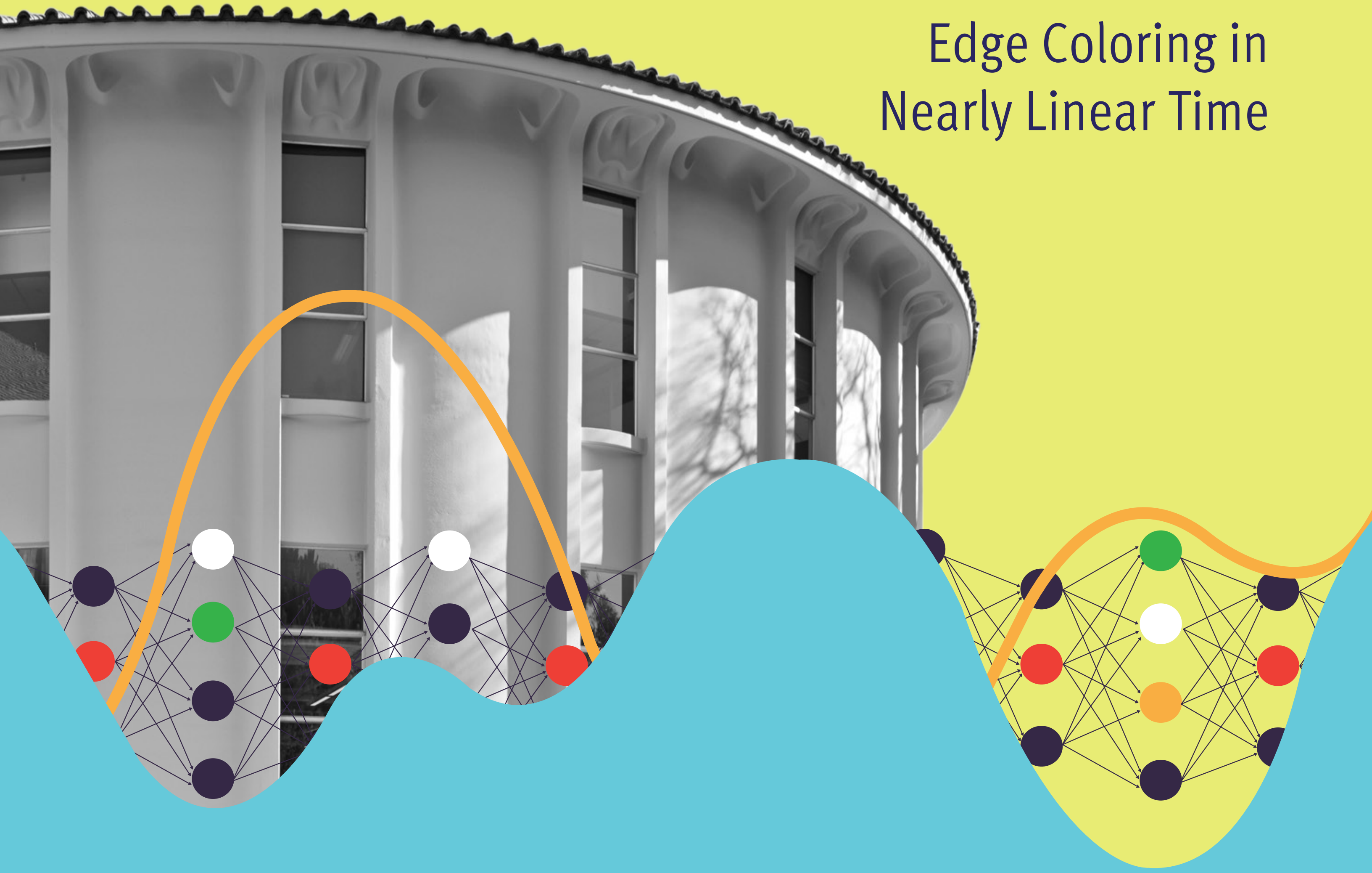
Polynomial Times

2025–26 | Issue 1

Watermarks and Pseudorandom Codes

Edge Coloring in Nearly Linear Time



The Compressed Oracle Method and Its Generalization

Optimal List Decoding

Simons Institute
for the Theory of Computing

Polynomial Times

the annual magazine of
the Simons Institute for the Theory of Computing

2025–26 | Issue 1

Polynomial Times contributors and editorial team

Venkatesan Guruswami, director
Sampath Kannan, associate director
Nikhil Srivastava, senior scientist
Anil Ananthaswamy,
science communicator at large
Kristin Kane,
editor & director of communications and finance
Barry Bödeker, graphic designer
Preeti Aroon, copy editor

Senior advisors

Peter Bartlett, research director, machine learning
Shafi Goldwasser, research director, resilience
Umesh Vazirani, research director, quantum
Jelani Nelson, senior advisor
Bin Yu, senior advisor

Operations team

Amy Ambrose,
senior development director & director of operations
Ashley Hasson, event coordinator
Robin Hoey, facilities manager
Frances Klingenberger, event coordinator
Kelsey Larsen, financial specialist
Drew Mason, information systems analyst
Joe Ratzlaff, information systems analyst
Atiya Wahab, program coordinator
Carolyn Winter, scientific program coordinator
Carson Fox Young, executive assistant

Scientific Advisory Board (2025)

Nina Balcan
Bonnie Berger
Shuchi Chawla
Paul Christiano
Uriel Feige
Parikshit Gopalan
(co-chair)
Martin Grohe
Monika Henzinger
Maurice Herlihy
Jonathan Kelner
Iordanis Kerenidis
Orna Kupferman
Moni Naor
Ryan O'Donnell
Thomas Rothvoss
Rocco Servedio
Nati Srebro
Salil Vadhan
(co-chair)
Mary Wootters

Simons Institute By the Numbers (2012–2025)

4,000+
long-term visitors

419
research fellows

68
research programs and clusters

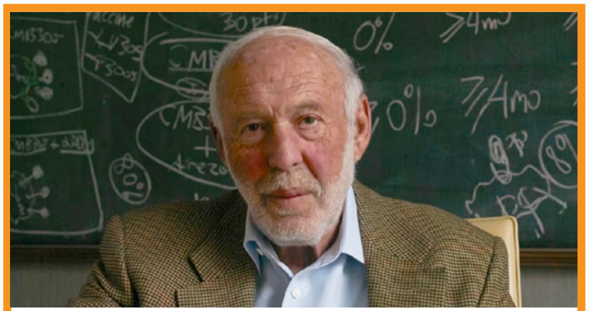
131
average research papers per program

4.4
average collaborators per visitor

37
countries of visitors' home institutions

6,000+
videos on YouTube

Table of Contents



This inaugural issue of *Polynomial Times* is dedicated to Jim Simons (1938–2024), whose visionary philanthropy has made our work possible.

November 24, 2025

Dear friends,

I am honored and humbled to write to you in my new role as the third director of the Simons Institute. I know I am stepping into some giant shoes, and I will do my best to ensure that the Institute continues to thrive and serve as the hugely influential and beloved global hub for the theory of computing. As one of my first communications as director, I'm delighted to share with you the inaugural issue of *Polynomial Times*, the annual magazine of the Simons Institute for the Theory of Computing. Released each year in the fall, the magazine will showcase some key results and connections emerging from our recent programs, review new initiatives and special convenings, and explore what's on the horizon for the Institute in the near future.

In 2024–25, we held standard research programs on Sublinear Algorithms (Summer 2024) and Modern Paradigms in Generalization (Fall 2024); a summer cluster on AI, Psychology, and Neuroscience (Summer 2024); an extended reunion for the program on Theoretical Foundations of Computer Systems (Summer 2024); and a Special Year on Large Language Models and Transformers (Fall 2024 and Spring 2025). We also held a number of special workshops not associated with programs, including a workshop on Theoretical Aspects of Trustworthy AI, and a joint workshop with SLMath on AI for Mathematics and Theoretical Computer Science. We hosted ongoing research pods on Machine Learning, Quantum Computing, and Resilience in Brain, Natural, and Algorithmic Systems, and presented two public lecture series, *Theoretically Speaking* and the *Richard M. Karp Distinguished Lectures*. We opened our doors to 268 long-term participants in our research programs and clusters this past year.

We continue to train the largest cohort of postdoctoral-level researchers in theoretical computer science worldwide, comprising multiyear postdoctoral positions in our research pods and semester-long research fellowships within each research program. Many of this past year's research fellows have gone on to tenure-track positions at prestigious institutions, including Cornell, Johns Hopkins, Yale, UMass, UW–Madison, UC Berkeley, UT Austin, NYU, Princeton, UC San Diego, Tel Aviv University, and Monash University.

