

COMMUNICATIONS

CACM.ACM.ORG OF THE

ACM

04/2026 VOL.69 NO.04

Where Are the City Trees? Monitoring Urban Trees across the U.S. Using Generative AI

How Liberating Structures Can
Revolutionize Our Conferences

Communication Bias in Large Language
Models: A Regulatory Perspective

General Game Playing

Association for
Computing Machinery

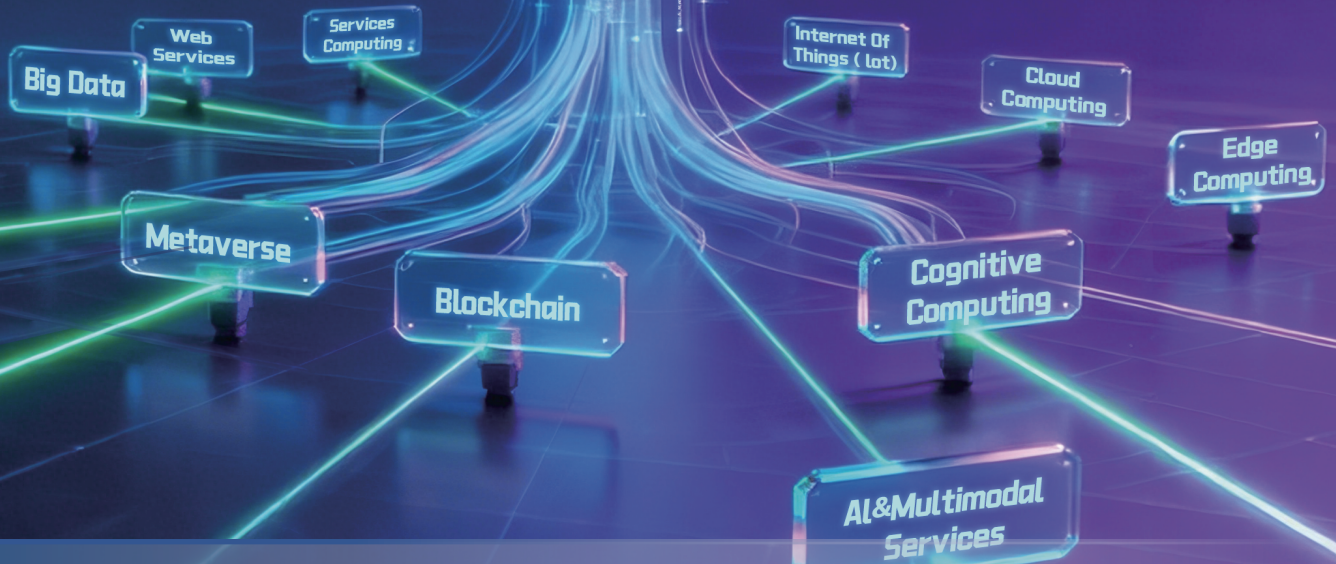
acm

2026 AIAAS Agentic AI as a Service

ICWS
Founded in 2003

Aug.22 - 25, 2026, Kuala Lumpur, Malaysia

CELEBRATING THE 24th GATHERING OF ICWS



- 1 2026 International Conference on AI and Multimodal Services (**AIMS 2026**)
- 2 2026 International Conference on Cognitive Computing (**ICCC 2026**)
- 3 2026 International Conference on Blockchain (**ICBC 2026**)
- 4 2026 International Conference on Metaverse (**METAVERSE 2026**)
- 5 2026 International Conference on Cloud Computing (**CLOUD 2026**)
- 6 2026 International Conference on Edge Computing (**EDGE 2026**)
- 7 2026 International Conference on Big Data (**BigData 2026**)
- 8 2026 International Conference on Internet of Things (**ICIOT 2026**)
- 9 2026 International Conference on Web Services (**ICWS 2026**)
- 10 2026 International Conference on Services Computing (**SCC 2026**)



Submission Deadlines

Early-Birds Submission: 3/20/2026
Regular Submission: 5/21/2026
Camera-Ready Manuscripts: 6/28/2026

Contact

confs@servicessociety.org
www.icws.org



ICWS.ORG

The Proceedings of SCF 2026 will be published in the LNCS by Springer. Currently the Abstracting and Indexing services covered by Springer's data feeds for LNCS proceedings include ISI Conference Proceedings Citation Index (included in Web of Science), Engineering Index EI (Compendex and Inspec databases), DBLP, Google Scholar, IO-Port, MathSciNet, Scopus, and ZBIMath.



