saved 18 Oct 2026 (the page as published).

SHA-256 of the saved text: 4c4210889a471dbc8fe84eb0e11feb4d81efe58b7dcdea074abe235630ed07b6

December 14, 2023 Science FunSearch: Making new discoveries in mathematical sciences using Large Language Models Alhussein Fawzi and Bernardino Romera Paredes

Share

By searching for “functions” written in computer code, FunSearch made the first discoveries in open problems in mathematical sciences using LLMs Update: In December 2024, we published a report on arXiv showing how our method can be used to amplify human performance in combinatorial competitive programming. Large Language Models (LLMs) are useful assistants - they excel at combining concepts and can read, write and code to help people solve problems. But could they discover entirely new knowledge? As LLMs have been shown to “hallucinate” factually incorrect information, using them to make verifiably correct discoveries is a challenge. But what if we could harness the creativity of LLMs by identifying and building upon only their very best ideas? Today, in a paper published in Nature , we introduce FunSearch, a method to search for new solutions in mathematics and computer science. FunSearch works by pairing a pre-trained LLM, whose goal is to provide creative solutions in the form of computer code, with an automated “evaluator”, which guards against hallucinations and incorrect ideas. By iterating back-and-forth between these two components, initial solutions “evolve” into new knowledge. The system searches for “functions” written in computer code; hence the name FunSearch. This work represents the first time a new discovery has been made for challenging open problems in science or mathematics using LLMs. FunSearch discovered new solutions for the cap set problem, a longstanding open problem in mathematics. In addition, to demonstrate the practical usefulness of FunSearch, we used it to discover more effective algorithms for the “bin-packing” problem, which has ubiquitous applications such as making data centers more efficient. Scientific progress has always relied on the ability to share new understanding. What makes FunSearch a particularly powerful scientific tool is that it outputs programs that reveal how its solutions are constructed, rather than just what the solutions are. We hope this can inspire further insights in the scientists who use FunSearch, driving a virtuous cycle of improvement and discovery. Driving discovery through evolution with language models FunSearch uses an evolutionary method powered by LLMs, which promotes and develops the highest scoring ideas. These ideas are expressed as computer programs, so that they can be run and evaluated automatically. First, the user writes a description of the problem in the form of code. This description comprises a procedure to evaluate programs, and a seed program used to initialize a pool of programs. FunSearch is an iterative procedure; at each iteration, the system selects some programs from the current pool of programs, which are fed to an LLM. The LLM creatively builds upon these, and generates new programs, which are automatically evaluated. The best ones are added back to the pool of existing programs, creating a self-improving loop. FunSearch uses Google’s PaLM 2 , but it is compatible with other LLMs trained on code.

The FunSearch process. The LLM is shown a selection of the best programs it has generated so far (retrieved from the programs database), and asked to generate an even better one. The programs proposed by the LLM are automatically executed, and evaluated. The best programs are added to the database, for selection in subsequent cycles. The user can at any point retrieve the highest-scoring programs discovered so far.

Discovering new mathematical knowledge and algorithms in different domains is a notoriously difficult task, and largely beyond the power of the most advanced AI systems. To tackle such challenging problems with FunSearch, we introduced multiple key components. Instead of starting from scratch, we start the evolutionary process with common knowledge about the problem, and let FunSearch focus on finding the most critical ideas to achieve new discoveries. In addition, our evolutionary process uses a strategy to improve the diversity of ideas in order to avoid stagnation. Finally, we run the evolutionary process in parallel to improve the system efficiency. Breaking new ground in mathematics We first address the cap set problem , an open challenge, which has vexed

mathematicians in multiple research areas for decades. Renowned mathematician Terence Tao once described it as his favorite open question . We collaborated with Jordan Ellenberg, a professor of mathematics at the University of Wisconsin–Madison, and author of an important breakthrough on the cap set problem . The problem consists of finding the largest set of points (called a cap set) in a high-dimensional grid, where no three points lie on a line. This problem is important because it serves as a model for other problems in extremal combinatorics - the study of how large or small a collection of numbers, graphs or other objects could be. Brute-force computing approaches to this problem don't work – the number of possibilities to consider quickly becomes greater than the number of atoms in the universe. FunSearch generated solutions - in the form of programs - that in some settings discovered the largest cap sets ever found. This represents the largest increase in the size of cap sets in the past 20 years. Moreover, FunSearch outperformed state-of-the-art computational solvers, as this problem scales well beyond their current capabilities.

These results demonstrate that the FunSearch technique can take us beyond established results on hard combinatorial problems, where intuition can be difficult to build. We expect this approach to play a role in new discoveries for similar theoretical problems in combinatorics, and in the future it may open up new possibilities in fields such as communication theory. FunSearch favors concise and human-interpretable programs While discovering new mathematical knowledge is significant in itself, the FunSearch approach offers an additional benefit over traditional computer search techniques. That's because FunSearch isn't a black box that merely generates solutions to problems. Instead, it generates programs that describe how those solutions were arrived at. This show-your-working approach is how scientists generally operate, with new discoveries or phenomena explained through the process used to produce them. FunSearch favors finding solutions represented by highly compact programs - solutions with a low Kolmogorov complexity†. Short programs can describe very large objects, allowing FunSearch to scale to large needle-in-a-haystack problems. Moreover, this makes FunSearch's program outputs easier for researchers to comprehend. Ellenberg said: "FunSearch offers a completely new mechanism for developing strategies of attack. The solutions generated by FunSearch are far conceptually richer than a mere list of numbers. When I study them, I learn something". What's more, this interpretability of FunSearch's programs can provide actionable insights to researchers. As we used FunSearch we noticed, for example, intriguing symmetries in the code of some of its high-scoring outputs. This gave us a new insight into the problem, and we used this insight to refine the problem introduced to FunSearch, resulting in even better solutions. We see this as an exemplar for a collaborative procedure between humans and FunSearch across many problems in mathematics.