We announced calls for two named workshop series this past year: *Breakthroughs Workshops* and *Goldwasser Exploratory Workshops*, the first of which will be held in 2025–26. From time to time, the steady progress of research is interrupted by a massive leap forward, due to a particular breakthrough result. When such breakthroughs happen, they enable a cascade of progress as researchers examine their implications for a wide range of problems and applications. The Simons Institute's *Breakthroughs Workshops* celebrate breakthrough results and provide a forum for the integration and extrapolation to follow. Meanwhile, the *Goldwasser Exploratory Workshops* honor Simons Institute Director Emerita Shafi Goldwasser, whose ventures into uncharted territory have led to field-transforming discoveries, including zero-knowledge proofs, for which she and Silvio Micali received the Turing Award. In this spirit, each *Goldwasser Exploratory Workshop* will stake out new territory, explore new interdisciplinary alliances, or advance unexpected approaches to long-standing problems.

We launched another initiative during 2024–25: Circles, the Simons Institute – Jane Street Small Group Collaborations. Supported by a gift from Jane Street, this initiative supports groups of three to six researchers for four weeklong visits (two visits to Jane Street in New York and two to the Simons Institute) spread over two years, to collaborate intensively on an ambitious research project. The inaugural accepted projects — one on Building Bridges: Codes, TCS, and Geometric Group Theory, and another on Approaches to the Metamathematical Difficulty of Complexity Lower Bounds

— will bring together collaborative groups where some members have worked together at the Simons Institute during our past research programs.

In May 2025, we upgraded the audiovisual system in our main auditorium. Under the new setup, presenters can now independently initiate Zoom webinars, without staff assistance. New audience-facing cameras enrich the experience for remote presenters and our worldwide online audience. And the new equipment offers improved audio quality in our livestreams and video recordings.

In the current (2025–26) academic year, we've already run a summer research program, Cryptography 10 Years Later: Obfuscation, Proof Systems, and Secure Computation; as well as a Summer Cluster on Quantum Computing. Also on the docket this year: Algorithmic Foundations for Emerging Computing Technologies (Fall 2025), Complexity and Linear Algebra (Fall 2025), and Federated and Collaborative Learning (Spring 2026). We are excited to have been selected as an inaugural member of the Google DeepMind x Google.org AI for Math Initiative, and look forward to engaging our research community in building out our participation in the consortium.

The Simons Institute is fundamentally community driven, with its programmatic agenda shaped by our brilliant and collaborative research community. All of us in the Institute's leadership are tremendously grateful to all the researchers, funders, and broader community who give the Institute its vitality. As director, I look forward to collaborating with all of you to sustain and deepen the Institute's hallmark atmosphere of immersion and intensity, which time and again has fueled stunning advances only possible through a sustained cross-fertilization of ideas among researchers with complementary expertise.

We've dedicated this inaugural issue to our founding benefactor, Jim Simons, who passed away in 2024. Jim and Marilyn Simons' broad vision and commitment to basic science inspired them to make philanthropic investments that have transformed the face of our field. We are deeply grateful to them and to the Simons Foundation, for without their support, the work described in these pages might never have been done.

I hope you enjoy the first issue of *Polynomial Times*, including the research vignettes we share with you here. I look forward to seeing you in Berkeley soon.

Yours,

Venkatesan Guruswami
Director

Watermarks and Pseudorandom Codes

Anil Ananthaswamy, science communicator at large

In May 2023, an account on Twitter posted an image of an explosion near the Pentagon. The image was shared widely on social media. Stock markets dipped briefly but recovered when authorities — including the Arlington County Fire Department in Virginia — confirmed that there had been no such explosion. Hany Farid, a professor of computer science at UC Berkeley and an expert in digital forensics, misinformation, and image analysis, told the media that many features in the image were inconsistent with real images of the Pentagon. This suggested that the image had been generated using an AI model.[1]

Since then, the use of artificial intelligence for generating text, images, and even video has become so much more sophisticated, making it easier to fool more people for longer. Simply using human cognition to flag AI-generated content is futile. Methods to watermark such content, by embedding hidden patterns and even messages (such as time stamps and user IDs) into the generated data, are becoming an imperative.

But when Sam Gunn, a computer science PhD student in the Theory Group at UC Berkeley, and Miranda Christ, a PhD student and member of the Theory Group and the Crypto Lab at Columbia University in New York, began looking at existing watermarking methods, they found them lacking. So, during time at the Simons Institute for the Spring 2023 program on Meta-Complexity, Gunn and Christ studied the use of pseudorandomness — the bedrock of cryptography — to construct pseudorandom codes for watermarking.

Sam Gunn

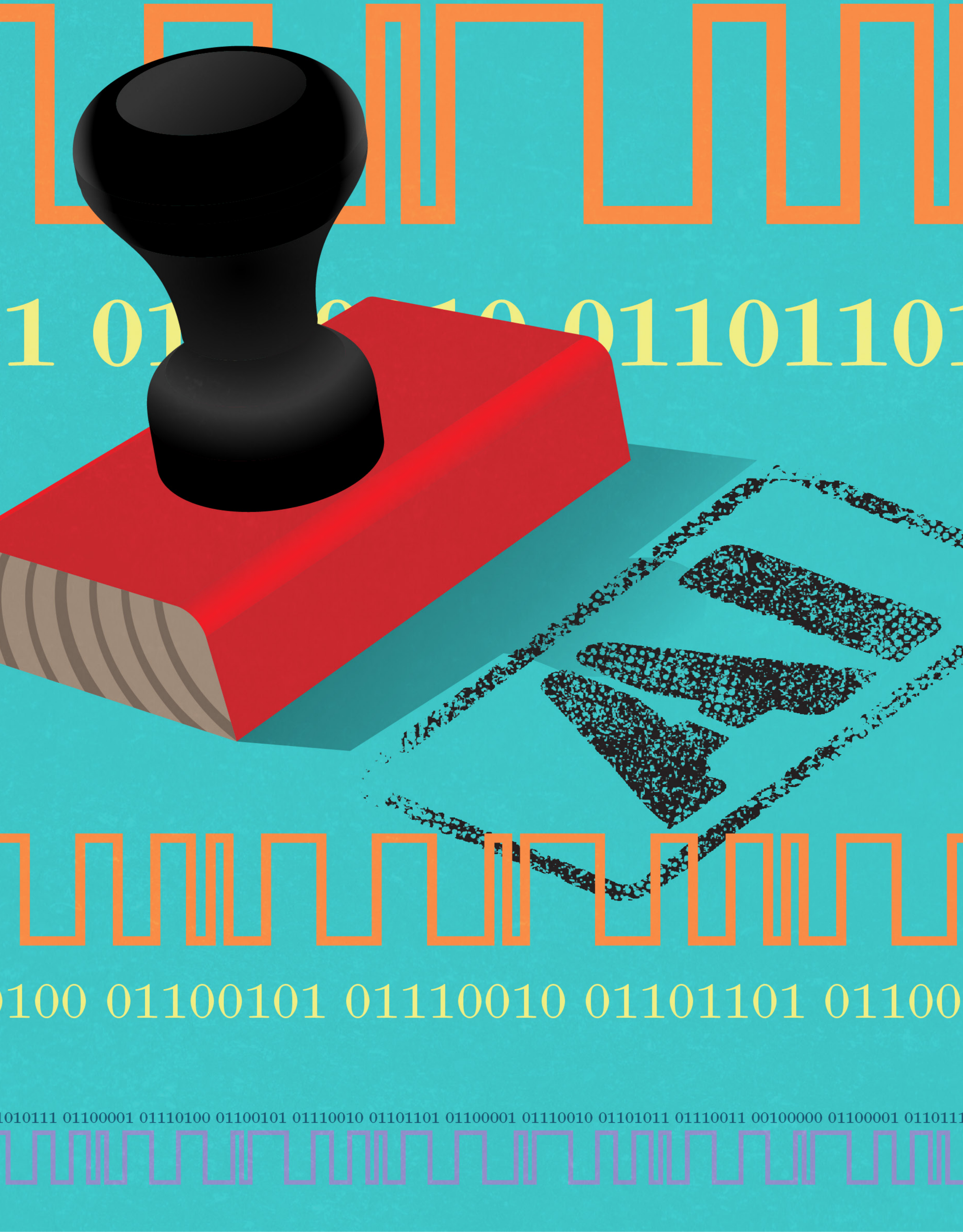
Any watermark should satisfy three requirements:

1) Quality: the watermark shouldn't degrade the generated content; watermarked content should look, sound, or read no differently from the unwatermarked counterpart.

2) Robustness: schemes for detecting watermarks in generated content should have a high true-positive rate (correctly flagging content as AI generated when it is), ideally even after malicious perturbations to the watermarked content.