Leading not-for-profits organization (501(c)(3))
dedicated for serving 30,000+ worldwide
services computing professionals

NEW BOOK RELEASE



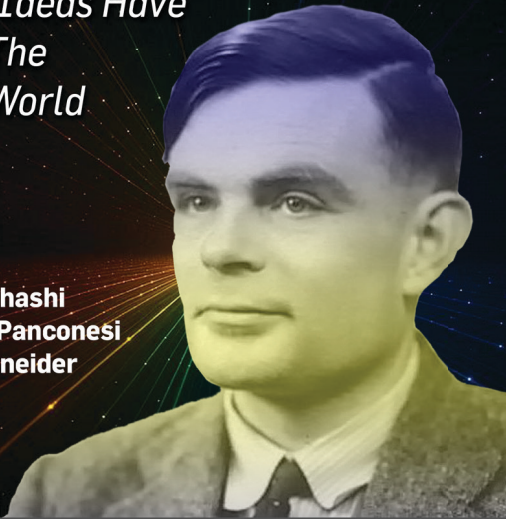
ACM BOOKS

Collection III

Turing's Children

How His Ideas Have Shaped The Modern World

**Devdatt Dubhashi
Alessandro Panconesi
Gerardo Schneider**



ASSOCIATION FOR COMPUTING MACHINERY

Turing's Children

How His Ideas Have Shaped the Modern World

**Devdatt Dubhashi
Alessandro Panconesi
Gerardo Schneider**

ISBN: 979-8-4007-3177-8

DOI: 10.1145/3731975

Exploring the profound legacy of Alan Turing, and how his five great ideas have been developed by the winners of the Turing Prize to shape the digital world we live in today.

These five groundbreaking concepts—universal computability and insolvability, cryptography, electronic digital computers and software, artificial intelligence (AI), and computational biology—serve as the pillars of the narrative. *Turing's Children* makes the complex ideas in the work of the Turing Prize winners accessible to a broad audience. It traces the evolution of these ideas from abstract mathematical principles to their real-world applications, demonstrating their influence on everyday life. Readers will discover how Turing's legacy is woven into the fabric of modern technology: from the smartphones in our pockets and secure online banking, to the development of life-saving drugs and the conversational abilities of AI systems like ChatGPT.

The book weaves together historical anecdotes, profiles of pioneering scientists, and vivid examples of current innovations. It offers a comprehensive view of how Turing's vision continues to inspire breakthroughs across science and technology.

<http://books.acm.org>

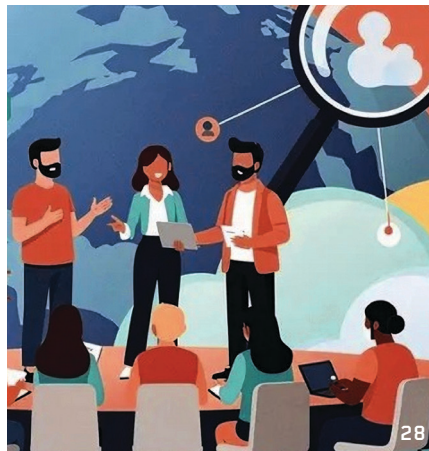
News

- 12 **How Avatars Make People Feel**
Studies examine the effects of avatars on their users and others.
By Neil Savage
-
- 15 **AI Multiplies for Drug Discovery**
Throttling access to a leading AI tool for drug discovery has spawned many variants and advances. But there is still some way to go.
By Chris Edwards
-
- 18 **In Memoriam**
David J. Farber
“Grandfather of the Internet” who excelled in academic and industry leadership roles made networking his life.
By Simson Garfinkel and Eugene H. Spafford
-
- 21 **When Seeing Isn’t Believing**
New research reveals how AI can alter human memories—and reshape our sense of reality.
By Alex Wright
-
- 24 **ACM Digital Library**
Evolving the ACM Digital Library
Open access, Basic and Premium, and what we learned along the way.
By Jack Davidson and Wayne Graves

Last Byte

- 96 **Upstart Puzzles**
TimesSquare
Squaring up the grid.
By Dennis Shasha

Opinion



- 5 **Cerf’s Up**
It’s about Time
By Vinton G. Cerf
-
- 6 **Letters to the Editor**
AI and the Evolving Role of the Scientific Paper
-
- 9 **Career Paths in Computing**
My Journey in AI: Inspired by a Nobel Laureate and an Unusual Fish
By Qiang Yang
-
- 10 **BLOG@CACM**
Teaching Programmers a Survival Mindset
Oscar M. Bonastre looks at trends shaping the teaching of programming.
-
- 28 **Opinion**
From Passive to Participatory: How Liberating Structures Can Revolutionize Our Conferences
Tracking conference restructuring toward generating new ideas and discussing established work.
By Daniel Russo and Margaret-Anne Storey



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/revolutionize-conferences>

Opinion

- 32 **Opinion**
The Coming Commoditization of Computational Thinking
Reimagining professional and educational practices for an AI-augmented future.
By Douglas C. Schmidt
-
- 35 **Technology Strategy and Management**
The Importance of Geopolitics in AI Development
Considering the unique challenges for learning from past geopolitical tensions.
By Mari Sako
-
- 39 **Kode Vicious**
A Trunk Full of Swords
The shiniest tool might cut the deepest.
By George V. Neville-Neil
-
- 41 **Opinion**
Redefining the Software Engineering Profession for AI
Without the hiring of early-in-career developers, the profession’s talent pipeline will collapse, and organizations will face a future without the next generation of experienced engineers.
By Mark Russinovich and Scott Hanselman
-
- 45 **Opinion**
AI Individualism: What Are Social Relationships in the Age of Artificial Intelligence?
Seeking a hybrid social design approach that blends human and AI relationships in socially responsible ways.
By Petter Bae Brandtzaeg, Marita Skjuve, and Asbjorn Følstad