Left: Inspecting code generated by FunSearch yielded further actionable insights (highlights added by us). Right: The raw "admissible" set constructed using the (much shorter) program on the left.

The solutions generated by FunSearch are far conceptually richer than a mere list of numbers. When I study them, I learn something.

Jordan Ellenberg collaborator and professor of mathematics at the University of Wisconsin–Madison

Addressing a notoriously hard challenge in computing Encouraged by our success with the theoretical cap set problem, we decided to explore the flexibility of FunSearch by applying it to an important practical challenge in computer science. The "bin packing" problem looks at how to pack items of different sizes into the smallest number of bins. It sits at the core of many real-world problems, from loading containers with items to allocating compute jobs in data centers to minimize costs. The online bin-packing problem is typically addressed using algorithmic rules-of-thumb (heuristics) based on human experience. But finding a set of rules for each specific situation - with differing sizes, timing, or capacity – can be challenging. Despite being very different from the cap set problem, setting up FunSearch for this problem was easy. FunSearch delivered an automatically tailored program (adapting to the specifics of the data) that outperformed established heuristics – using fewer bins to pack the same number of items.

Your browser does not support the video tag. Your browser does not support the video tag.

Illustrative example of bin packing using existing heuristic – Best-fit heuristic (left), and using a heuristic discovered by FunSearch (right).

Hard combinatorial problems like online bin packing can be tackled using other AI approaches, such as neural networks and reinforcement learning. Such approaches have proven to be effective too, but may also require significant resources to deploy. FunSearch, on the other hand, outputs code that can be easily inspected and deployed, meaning its solutions could potentially be slotted into a variety of real-world industrial systems to bring

swift benefits. Update: Enhancing human performance in combinatorial competitive programming In December 2024, we published a report by Veličković et al on arXiv showing how our method can be used to amplify human performance in combinatorial competitive programming. In traditional coding contests like Codeforces which was targeted by AlphaCode , competitors need to provide complete solutions to classical algorithmic challenges in a time- and memory-constrained setting. In comparison, combinatorial contests feature highly complex problems where the objective is not to find the right answer but the best possible approximate solution, similar to problems like finding cap sets. Given the hardness of these problems for humans, our method can produce solutions that outperform ones that were found by the top percentile of competitors. And it uses an approach that lends itself well to human-AI collaboration: human programmers write the ‘backbone’ of the solution code and then allow an LLM to creatively evolve the function that steers it.

This is an exciting approach to combine work of human competitive programmers and LLMs, to achieve results that neither would achieve on their own.

Petr Mitrichev Software Engineer, Google, World-class Competitive Programmer

With improved generalist LLMs, we no longer require code-specialised models and can build on Gemini 1.5 Flash . Beyond competitive programming, we used FunSearch to find better ways to optimize functions within the framework of Bayesian optimization. LLM-driven discovery for science and beyond FunSearch demonstrates that if we safeguard against LLMs’ hallucinations, the power of these models can be harnessed not only to produce new mathematical discoveries, but also to reveal potentially impactful solutions to important real-world problems. We envision that for many problems in science and industry - longstanding or new - generating effective and tailored algorithms using LLM-driven approaches will become common practice. Indeed, this is just the beginning. FunSearch will improve as a natural consequence of the wider progress of LLMs, and we will also be working to broaden its capabilities to address a variety of society’s pressing scientific and engineering challenges.

Learn more about FunSearch

Read the open access copy of the paper* Read our paper in Nature Amplifying human performance in combinatorial competitive programming FunBO: Discovering Acquisition Functions for Bayesian Optimization with FunSearch

Acknowledgements Petar Veličković, Alex Vitvitskyi, Larisa Markeeva, Borja Ibarz and Alexander Novikov contributed to the December 2024 update on ‘Enhancing human performance in combinatorial competitive programming’. Matej Balog, Emilien Dupont, Alexander Novikov, Pushmeet Kohli, Jordan Ellenberg for valuable feedback on the blog and for help with the figures. This work was done by a team with contributions from: Bernardino Romera Paredes, Amin Barekatain, Alexander Novikov, Matej Balog, Pawan Mudigonda, Emilien Dupont, Francisco Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, George Holland, Pushmeet Kohli and Alhussein Fawzi. *This is the author’s version of the work. It is posted here by permission of Nature for personal use, not for redistribution. The definitive version was published in Nature: DOI: 10.1038/s41586-023-06924-6 . †Kolmogorov complexity is the length of the shortest computer program outputting the solution.

Related posts

Discovering novel algorithms with AlphaTensor October 2022 Science Learn more