3) Unforgeability: detectors should have a low false-positive rate (erroneously flagging content as the output of an AI model when it isn't), even after malicious perturbations to unwatermarked content.

[1] https://www.ischool.berkeley.edu/news/2023/hany-farid-breaks-down-fake-pentagon-images-cnn-article


1 01 10 0110110
100 01100101 01110010 01101101 0110
1010111 01100001 01110100 01100101 01110010 01101101 01100001 01110010 01101011 01110011 00100000 01100001 0110111
AI

> “They begin by showing how to build a pseudorandom error-correcting code, or simply a pseudorandom code, which they parametrize with a decoding key to generate codewords.”

Christ and Gunn realized that existing watermarking methods involved untenable trade-offs among these properties, and they proved that pseudorandom codes (PRCs) are necessary to achieve all three properties simultaneously. The researchers constructed PRCs by combining two foundational concepts: pseudorandomness (from cryptography) and error-correcting codes (from theoretical computer science). Pseudorandomness enables, for example, the generation of bit strings that are n bits long, using a deterministic function that uses as its input bit strings that are k bits long, where $k << n$, such that the generated bit strings look uniformly random to any polynomial-time adversary (i.e., an adversary using an algorithm whose running time is polynomial in n). Error-correcting codes involve adding extra information to the generated bit strings such that even if some fraction of the bits were to be corrupted, one can reconstruct the original bit strings. “PRCs are nontrivial to construct, despite the simplicity of both pseudorandomness and error-correcting codes,” said Gunn.

But once they built a PRC, using it for watermarking came down to identifying a source of randomness in the generative AI algorithm and replacing some of that randomness with outputs from a PRC.

Any generative AI model implicitly or explicitly does two things: it first learns a probability distribution over the training data (such as text, images, or videos), and then it samples from that distribution to produce data whose statistics resemble the statistics of the training data. The sampling introduces an element of randomness. Generative AI models differ in the exact specifics of how they accomplish these tasks.

For example, a diffusion model for image generation is trained to sample from a unit normal Gaussian (pure noise) and then denoise the sample (the so-called reverse diffusion process) to produce an image that looks like a sample from the distribution over the images in the training data. Sampling from the unit normal involves randomness.

Or take a large language model (LLM). An LLM, given some prompt, produces a probability distribution over its entire vocabulary of words (or, more precisely, tokens). The algorithm for generating text then samples from this distribution to predict the next word or token. Again, randomness comes in at this stage of data generation.

Miranda Christ

In their paper, “Pseudorandom Error-Correcting Codes,” the first version of which was published on arXiv in February 2024, Christ and Gunn presented a watermarking scheme for language models.[2] They begin by showing how to build a pseudorandom error-correcting code, or simply a pseudorandom code, which they parametrize with a decoding key to generate codewords. Without this key, any polynomial number of codewords would appear pseudorandom to an adversary.

[2] https://arxiv.org/abs/2402.09370

The PRC can also correct for errors (the number of which is bounded) and is thus robust. So if a message m is encoded into a message x, and x' is a corrupted version of x, then an algorithm can decode x' to recover the original message m. In this scenario, x corresponds roughly to the watermarked content of a model, and x' to the perturbed content resulting from an adversary trying to remove the watermark. The pseudorandomness of the PRC enables the watermark's high quality, and the robustness of the PRC allows the watermark detector to work despite the malicious intervention.

Such a PRC can be used to watermark the content of a language model. Christ and Gunn define an abstract algorithm called *Generate* that takes as input a prompt and a random seed $x \in \{0, 1\}^n$ and samples a response $t \in \{0, 1\}^n$. (They develop their method for binary tokens — i.e., the language model has an alphabet 0 and 1 — and then show that their results generalize to a language model with an arbitrary token alphabet.) *Generate* works iteratively, or auto-regressively, by first taking the user's prompt, and sampling the next token, appending the token to the prompt, and sampling the next token, and so on, until it generates an end-of-text token. When *Generate* is given a seed from a PRC, the generated text is said to be watermarked.

Previous methods used the same seed for all responses. This resulted in a lack of diversity in a language model's responses. To improve diversity, the scheme with a single seed had to increase the length of the seed, costing the detector more compute time to spot the watermark. Such schemes had to trade off generation diversity against detector efficiency.

Christ and Gunn circumvented this trade-off by sampling a new seed for each response, ensuring that there are no discernible correlations between responses, making the watermark undetectable to anyone without the key. The algorithm preserves the language model's output diversity while simultaneously ensuring that a detector with a key can spot the watermark, regardless of the model's output length and diversity.

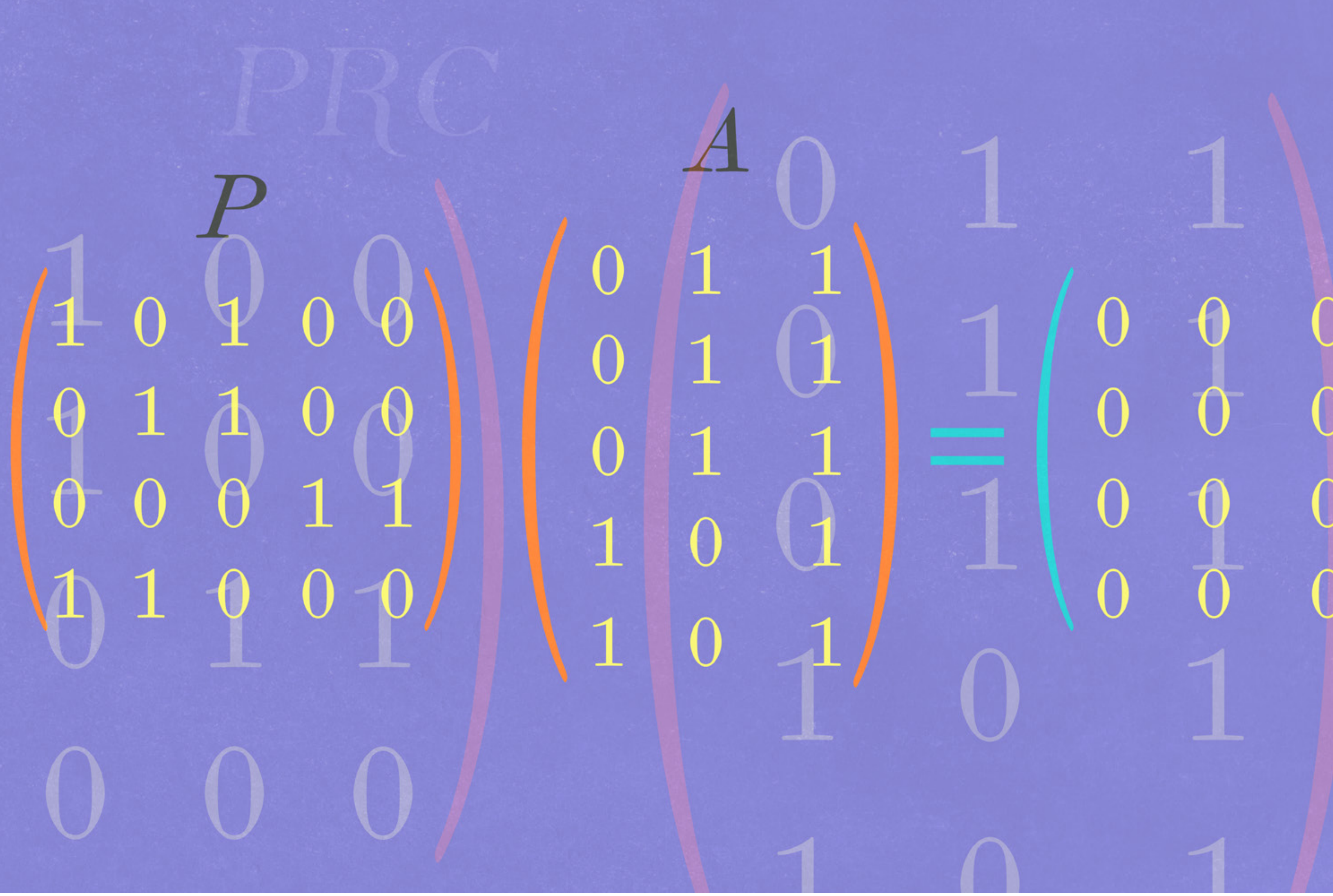
Crucially, the *Generate* function satisfied two important properties. One: given random seeds, the function's output matched the language model's unwatermarked distribution. Two: given structured seeds from a PRC, the function's outputs are detectably correlated with the seeds.