Research and Advances

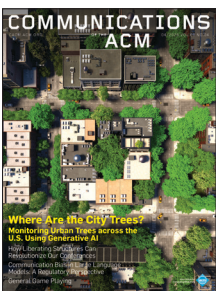


50

50 Where Are the City Trees? Monitoring Urban Trees across the U.S. Using Generative AI
For years, conducting an urban tree inventory was a luxury only large, highly resourced cities could afford. A new AI-based approach not only brings this capability within reach to smaller cities but also requires significantly less time.
By Adnan Firoze, Akshaj Uppala, Lindsay Darling, Raymond A. Yeh, Bedrich Benes, Brady Hardiman, Songlin Fei, and Daniel Aliaga



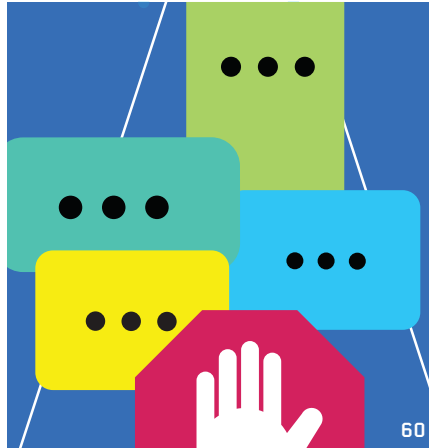
Watch the authors discuss this work in the exclusive *Communications* video. <https://cacm.acm.org/videos/city-trees>



About the Cover: Seen from high above, streets and buildings dominate the cityscape. But as we zoom in, trees enter the picture—trees that not only provide shade for residents but also capture excess water and purify the air. Typically, cities keep track of these precious trees through laborious, expensive manual tree counts. But a new generative-AI-based

approach that uses satellite images aims to change this, detecting trees otherwise lost in the concrete jungle. Cover illustration by JaceyTec.

Research and Advances



60

60 Communication Bias in Large Language Models: A Regulatory Perspective
While new regulatory frameworks address fairness, accountability, and safety in AI systems, they often fail to directly mitigate the subtle “communication bias” in LLMs that can distort public discourse and democratic processes.
By Adrian Kuenzler and Stefan Schmid

72 General Game Playing
General game players are computer systems capable of playing strategy games based solely on game descriptions supplied at “runtime.” While general game playing is intellectually engaging and fun, it also serves as a laboratory for practical applications, and provides a theoretical framework for defining rationality.
By Michael Genesereth

Research and Advances



80

80 Learning to Flow (Between Datacenters)
Leveraging ML models to directly learn network flow configurations from empirical data can deliver robustly high performance, surpassing that of demand-prediction methods.
By Yarin Perry, Srikanth Kandula, Ishai Menache, Michael Schapira, and Aviv Tamar

Research Highlights

- 88 Technical Perspective**
Fusing LLMs with Completion Engines for Automated Program Repair
By Gabriele Bavota
- 89 Copiloting the Copilots for Automated Program Repair**
By Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang



ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

Acting Executive Director and CEO
Patricia Ryan
Deputy Executive Director and COO
Patricia Ryan
Director, ACM Digital Library
Wayne Graves
Director, Office of Financial Services
James Schembari
Director, Office of SIG Services
Donna Cappo
Director, Office of Publications
Scott E. Delman

ACM COUNCIL

President
Yannis Ioannidis
Vice-President
Elisa Bertino
Secretary/Treasurer
Rashmi Mohan
Past President
Gabriele Kotsis
Chair, SGB Board
Neha Kumar
Co-Chairs, Publications Board
Wendy Hall and Divesh Srivastava
Members-at-Large
Tom Crick, Odest (Chad) Jenkins, John Kim, Tanara Lauschner, Alison Derbenwick Miller, Alejandro Saucedo, Michelle Zhou
SGB Council Representatives
Adrienne Decker, Jens Palsberg, and Vivek Sarkar

BOARD CHAIRS

Education Board
Elizabeth Hawthorne and Alison Derbenwick Miller
Practitioners Board
Marlene Mhangami and Sophie Watson
Digital Library Board
Jack Davidson

TOPIC AND REGIONAL COUNCIL CHAIRS

Diversity, Equity, and Inclusion Council
Timothy Pinkston
Technology Policy Council
Virginia Dignum and Jeanna Matthews
ACM Europe Council
Rosa Badia
ACM India Council
Meenakshi D'Souza
ACM China Council
Huadong Ma

PUBLICATIONS BOARD

Co-Chairs
Wendy Hall and Divesh Srivastava
Board Members
Jonathan Aldrich; Rick Anderson; Tom Crick; Jack Davidson; Mike Heroux; Michael Kirkpatrick; James Larus; Marc Najork; Beng Chin Ooi; Mauro Pezzè; Francesca Rossi; Bobby Schnabel; Stuart Taylor; Bhavani Thuraisingham; Adelinde Uhrmacher; Philip Wadler; John West; Min Zhang

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

STAFF

DIRECTOR OF PUBLICATIONS
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor, ACM Magazines