"The watermark is undetectable in the sense that any number of samples of watermarked text are computationally indistinguishable from text output by the original model," wrote Christ and Gunn in their paper. "This is the first undetectable watermarking scheme that can tolerate a constant rate of errors."

> "The pseudorandomness of the PRC enables the watermark's high quality, and the robustness of the PRC allows the watermark detector to work despite the malicious intervention."

Then, in October 2024, Gunn and Xuandong Zhao, a postdoctoral researcher with UC Berkeley professor Dawn Song, used a similar technique to watermark image generation models. In particular, they showed how to watermark images generated by Stable Diffusion 2.1 (it came down to replacing the random samples of Gaussian noise with seeds from their PRC). In their paper,[3] they concluded not only that their scheme allowed them to watermark images and encode long messages in the watermark (which could be extracted by a decoder with the key) without creating any discernible shift in the distribution of generated images, but that it's also robust to adversarial attacks: adversaries cannot remove the watermark without significantly altering the quality of the generated images.

[3] https://arxiv.org/abs/2410.07369

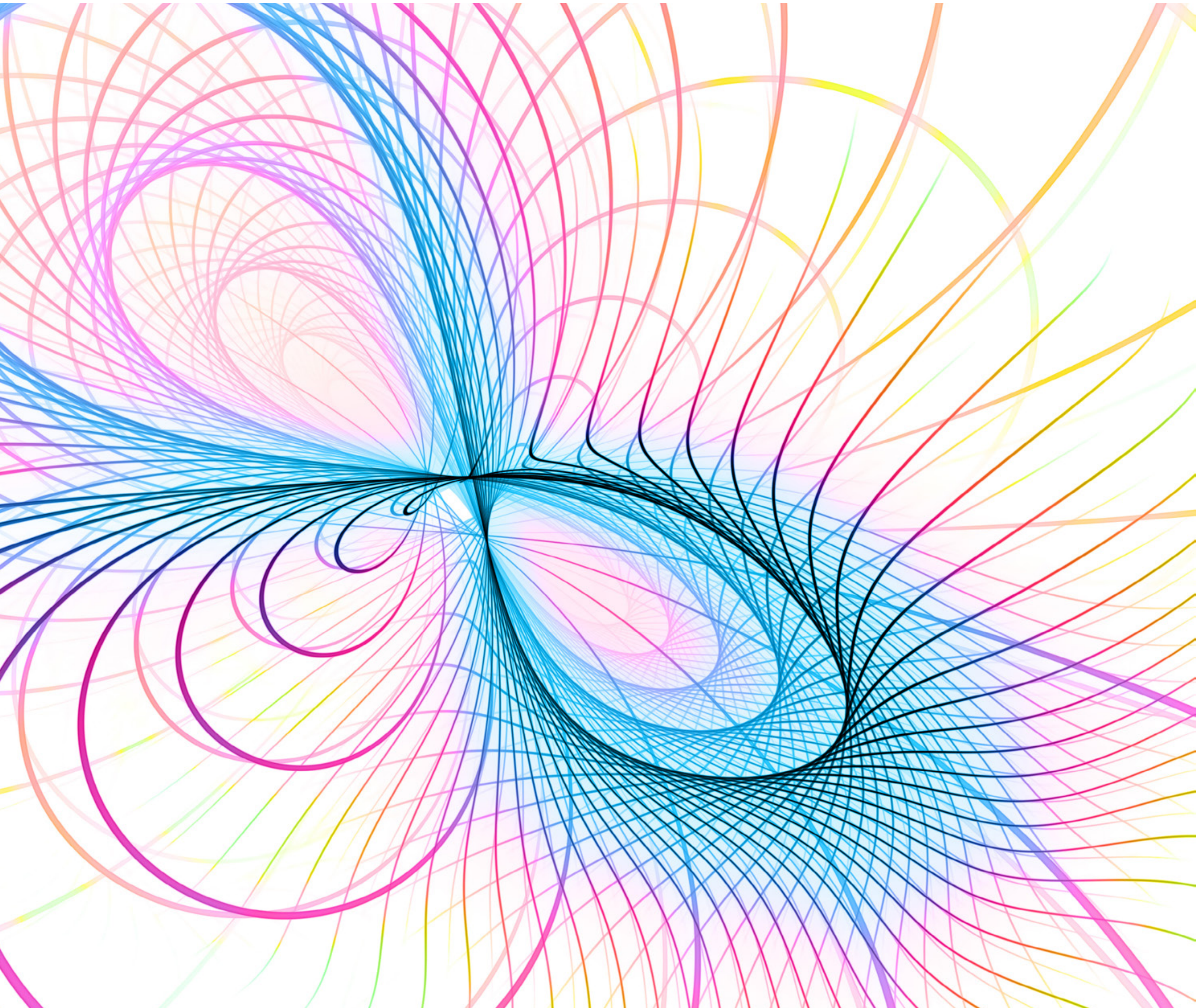


"Pseudorandom code is the only way that we know how to do quality-preserving watermarks for image models," said Gunn. But for language models, while Christ and Gunn proved that their watermarking scheme would work asymptotically for LLMs, they have yet to implement it, because the number of generated tokens needed for the watermark to appear and be robust is currently impractical. "We can prove that it will work. We know exactly how to do it," said Gunn. "It's a problem that can be solved with some more work."

To further strengthen the security of their scheme, Christ, Gunn, Omar Alrabiah (an EECS graduate student at UC Berkeley), and colleagues addressed some additional concerns. The watermarking method described above is robust against errors that are introduced obliviously, or by a memory-less channel, meaning the errors are the outcome of a process that has no knowledge of the watermark or the key, and each error is independent of previous errors. Adversaries in real life might have access to the key, however, or to multiple instances of watermarked content, or to a decoding oracle that might be able to detect watermarks. Any watermarking scheme using pseudorandom codes that can withstand such an adversary is termed adaptively robust. In their latest paper, "Ideal Pseudorandom Codes," published on arXiv in November 2024 and presented at STOC 2025, the authors proved that a small tweak to their earlier PRC can make it adaptively robust, even for certain worst-case settings.[4] "These results immediately imply stronger robustness guarantees for generative AI watermarking schemes," the authors write. ●

[4] https://arxiv.org/html/2411.05947

“Yet, the question of full PRUs that could fool *adaptive* adversaries — those whose behavior could depend on the outcomes of previous queries — remained mysterious.”



The Compressed Oracle Method and Its Generalization

Nikhil Srivastava, senior scientist

The compressed oracle method and its generalization, the path-recording oracle, are beautiful linear algebraic techniques that have led to fundamental discoveries in quantum cryptography and complexity over the past year and a half, in the Simons Institute's Quantum Pod and its thematic research programs.

The quest for pseudorandom unitaries

To set the stage, recall that a (cryptographic) pseudorandom permutation (PRP) is a polynomial-time computable random permutation on $\{0,1\}^n$ that cannot be distinguished from a uniformly random permutation by any $poly(n)$ time algorithm given black-box query access to it. PRPs are known to exist under standard cryptographic assumptions and are a foundational object in classical cryptography.

The quantum analogue of PRPs is pseudorandom unitaries (PRUs), first defined by Ji, Liu, and Song in 2017.[1] A pseudorandom unitary is an efficiently computable $2^n \times 2^n$ unitary matrix that is indistinguishable from a uniformly (Haar) random unitary by any $poly(n)$ time quantum algorithm, which we will refer to as an adversary. Besides being a natural object in quantum cryptography, PRUs were of interest to physicists, who use them to model black holes, among other things.

The question of the existence of PRUs captivated the quantum community, including at the Simons Institute, where many talks were given about it over the past few years. The first constructions of PRUs that could fool *nonadaptive* adversaries were given in early 2024, in two independent breakthrough works by Metger-Poremba-Sinha-Yuen[2] and by Chen-Bouland-Brandão-Docter-Hayden-Xu.[3] Nonadaptive adversaries are mathematically nice because the output of a $t = poly(n)$ time distinguishing algorithm Alg given black-box access to a unitary U, which we will denote $|Alg^U\rangle$, is linear in the tensor power $U^{\otimes t}$, and the problem boils down to showing closeness of (the covariance matrices of) these tensor powers of the random and pseudorandom unitaries in an appropriate norm.[4] This is still difficult, but it can be approached using the framework of representation theory and random matrix theory.

Interestingly, both papers found (different) ways to reduce PRUs to PRPs. Metger et al. introduced a particularly simple construction based on representation theory called the "PFC ensemble," which they conjectured could actually fool adaptive adversaries. Chen et al. developed a new approach in the spirit of random matrix theory, which as I pointed out in a Simons Institute newsletter article[5] had a huge impact on random matrix theory itself.