Ralph Raiola
Senior Editor
John Stanik
Managing Editor
Thomas E. Lambert
Senior Editor/News
Lawrence M. Fisher
Web Editor
David Roman
Editorial Assistant
Sophia Navalta

Art Director

Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balauqiot
Production Manager
Bernadette Shade
Intellectual Property Rights Coordinator
Barbara Ryan
Advertising Sales Account Manager
Ilia Rodriguez

Columnists

Saurabh Bagchi; Michael L. Best; Michael A. Cusumano; Peter J. Denning; Thomas Haigh; Leah Hoffmann; Mari Sako; Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS

Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

REGIONAL SPECIAL SECTIONS

Co-Chairs
Virgilio Almeida, Haibo Chen, Jakob Rehof, and P J Narayanan
Board Members
Sherif G. Aly; Panagioti Fatourou; Chris Hankin; Sue Moon; Tao Xie; Kenjiro Taura

WEBSITE

<https://cacm.acm.org>

WEB BOARD

Chair
James Landay
Board Members
Marti Hearst; Jason I. Hong; Wendy E. MacKay

AUTHOR GUIDELINES

<https://cacm.acm.org/author-guidelines/>

COMPUTER SCIENCE TEACHERS ASSOCIATION

Jake Baskin
Executive Director

EDITORIAL BOARD

EDITOR-IN-CHIEF
James Larus
eic@cacm.acm.org

SENIOR EDITORS

Andrew A. Chien
Moshe Y. Vardi

EDITORS, IN MEMORIAM SECTION

Simson L. Garfinkel
Eugene H. Spafford

NEWS

Board Members

Siobhán Clarke; Lance Fortnow; Irwin King; Mei Kobayashi; Rajeev Rastogi; Vinoba Vinayagamoorthy

OPINION

Co-Chairs

Jeanna Neefe Matthews and Chiara Renso

Board Members

Roland Aydin; Saurabh Bagchi; Mike Best; Judith Bishop; Florence M. Chee; Danish Contractor; Lorrie Cranor; Janice Cuny; Jeremy Epstein; Armando Fox; Ophir Frieder; James Grimmelmann; Mark Guzdial; Mark D. Hill; Brittany Johnson; Bran Knowles; Raula Gaikovina Kula; Tim Menzies; Beng Chin Ooi; Anne-Cécile Orgerie; Alessandra Raffaetà; Francesca Rossi; Eve Schooler; R. Benjamin Shapiro; Len Shustek; Bernd Stahl; Stuart Taylor; Loren Terveen; Marshall Van Alstyne; Matt Wang; Robert West

PRACTICE

Chair

Terence Kelly

Board Members

Satish Chandra; Vincent Hellendoorn; Mike Hoyer; Pankaj Jalote; Saurabh Kadekodi; Donald Kossmann; Vinay Kulkarni; Stan Park; Steve Schirripa; Arie van Deursen; Victor Yodaiken

RESEARCH AND ADVANCES

Co-Chairs

m.c. schraefel and Premkumar T. Devanbu

Board Members

Alan Bundy; Peter Buneman; Haibo Chen; Monojit Choudhury; Jane Cleland-Huang; Gerardo Con Diaz; Anna Cox; Kathi Fisler; René Foster; Rebecca Isaacs; Trent Jaeger; Fabio Kon; Ben C. Lee; David Lo; Renée Miller; Ankur Moitra; Sarah Morris; Abhik Roychoudhury; Katie A. Siek; Daniel Susser; Charles Sutton; Thomas Zimmermann

RESEARCH HIGHLIGHTS

Co-Chairs

Shriram Krishnamurthi and Caroline Appert

Board Members

Martin Abadi; Sanjeev Arora; Maria-Florina Balcan; David Brooks; Selcuk Candan; Stuart K. Card; Jon Crowcroft; Lieven Eeckhout; Gernot Heiser; Takeo Igarashi; Nicole Immerlica; Srinivasan Keshav; Sven Koenig; Karen Liu; Claire Mathieu; Joanna McGrenere; Tamer Özsu; Tim Roughgarden; Chirag Shah; Guy Steele, Jr.; Wang-Chiew Tan; Robert Williamson; Andreas Zeller

Association for Computing Machinery (ACM)

1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 869-7440; F (212) 869-0481

ACM Copyright Notice

Copyright © 2026 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@hq.acm.org or fax (212) 869-0481.

For other copying of articles that carry a code at the bottom of the first or last page or screen display, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center; www.copyright.com.

Subscriptions

An annual subscription cost is included in ACM member dues of \$99 (\$40 of which is allocated to a subscription to *Communications*); for students, cost is included in \$42 dues (\$20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is \$269.

Single Copies

Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

ACM ADVERTISING DEPARTMENT

1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 626-0686
F (212) 869-0481

Advertising Sales Account Manager

Ilia Rodriguez
ilia.rodriguez@hq.acm.org

Media Kit acmmediasales@acm.org

COMMUNICATIONS OF THE ACM

(ISSN 0001-0782) is published monthly by ACM Media, 1601 Broadway, 10th Floor New York, NY 10019-7434 USA. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER

Please send address changes to *Communications of the ACM* 1601 Broadway, 10th Floor New York, NY 10019-7434 USA

Printed in the USA.



Association for
Computing Machinery





Vinton G. Cerf

DOI:10.1145/3799690

It's about Time

FOR SOME READERS, this essay will be old news. If it is, please consider responding to the implicit and explicit questions raised here. If it is not old news, please give some thought to practical responses that might mitigate the potential hazard of time-stamp variables wrapping around to reset the interpretation of the time variable to an unintended previous time.

There are two problems: One will hit in 2036^a and the other in 2038.^b Let's take the first one. The Network Time Protocol (NTP) uses a 64-bit timestamp. The first 32 bits represent seconds since Jan. 1, 1900, and the last 32 bits are fractions of seconds. The first part, seconds, will wrap around back to zero after 136 years, 4 days, 6 hours, 28 minutes, and 16 seconds. That will occur on Feb. 6, 2036. At that point, the time stamp would be interpreted as Jan. 1, 1900. NTPv4 uses the concept of *eras* to deal with this, but older systems may think they have been transported back to the beginning of the 20th century.

The second problem hits in 2038, when UNIX and its derivative systems that rely on counting seconds since the "UNIX epoch" (Jan. 1, 1970) using a signed 32-bit integer reach maximum positive value. When the clock hits 03:14:07 UTC on Tuesday, Jan. 19, 2038, the clock value *goes negative*, and the system will think it is 08:45 pm on Dec. 13, 1901.

Many systems have been upgraded to 64 bits for seconds, which will keep running correctly for 292 billion years—if they are still around to compute! The big problem is legacy imple-

mentations in devices not normally subject to regular software updates. Think Internet of Things devices or other devices users do not bother to update. Medical devices, embedded sensors, or other "install-and-forget" appliances are all at risk if they rely on these time-of-day mechanisms.

There is likely COBOL code from the 1950s that uses 32-bit date/time variables. Apart from re-hiring retired COBOL programmers (YAY!) as happened with the famous Y2K two-digit year problem that threatened at midnight on Dec. 31, 1999, it might be timely to think about introducing typed variables known to represent time stamps at least in languages supporting types. Some do not, of course. Ideally, we would introduce testing for imminent wrap-around conditions and take some corrective action.

There are likely several other similar "gotchas" where variables are not sized in anticipation of the hazards of wrap-around consequences. Perhaps a campaign to identify open source libraries in need of repair would be helpful. One wonders whether *vibe coding*

How might we mitigate the potential hazard of time-stamp variables wrapping around to reset the interpretation of the time variable to an unintended previous time?

systems could be trained to recognize and avoid wrap-around hazards. Perhaps a bug-finding reward system would also be helpful. That practice seems to have worked reasonably well for other kinds of bugs.

One wonders what other counter-based constructs subject to overflow are lurking in code we rely on daily to operate correctly. Could we train AI programming tools to look for potential overflow problems before they are embedded in widespread software? Could we train programmers to avoid making these mistakes in the first place?

As an undergraduate at Stanford in the early 1960s, I took classes in numerical analysis from George Forsythe and Gene Golub, where I learned that finite significant digits could dwindle to zero when certain kinds of computations were done, such as matrix inversion. Scary scenarios of planes flipping over in flight were offered by ACM A.M. Turing Award recipient William Kahan at the annual Heidelberg Laureate Forum. Kahan received this recognition for his pioneering work in digital numerical analysis. There are potential computational hazards lurking everywhere.

Setting aside all the various apocalyptic scenarios associated with artificial intelligence, these bread-and-butter pedestrian computations strike me as just as scary because of their predictability. I am sure some of you are already all over this problem; others, not so much. I'd like to hear from you. □

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

a https://en.wikipedia.org/wiki/Network_Time_Protocol#Timestamps

b https://en.wikipedia.org/wiki/Year_2038_problem

AI and the Evolving Role of the Scientific Paper

VINTON G. CERF'S February 2026 *Communications* Cerf's Up column, "Does AI Now Represent a Paradigm Shift?" rightly characterizes modern AI systems as a departure from traditional computing models: probabilistic rather than deterministic, emergent rather than explicitly programmed. But if AI represents a paradigm shift in how computation is performed, the more profound shift may lie in what AI now makes *possible* for scientific communication itself—across all domains, not just computer science.

Much of today's scientific and technical literature still treats the paper as a static narrative artifact: prose, figures, and perhaps equations describing results assumed to be reproducible in principle. That assumption has always been fragile—a reader cannot truly verify a claimed medical treatment effect, a robotics navigation algorithm, or a financial risk model from textual description alone. What has changed is that AI now provides the tools to address this limitation at scale.

Consider common claims across domains: "Our treatment protocol improved outcomes by 15%" (medicine), "Our robot navigated cluttered environments with 95% success" (robotics) [or] "Our model predicted market downturns with 80% accuracy" (finance). Traditionally, verifying such claims required access to proprietary patient data, expensive hardware platforms, or confidential financial records—barriers that made true reproducibility impossible for most readers.