Yet, the question of full PRUs that could fool *adaptive* adversaries — those whose behavior could depend on the outcomes of previous queries — remained mysterious.

[1] https://arxiv.org/pdf/1711.00385
[2] https://arxiv.org/pdf/2404.12647
[3] https://arxiv.org/pdf/2404.16751
[4] https://www.math3ma.com/blog/the-tensor-product-demystified
[5] https://simons.berkeley.edu/news/theory-institute-beyond-october-2024

Hidden symmetries, and a crash course in tensor products

Enter the compressed oracle method, invented in 2018 by Mark Zhandry.[6] In the classical world, you can analyze the interaction between an adversary and a random function $f : \{0,1\}^n \to \{-1,1\}$ given as a black box — called a random oracle — by *lazy evaluation*: basically, you sample the bits of the oracle on the fly depending on the adversary's queries and store past answers, yielding a succinct "stateful" description of the oracle that is useful in proofs. But this idea doesn't work in the quantum world, because the adversary can query the oracle in *superposition* — mathematically, the oracle is a random ± 1 *diagonal* unitary matrix U, and a single query means preparing a quantum state

$$\sum_{x\in\{0,1\}^n} \alpha_x U|x\rangle,$$

which can depend on all the values of $U(x)$.

Zhandry's beautiful insight was that it is nonetheless possible to "quantize" the lazy evaluation argument by exploiting three facts about the way measurement works in quantum mechanics.

Observe that if U is a (discrete, for technical convenience) random variable with probability distribution $p(\cdot)$, then one can encode the behavior of an adversary on the entire distribution of U at once by considering the "purified" state

$$|\psi\rangle := \sum_U \sqrt{p(U)}|Alg^U\rangle|U\rangle,$$

where $|a\rangle|b\rangle$ is the bra-ket notation for the tensor product $a \otimes b$ of two vectors. If that seems unfamiliar, this is a good moment to study the definition of the tensor product of two vector spaces.

Fact 1. The covariance matrix

$$\mathbb{E}_{U\sim p(U)}|Alg^U\rangle\langle Alg^U|$$

(which is the thing we care about in the PRU problem and in quantum query complexity) is equal to the reduced density matrix

$$\rho_{Alg} := Tr_2(|\psi\rangle\langle\psi|)$$

on the first tensor factor, where Tr_2 denotes the partial trace acting on the second tensor factor. This is the quantum analogue of taking a marginal probability distribution.

Fact 2. The reduced density matrix ρ_{Alg} is *invariant* under applying a unitary transformation of type $I \otimes T$ to $|\psi\rangle$, i.e.,

$$\rho_{Alg} = Tr_2(I \otimes T(|\psi\rangle\langle\psi|)I \otimes T^\dagger)$$

for all unitary T. Here $I \otimes T$ denotes the tensor product of two operators.

Fact 3. The tensor product is bilinear, in particular

$$(\alpha a) \otimes b = a \otimes (\alpha b)$$

for a scalar alpha.

Conceptually, what this is saying is that there is a *different description* of the covariance matrix ρ_{Alg} in an enlarged space, which admits *many more symmetries* than the original description, in the form of Fact 2. The punch line is that by choosing T to be the Fourier transform — a certain symmetry of the uniform distribution on diagonal U — one obtains an alternate, succinct combinatorial description of the U oracle. This is the "compressed oracle," which may be viewed as an efficient data structure that exactly simulates a random diagonal U to an efficient adversary. Though the proof is short, it seems magical to me that a mathematical duality (of the Fourier transform) yields a computational duality (of efficiency in the adversary and in the oracle), essentially by using Fact 3 to "push" the behavior of the adversary onto the oracle.

[6] https://eprint.iacr.org/2018/276.pdf

> “Unlike Zhandry, they did not particularly care about the succinctness or efficiency of this oracle viewed as a data structure. What was really important was the symmetries satisfied by the path-recording oracle.”

In October 2024, Fermi Ma and Hsin-Yuan Huang fully solved the problem of the existence of PRUs by showing that the PFC ensemble is in fact secure against adaptive adversaries assuming the existence of one-way functions, as Metger et al. had conjectured. Their striking insight was that the compressed oracle method can be generalized to the case of Haar random U (i.e., not diagonal) if one allows a small error — a generalization they named the “path-recording oracle.” This is surprising since the unitary group is highly noncommutative and does not admit nearly as nice a “Fourier transform.”

Hsin-Yuan Huang

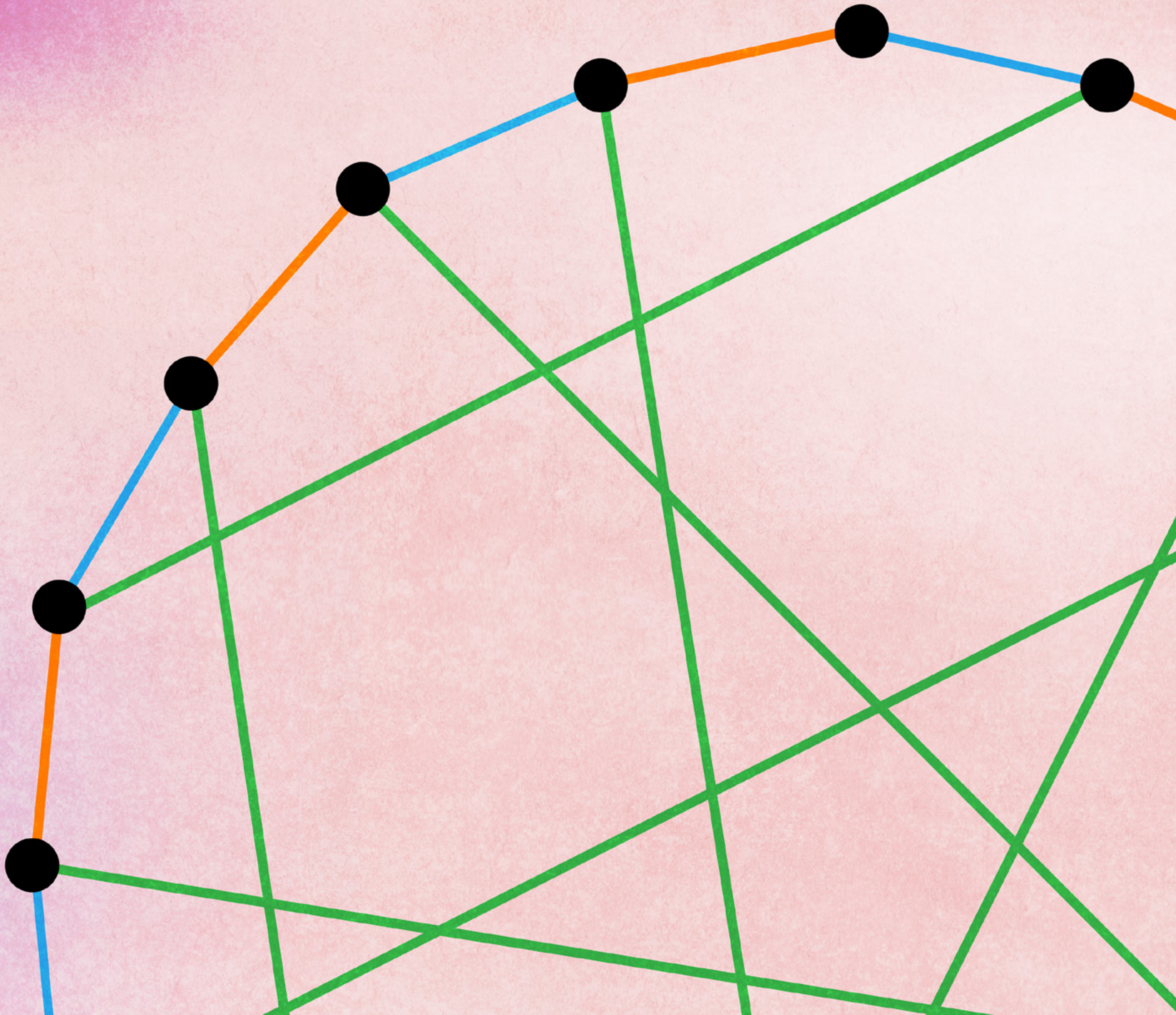
Fermi Ma

Unlike Zhandry, they did not particularly care about the succinctness or efficiency of this oracle viewed as a data structure. What was really important was the symmetries satisfied by the path-recording oracle — in particular, that it *exactly* simulates the uniform distribution on random signed $2^n \times 2^n$permutations, for a *large subclass* of adversaries known as the “distinct subspace,” which also played a role in the work of Metger et al. By exploiting a higher-order analogue of Fact 3, they upgraded this to the following dramatic conclusion:

The path-recording oracle *approximately* simulates *every* “mildly symmetric” random variable U against *all* adversaries.