AI systems can now generate synthetic patients that preserve statistical properties of clinical populations, simulate robotic environments with realistic physics and sensor noise, and create synthetic market scenarios reflecting historical volatility patterns. These are not perfect replicas, but they are *inspectable* representations that allow readers to probe methodology, test edge cases, and verify claims in ways that prose descriptions never could.

The paradigm shift, then, extends beyond how we build software to how we communicate findings across all empirical disciplines. Static papers were adequate when reproducibility was largely aspirational. They are inadequate when we now possess tools to make verification practically accessible. Just as the Web demanded new norms for information sharing, AI may require us to treat papers not as endpoints, but as interfaces—entry points into executable, explorable systems.

This is not a call for every publication to distribute proprietary data or production infrastructure. Rather, it is an invitation to establish new minimal standards for evaluability in an AI-enabled research ecosystem. A medical paper might include a synthetic patient simulator; a robotics paper, a minimal navigation environment; a finance paper, representative synthetic time series. Even simplified artifacts can dramatically improve transparency and understanding compared to prose alone.

Importantly, this standard should apply universally—not just to papers *about* AI, but to papers *enabled by* AI's capacity to generate representative computational artifacts across domains.

If AI represents a paradigm shift in computing, it may ultimately be remembered as the force that enabled a long-overdue shift in scientific communication itself: from static descriptions of results to executable, inspectable repre-

Much of today's scientific and technical literature still treats the paper as a static narrative artifact.

sentations of empirical claims—not just in computer science, but across the full spectrum of quantitative research.

Uri Kartoun,
Cambridge, MA, USA

Author's response:

I resonated with this proposition. If AI helps to improve the scientific method, we are all well served.

Vinton G. Cerf,
McLean, VA, USA

Ascertaining Certifiability

We read the January 2026 *Communications* article "Certificates in AI: Learn but Verify" by Barrett, Henzinger, and Seshia, and with great interest. The authors articulate a timely and important vision: moving beyond purely statistical evaluation toward AI systems that produce outputs together with verifiable certificates.

Communications readers may be interested to know of a theoretical foundation for this paradigm developed in our work "Models That Prove Their Own Correctness" (Amit, Goldwasser, Paradise, Rothblum; *arXiv* May 2024, *NeurIPS* December 2025). In that paper, we formalize the idea of a self-proving model that proves the correctness of a generated answer to a verifier via an interactive proof. Certificates considered by Barrett et al. are a special case of interactive proofs in which the interaction with the verifier consists of a single message sent by the prover (see Related Work section of Barrett et al.).

Self-proving models represent a synthesis of learning and verification. They retain the expressive power of modern machine learning while inheriting the accountability of proof systems. We believe this combination offers a promising foundation for addressing one of the central challenges of the current AI era: how to obtain trustworthy behavior from increasingly capable yet untrusted models.

Barrett et al. beautifully connect this vision to the history of formal methods.

Our paper provides a complexity-theoretic foundation (with soundness and completeness guarantees adapted to the setting of generative AI). We also provide learning algorithms to train models that generate per-output proofs. In particular, we introduce Transcript Learning (TL) and Reinforcement Learning from Verifier Feedback (RLVF), and prove convergence guarantees for training autoregressive self-proving models using TL and RLVF. These results are complemented by small-scale interpretable experiments. Since the appearance of our work, variants of RLVF (sometimes referred to as RLVR) have been adopted in a range of settings, including code generation, medical reasoning, and multi-modal tasks.

It is not a coincidence that both works arrive at essentially the same framework, one from computational complexity theory and the other from formal verification. In fact, we believe it reflects a deeper scientific convergence. Modern machine learning has produced models of unprecedented capability, far outpacing our ability to reason formally about their behavior or to provide guarantees of correctness, safety, or reliability. At the same time, complexity theory and formal verification have, for decades, studied how powerful computational entities (for example, agents) could justify their outputs to a weaker verifier. The recent emergence of self-proving models and in particular certificate-producing models, can be seen as the natural point at which these two traditions meet. We look forward to continued scientific dialogue between these communities in service of trustworthy generative AI.

Shafi Goldwasser and Orr Paradise,
Weizmann Institute of Science,
Rehovot, Israel

Authors' response:

We thank Goldwasser and Paradise for their recommendation to think about connections between the important roles certificates play in complexity theory and in formal methods. We agree that the two approaches are synergistic and share a common vision for trustworthy generative AI. We hope that both works will help inspire further research in support of this vision.

Thomas Henzinger,
Institute of Science and Technology
Austria, Klosterneuburg, Austria

Often the best method for a machine to use is not the method that people use to solve the same problem.

Impacted Wisdom

If one revises the January 2026 *Communications* article “Shaping AI’s Impact on Billions of Lives” by replacing “AI” with more general terms such as “computer,” “computerized,” or “computer systems,” and makes the necessary minor grammatical corrections, the result is just as relevant and valid as the original. Long before the recent surge of attention to “artificial intelligence,” people were concerned about the impact of computer systems on employment, education, healthcare, information distribution, media/entertainment, governance, national security, and science—the very areas discussed in this article. There have long been people warning about the dangers of automation while others tout the tremendous benefits of computers. That has not changed.

Given the absence of a widely accepted definition that would allow us to distinguish products that exhibit intelligence from routine applications of IT, this situation should not surprise anyone. It should make us ask why we use such a fuzzy term at all. The hype gives the impression that AI experts are breathing life into inanimate objects rather than building mechanical systems that are often poorly understood. The term AI also seems to block clear thinking about liability.

The authors liken AI systems to other innovations such as aviation, television, and the Internet. However, each of those inventions gave humankind the ability to do things that we could not do before. We know from the work of researchers in the field of computability (such as Alan Turing) that this is not true of AI. We also know that AI systems offer no new capabilities because they can be successfully simulated on conventional hardware.

One property that most systems

identified as “AI” have is that they imitate humans. They either imitate our externally visible behavior or imitate methods that humans seem to use. We should be asking whether this is a good idea. Photocopy machines do not imitate scribes or forgers; they do their job better than people did. Often the best method for a machine to use is not the method that people use to solve the same problem.

The real impact of AI may be to slow progress as resources are diverted from other approaches.

David Lorge Parnas,
McMaster University,
Hamilton, Ontario, Canada

Authors' response:

We believe the high impact of AI over conventional computing is becoming more widespread. Here are examples of new capabilities on formerly “intractable” problems where the deep neural network version of AI has already made advances well beyond conventional computer systems, some since we submitted our paper:

► **Game Playing:** At the ACM Turing Centenary Celebration on June 15, 2012, I asked Turing Laureate Ken Thompson when he thought a computer would beat a human champion at Go. He said it was “way, way out.” When the audience of hundreds was polled, the most popular choice was 20 to 30 years out, and only a single person in the audience thought it would be under 10 years. Less than four years later, AlphaGo beat Go wunderkind Lee Sedol.

► **Science:** AlphaFold made tremendous progress on the long-standing protein-folding challenge, and just six years after its release AlphaFold was recognized by a Nobel Prize. Another Nobel laureate has said it advanced the field by 20 years. Today, more than two million scientists use AlphaFold.

► **Healthcare:** The largest randomized control trial of medical AI use (more than 100,000 women) compared a radiologist plus AI versus two radiologists for two-year follow-up visits for women diagnosed with breast cancer. AI added 29% more cancers detected at a 44% reduced workload with fewer cancers in the following visit, and when found, the cancers were less aggressive.

► **Programming:** Surveys find that most programmers are now using AI

tools to improve productivity. As concrete examples, studies on Github Copilot and Amazon's CoreWhisper conclude that programmers using AI completed tasks more than 50% faster.

► Parnas writes that AI offers no new capabilities and that “The real impact of AI may be to slow progress as resources are diverted from other approaches.”

Together with my co-authors, we believe AI is already demonstrating new capabilities across a remarkably broad set of fields, and will positively impact the world profoundly across many areas, including those highlighted in our paper.

I love that our field moves so fast that we do not need to wait long to resolve disagreements. We will all know soon enough whether AI slows progress or advances it and provides new capabilities.

David Patterson,
Berkeley, CA, USA

Whither Forecasting

Peter Denning's December 2025 *Communications* The Profession of IT column, “Forecasting,” begins by explaining the traditional roles of a model, the most important of which is to predict the modeled phenomenon's future behavior. Before a model can be used safely for prediction, it must be subjected to validation that shows that it accurately captures the phenomenon's past behavior.

Denning shows that because LLMs derive their outputs entirely from the statistics of their training data, with no additional checks for the truth of that data, LLMs are prone to generate nonsense, to hallucinate, making it dangerous to accept their outputs as reliable predictions, particularly when a mistake can be costly or deadly. Examples of LLM tasks with high rates of hallucination are those requiring knowledge of science, law, mathematics, and logic.

Denning's explanation suggests a metamodel of LLMs, and of ANNs in general, that allows predicting what an LLM can accurately predict and what it cannot. The metamodel follows:

An LLM is a good model of a language—usually English—no more and no less. And this is “language” in the strictest sense, that is, what makes a grammatically correct utterance in the

An LLM is a good model of a language—usually English—no more, no less.

language, independent of meaning and truth. An LLM predicts the next word likely to appear in text that follows the prompt, based on the mass of text in its learning data, which is typically a collection of documents written by writers who are fluent, if not native, in the language, usually English. Consequently, the text that an LLM generates from a prompt resembles, in my experience, text written by a native speaker of the language. Indeed, the most common legitimate use of ChatGPT in my university and in my research field is for writers, who are not native in English to ask ChatGPT to rewrite into grammatically correct, native English, with no embellishment, content that they have written themselves.

The data in the LLM does not give it the ability to know what is the truth, but no model of language in the strictest sense does, because people who make mistakes, lie, or bullshit can do so with correct grammar. A completely different model, independent of the LLM, would be needed for an ANN to predict truth. However, as Denning explains elsewhere,^a deciding what is truth is very difficult, because in the end, truth is what one community says is true, and different communities my have different truths.