It turned out that both the PFC ensemble and the Haar unitary ensemble satisfied the required mild symmetries, so Ma and Huang concluded that they must *both* be close to the path-recording oracle and therefore to each other in the appropriate sense, solving the problem. It is hard to imagine a more elegant proof than theirs. It is all identities, save for a single inequality which amounts to showing that a Euclidean projection cannot increase the norm of a vector. And yet, the motivation for this crisp linear algebraic proof came from considerations that are emblematic of theoretical computer science: simulation, interaction, efficiency, and approximation. ●

“At first glance, this problem sounds similar to the notoriously difficult problem of graph coloring, where the goal is to color the *vertices* of a graph using as few colors as possible, so that adjacent vertices get different colors. For this latter problem, even getting a very crude approximation to the number of colors required is known to be NP-hard.”



Edge Coloring in Nearly Linear Time

Sampath Kannan, associate director

Given a graph, how do we color its *edges* using as few colors as possible, so that any two edges sharing an endpoint get different colors? This is a classic problem, for which a group of collaborators in the Summer 2024 Simons Institute research program on Sublinear Algorithms came up with a deterministic $O(m \log \Delta)$ algorithm that earned the Best Paper Award at STOC — one of the flagship conferences for theoretical computer science.

At first glance, this problem sounds similar to the notoriously difficult problem of graph coloring, where the goal is to color the *vertices* of a graph using as few colors as possible, so that adjacent vertices get different colors. For this latter problem, even getting a very crude approximation to the number of colors required is known to be NP-hard.

Somewhat surprisingly, in 1964 Vadim Vizing proved a remarkable result that gave pretty much the exact number of colors required for edge coloring. Let Δ be the maximum degree of a given graph — i.e., the maximum overall vertices v of the number of edges that have an endpoint at v. Since each edge incident on a vertex v must get a different color, it is clear that Δ colors are necessary for any edge coloring. Vizing gave an algorithm for coloring the edges of the graph that used at most $\Delta + 1$ colors! If n is the number of vertices and m is the number of edges, his algorithm runs in time $O(mn)$.

The edge coloring problem is more than a nice puzzle. It has applications in areas such as scheduling, communication channel assignment, and compiler design. To elaborate on one application, imagine a communication network where nodes communicate with their neighbors along edges. To avoid interference or cross talk in communicating with its neighbors, a node needs to use different frequencies on each of its edges. But we don't want to use too many frequencies since the spectrum is a scarce resource. Thinking of frequencies as colors, we get precisely the edge coloring problem.

Since the graphs involved in some applications can be quite large, there has been a great deal of interest in finding the most efficient algorithm for coming up with an edge coloring. In the 1980s, two independent sets of researchers gave an $\tilde{O}(m\sqrt{n})$ algorithm, and there it more or less stood for over 40 years. In Summer 2024, during the Simons Institute program on Sublinear Algorithms, Sepehr Assadi presented his latest work, where he gave an algorithm that achieved runtime $O(m \log \Delta)$ but used $O(\log n)$ more colors than the bound promised by Vizing's theorem. He also gave a randomized algorithm that produced a $(\Delta + 1)$ -coloring, running in expected time $O(n^2 \log n)$. He and Soheil Behnezhad, who was in the audience, started working on further improvements during this program. While neither the result Assadi presented nor the final result is sublinear, the program enabled the discovery of the latter result. Independently and simultaneously, Sayan Bhattacharya, Din Carmon, Martín Costa, Shay Solomon, and Tianyi Zhang also broke the $O(m\sqrt{n})$ barrier, giving a randomized algorithm running in $\tilde{O}(mn^{1/3})$. Both teams were intrigued by the fact that the techniques in these results were entirely different and decided to work together to see whether they could combine their ideas to get even better results. This collaboration bore fruit in the algorithm of Assadi, Behnezhad, Bhattacharya, Costa, Solomon, and Zhang that won the Best Paper Award at STOC '25.

To understand some technical details in these latest developments, it helps to start by reviewing Vizing's algorithm.

Vizing's algorithm starts with a graph G of maximum degree Δ. Without loss of generality, we can assume that G is connected, since we otherwise can treat each component on its own.

One idea in Vizing's algorithm is that any graph with maximum degree Δ can be partitioned into two graphs, each with maximum degree at most $\lceil \frac{\Delta}{2} \rceil$, using a technique called Eulerian partitioning. By adding one edge between pairs of odd-degree vertices, we can ensure that every vertex has even degree. Now a well-known result in graph theory says that we can find an "Eulerian tour" — i.e., a walk that goes through all edges exactly once. If we now put all edges occurring in odd positions on this tour in one component, and all edges in even positions in the other, we have the desired decomposition. Now recursively we can color each of these graphs with at most $\lceil \frac{\Delta}{2} \rceil + 1$ colors, which may overall use $\Delta + 3$ colors.

Since this exceeds the desired bound by 2, the algorithm uncolors the edges colored with two of these colors (say, the least frequently used colors) and colors them again one by one using the other $\Delta + 1$ colors. Since the maximum degree is Δ, every vertex must have at least one unused or free color around it. If both u and v have the same free color, then we can just color the edge (u, v) with this free color. If instead green is free at u and blue is free at v, then Vizing's algorithm performs a series of color swaps along paths whose edges are alternately colored green and blue. If even this fails, the algorithm recolors edges in an ingenious structure called the Vizing fan, and again ends up finding a color for (u, v). Each edge can take $O(n)$ time to recolor in this manner, leading to a recurrence that solves to $O(mn)$.

Some of the key observations leading to the overall improvement are:

1) All $O(n)$ uncolored edges at some stage can be categorized into $O(\Delta^2)$ categories depending on the free colors at the two endpoint vertices.

2) We can efficiently increase the number of uncolored edges in one category to be a $\Theta(1/\Delta)$-fraction of all uncolored edges in expected $O(n)$ time by again using alternating paths.

Photo by Bruce Damonte

3) All uncolored edges in one category can be colored in $O(n)$ time using a Vizing-like procedure since the alternating paths or fans for coloring these edges do not overlap with each other. (The simple fan structure in Vizing's algorithm does not work, but a more complex variant does.)

4) Repeating Steps 2 and 3 above $\tilde{O}(\Delta)$ times allows us to color all uncolored edges in expected $O(n\Delta) = O(m)$ time.

Putting all this together leads to an $O(m \log \Delta)$ algorithm that seems "simple enough" to also be fast in practice. ●

Optimal List Decoding

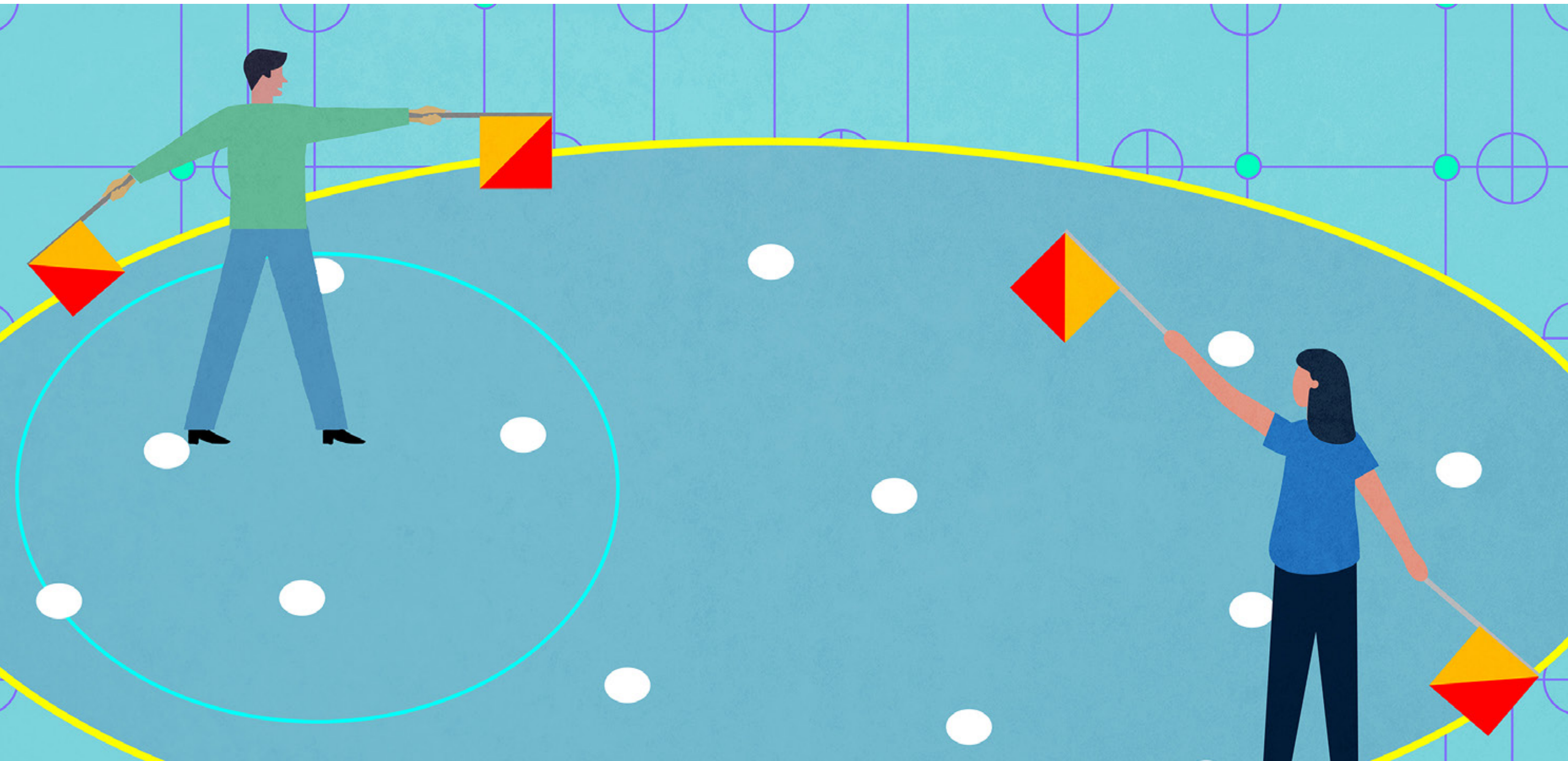
How a recent Simons Institute program helped push list decoding to its theoretical limit

Venkatesan Guruswami, director

In coding theory, erasures are the easy enemy: if we want to tolerate the loss of 20% of the symbols (chosen adversarially), we can do so with a code with redundancy equal to 20%, where we encode $0.8n$ data symbols into a redundant set of n codeword symbols. In fact, the classical and ubiquitous Reed–Solomon codes do the job, and even undergraduates in computer science are often taught the interpolation algorithm that fills in the missing positions.

Errors are trickier, as one does not immediately know which symbols are affected. Indeed, classical algorithms to tackle a 20% error rate (again for Reed–Solomon codes) require 40% redundancy, which is two times more than the erasure case. And if the errors are worst case and we demand unambiguous recovery, this increased redundancy is inherently required. However, a work-around called *list decoding* relaxes the requirement on the error-correction algorithm, allowing it to output, in the worst case, a small list of codewords within the target error bound. (The hope is that in typical cases the list will anyway have a unique element, so in practice this doesn't affect the utility of error correction, except it lets one handle pathological cases within the worst-case model.)

In this model, remarkably, one can deal with errors with the same efficiency as erasures! Two decades ago, the author and Atri Rudra showed that a simple variant of RS codes — *folded Reed–Solomon codes*, which bundle several consecutive RS symbols into one larger "mega-symbol" — can efficiently correct a fraction ρ of errors with redundancy approaching the optimal bound of ρ, for any target error rate $\rho \in (0, 1)$. This present article reports on progress forged by the Spring 2024 Simons Institute research program on Error-Correcting Codes, which essentially closed the chapter opened by this work.



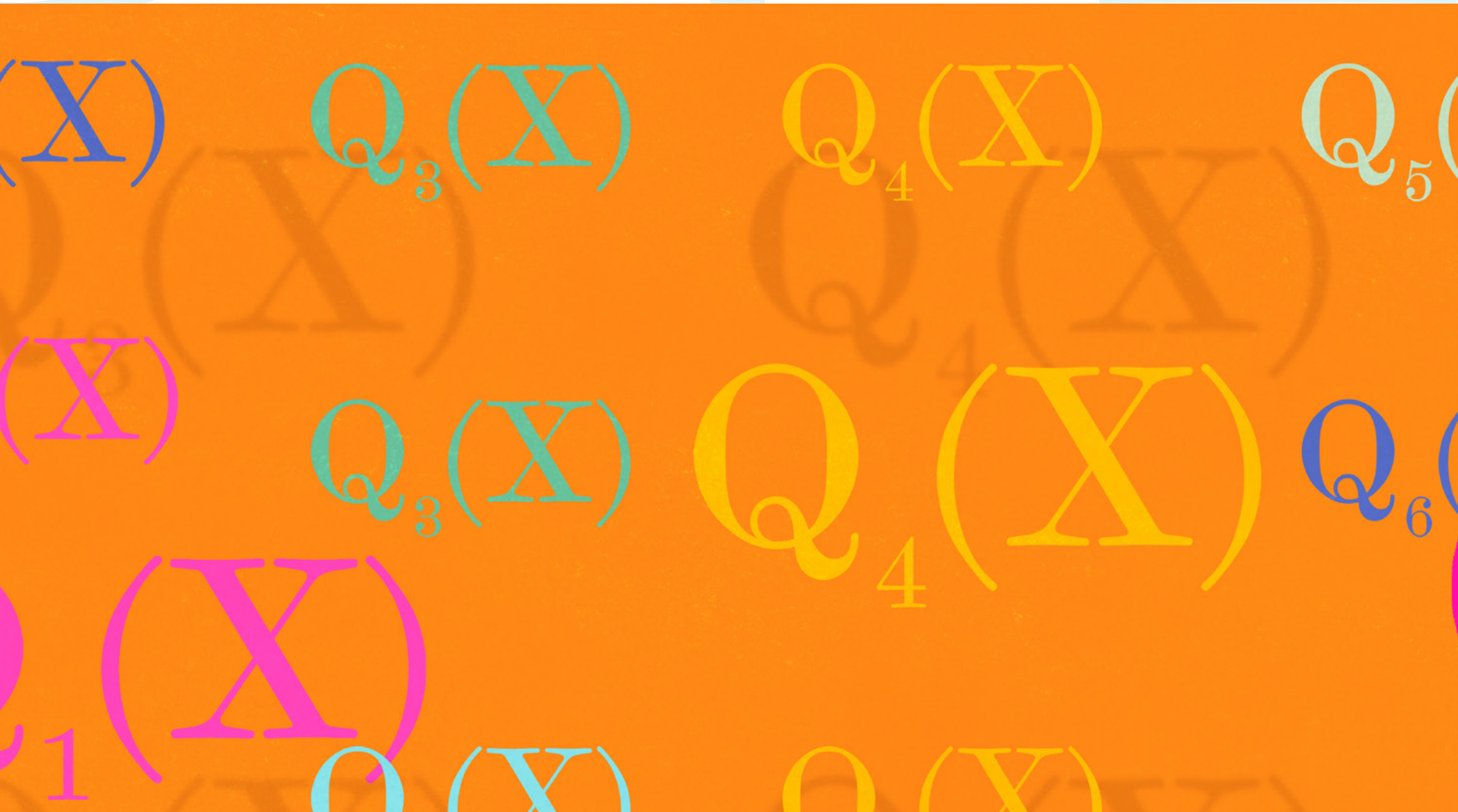
List-size for optimal decoding

As far as the trade-off between redundancy and error resilience goes, this result achieves the optimal trade-off. But a third parameter — the worst-case list-size — is also important, and when the redundancy is $\rho + \epsilon$, for some small $\epsilon > 0$, the result only established an upper bound of $n^{1/\epsilon}$ on the list-size, where n is the length of the code. It was known that the list-size cannot be any smaller than $1/\epsilon$, but this left a huge gap!

Though the list-size was large, it was later shown by the author and Carol Wang in the early 2010s that it is well structured in the sense that one can find a subspace of dimension $1/\epsilon$ containing the list. This led to constructions of other codes with improved list-size, but the situation for folded Reed–Solomon codes was not improved until 2018, when Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters showed that the list-size is bounded by $(1/\epsilon)^{O(1/\epsilon)}$ by elegantly pruning the subspace to zero in on the close-by codewords. The nice feature was that the list-size now had no degradation with n, but it was still exponentially off from the best-possible bound.

> “This is a textbook example of how the Simons Institute’s collaborative environment helps early-career researchers crystallize and exchange ideas, while benefiting from all the surrounding activity on closely related themes.”

Could the list-size be further improved by showing folded Reed–Solomon codes to be simultaneously optimal on all three aspects — error-fraction, redundancy, and list-size? Or could it be that we need to look elsewhere for such optimal codes? The answer wasn’t clear, and neither was an approach to attack the question apparent.



Two recent breakthroughs

In this context, a concurrent pair of recent student-authored works completely resolved the question, showing that, in fact, folded Reed–Solomon codes achieve a list-size of $1/\epsilon$, which as mentioned earlier is optimal!

In a paper that received the Best Paper Award at the 2025 ACM-SIAM Symposium on Discrete Algorithms (SODA), Shashank Srivastava showed a list-size bound of $O(1/\epsilon^2)$, a huge improvement over the previous exponential bounds and quadratically close to the optimal bound.

In a paper that received the Best Student Paper Award at the 2025 ACM Symposium on Theory of Computing (STOC), Yeyuan Chen and Zihan Zhang, two PhD students who began work on this problem during their visit to the Simons Institute's Spring 2024 program on Error-Correcting Codes, went a step further, and proved the optimal list-size bound of $1/\epsilon$.

These works not only achieve a long-sought-after milestone in error correction that simultaneously optimizes redundancy, list-decoding radius, and list-size, but do so most elegantly. As Yeyuan and Zihan explicitly acknowledge in their paper, their work was initiated during their visit to the Simons Institute. This is a textbook example of how the Simons Institute's collaborative environment helps early-career researchers crystallize and exchange ideas, while benefiting from all the surrounding activity on closely related themes. Their collaborations have continued apace beyond this work, and have led to three more papers, and counting.

At the heart of both the above works is the fact that folded Reed–Solomon codes have a so-called "subspace design" property. This property was already established in 2013 by the author and Swastik Kopparty for a different purpose, but its stunning power to lead to the optimal list-size came as a surprise even to experts. The revelation of the centrality and unifying power of subspace designs has fueled several exciting recent works and already had a lot of impact.

A work by Vikrant Ashvinkumar, Mursalin Habib, and Shashank Srivastava (to appear in SODA 2026) gives an algorithmic version of the above results that accomplishes the decoding task in time polynomial in both n and $1/\epsilon$. In a pair of works, Yeyuan and Zihan, together with Josh Brakensiek and Manik Dhar, have made striking connections between subspace designs and discrete Brascamp–Lieb inequalities, as well as random linear codes. Rohan Goyal and the author show that subspace design–based codes have optimal proximity gaps, an important concept in the analysis of succinct non-interactive arguments of knowledge (SNARKs) used in blockchain and related technologies.

References for readers who want to learn more

Capacity with folded Reed–Solomon codes: V. Guruswami and A. Rudra, "Explicit Codes Achieving List Decoding Capacity" (*STOC,* 2006; journal version in *IEEE Trans. Inf. Theory,* 2008). V. Guruswami and C. Wang, "Linear-Algebraic List Decoding for Variants of Reed–Solomon Codes" (*IEEE Trans. Inf. Theory,* 2013).

Pre-2024 list-size bounds: S. Kopparty, N. Ron-Zewi, S. Saraf, M. Wootters, "Improved List Decoding of Folded Reed-Solomon and Multiplicity Codes" (*FOCS,* 2018; *SICOMP,* 2023). I. Tamo, "Tighter List-Size Bounds for List-Decoding and Recovery of Folded Reed-Solomon and Multiplicity Codes" (*IEEE Trans. Inf. Theory,* 2024).

The new results: S. Srivastava, "Improved List Size for Folded Reed-Solomon Codes" (*SODA,* 2025). Y. Chen and Z. Zhang, "Explicit Folded Reed-Solomon and Multiplicity Codes Achieve Relaxed Generalized Singleton Bounds" (*STOC,* 2025).

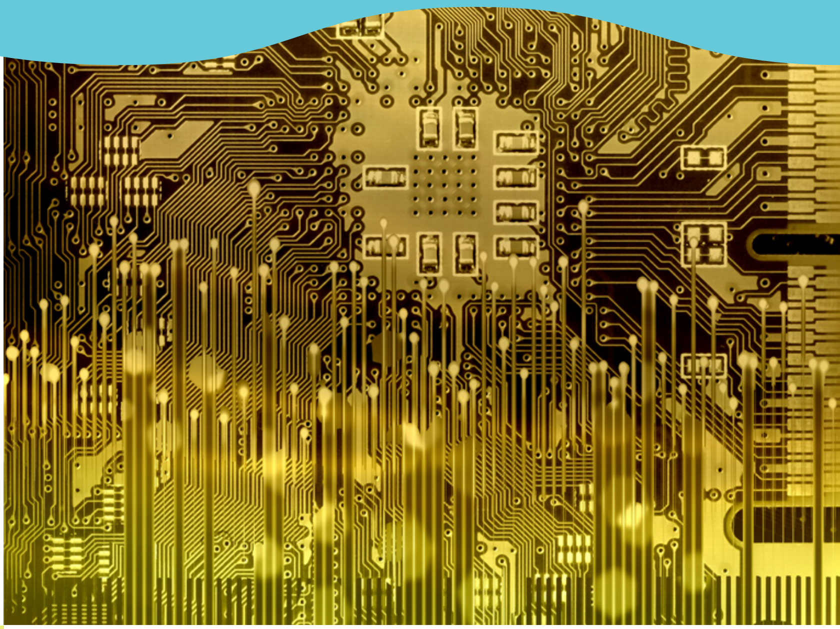
Subspace designs: V. Guruswami and S. Kopparty, "Explicit Subspace Designs" (*Combinatorica,* 2016). See also V. Guruswami and C. Xing, "List Decoding Reed-Solomon, Algebraic-Geometric, and Gabidulin Subcodes up to the Singleton Bound" (*STOC,* 2013) for the original notion in a different context.

Current and Upcoming Programs

Algorithmic Foundations for Emerging Computing Technologies
(Fall 2025)

Complexity and Linear Algebra
(Fall 2025)

Federated and Collaborative Learning
(Spring 2026)



Summer Cluster on Quantum Computing
(Summer 2026)

Pseudorandomness & High-Dimensional Expansion
(Fall 2026)

Spectral Theory Beyond Graphs
(Fall 2026)

Symmetry in Efficient Computation with Local Constraints
(Spring 2027)

Diffusion Generative Modeling
(Fall 2027)

Support for the Simons Institute

The Simons Institute gratefully acknowledges the individuals and institutions that provided us with financial support during 2024–25:

ADIA Lab
Algorand Foundation
Amazon Web Services
Amy Ambrose and Koyo Konishi
Anonymous
Apple
Association for Computing Machinery
Aakarsh Bengani
Peter and Nancy Bickel
Davide Bragetti
Bridgewater Associates
Jennifer Chayes and Christian Borgs
Fan Chung Graham
Rachel Cummings
Noah Daniels
Debasish Das
Defense Advanced Research Projects Agency
Department of Energy
Abhishek Ghose
Shafi Goldwasser
Ranelle Raquel Gomez Sr.
Google LLC
Parikshit Gopalan and Gagan Aggarwal
Venkatesan Guruswami
Daniel Gusfield
Soufiane Hayou
Kangmin Hu
Sheila Humphreys
Sandra Irani
Yuval Ishai
IVADO
Jane Street Group, LLC
Kristin Kane
Sampath Kannan
Dick Karp and Mary Jo Lazear
K. N. King
Scott Kirkpatrick
Miranda Lynch
Rachel and Karl Meyers
Stephen Mitchell
National Science Foundation
NTT Research Inc.
Office of Naval Research
Oski Technology/Nvidia
Sabesan Saidapet Pachai
Prabhakar and Srilatha Raghavan
Arvind Raghunathan and Bonnie Subramanian
Orna Resnekov and David Tennenhouse
Ronald Rivest
Roc360
Umit Sami
Lalitha Sankar
Scalar Capital Management and Linda Xie
Vira Semenova
Brian Shiratsuki
Jim and Marilyn Simons
Simons Foundation & Simons Foundation International
Vigyan Singhal and Sonali Bhushan
Stellar Development Foundation
Timothy Tangherlini and Margaret Kong
Éva Tardos
UC Noyce Initiative
Vijay Vazirani
VMware
Jae Oh Woo
Bin Yu and Ke-Ning Shen
Angela Zhou

All gifts and partnerships are matched one-to-one by Simons Foundation International.

To become an industry partner, make a gift to the Institute, or learn more about engagement possibilities, contact Director Venkatesan Guruswami (simonsdirector@berkeley.edu) or Senior Development Director Amy Ambrose (amyambrose@berkeley.edu).

Photo by Bruce Damonte

Simons Institute for the Theory of Computing

Simons Institute for the Theory of Computing
121 Calvin Lab #2190
UC Berkeley
Berkeley, CA 94720-2190

simons.berkeley.edu

To request this document in an alternative format, including large print, braille, or electronic formats, please contact Barry Bödeker at barry.b@berkeley.edu