All this explains why an LLM is very good at being a native writer of its language and doing some linguistic functions with the language, such as searching, but it is very bad at doing other tasks for which the language is not a model. For any of these other tasks, an ANN that is a well-trained model of that task must be used. Fundamentally, our problem is that we are trying to use a good model of a language as a model for more

a <https://doi.org/10.1145/3726009>

than a language. It is like expecting a good chess-move predictor to predict good moves for Go or expecting a good weather predictor to accurately predict the future value of a stock.

The metamodel tells how to predict whether an LLM can do a given task. When you must decide whether an LLM can do a task, you have to ask what is the LLM really modeling, that is, about what phenomena does the LLM predict well? The LLM will be good for the task if and only if these good predictions facilitate the task. If not, and the LLM is used to do the task, the LLM would make whatever predictions it can from the prompt. The output, though good for the LLM's model, will not be relevant to the task, and will be judged as hallucination. Thus, ChatGPT does a good job of writing a legal brief in the correct format and in fluent, native English, but the legal brief may reference non-existent statutes and court cases.

I personally think that the very fact that an LLM hallucinates—that is, bullshits—so fluently in native English leads humans to believe that the LLM is intelligent,^b and thus to ask the LLM to do tasks for which it has not learned a good model.

Daniel Berry,
Waterloo, Ontario, Canada

Author's response

That's right. There is no known way to build a “truth detector” so that the augmented LLM can distinguish true from false outputs. The problem is that “truth” is very slippery. Different communities can have different and contradictory truths. Science is the best approach we have evolved for finding statements confirmed by almost all the evidence and refuted by little or no evidence. Science depends on many experiments, debates, disagreements, controversies, and conversations before a claim evolves into widely accepted truth. I do not see how a machine can do this.

Peter J. Denning,
Monterey, CA, USA

b <https://doi.org/10.1109/MC.2025.3547458>

Communications welcomes your opinion. To contribute a letter to the editor, please limit your comments to 500 words or less and send to letters@cacm.acm.org

© 2026 Copyright held by the owner/author(s).



CAREER PATHS IN COMPUTING

DOI:10.1145/3799223

My Journey in AI: Inspired by a Nobel Laureate and an Unusual Fish



NAME

Qiang Yang

BACKGROUND

Born in China. Educated in China and the U.S.

CURRENT JOB TITLE/EMPLOYER

Dir. of the Acad. for AI and Chair Professor, Hong Kong Polytechnic University

EDUCATION

Ph.D. CS, Univ. of Maryland; B.S. Astrophysics, Peking Univ.

AS AN UNDERGRADUATE astrophysics student at Peking University, I met Nobel Prize-winning physicist Professor Charles Townes, who shared a childhood story that shaped his career: After catching an unusual fish, he sent it off to the Smithsonian Museum and was congratulated for discovering a new species. That moment cemented his path as a scientist; he went on to invent the maser and subsequently, the laser.

When I began my physics graduate studies in 1982 at the University of Maryland, I found my own “new fish”—artificial intelligence (AI). Like astrophysics, AI fascinated me with its blend of observation and model-building, but instead of studying the cosmos, it explored the human mind. After acquiring the necessary computer science (CS) background,


I switched to an AI Ph.D. program under Professor Dana Nau after completing my astrophysics master’s program.

My first faculty appointment, at the University of Waterloo, centered on “automated planning.” I have long believed the ability to engage in long-term thinking and planning is a true hallmark of intelligence—what is now referred to as the “System 2” component of human cognition. Using the computational power of state-of-the-art workstations available at the time, my students and I developed AbTweak, one of the earliest open source hierarchical AI planners. With AbTweak, we sought to explore how intelligently structuring domain knowledge—via logical frameworks and machine learning (ML)—could enable more efficient and intelligent planning systems. But I soon realized that AI planning suffered from a lack of quality data. To address this, I explored AI planning domain-knowledge learning—even taking woodworking classes to model human carpentry!


My work in “transfer learning” was partly inspired by my infant son, Andrew, who effortlessly applied knowledge across different tasks with minimal training. This reminded me of heat transfer in physics—could we develop the “thermodynamics of transfer learning”? My team explored key questions: What is the “invariant core” enabling knowledge transfer? Can AI achieve “lifelong learning” like humans? How can knowledge transfer work across “heterogeneous domains”? Is a picture truly worth a thousand words? (How can we transfer between images and text?)

One of the accomplishments I am most proud of is building the FATE federated learning system and establishing the world’s first industry standard

for federated learning.^a This work was not only technically ambitious but also deeply meaningful: It turned a theoretical idea into real-world infrastructure that protects privacy while enabling AI to scale across industries. In 2018, I joined a digital bank to start an AI project, when I realized that the digital finance sector faced an acute paradox: AI models needed massive data to perform well, yet data could not be centralized or shared due to strict privacy regulations and ethical concerns. Data was locked in silos, and traditional ML simply could not operate. Our solution was to advance federated learning, an ML paradigm that enables AI models to be built over several distributed data silos where data privacy is strictly protected. Our challenge was turning federated learning from a research concept into a stable, industrial-grade system. We had to design algorithms that could work across hundreds of heterogeneous data sources with vastly different distributions. We had to ensure the system was secure, efficient, and scalable enough for real financial transactions. Most difficult of all, we had to unify industry expectations and drive consensus toward a global standard, even while the technology was still evolving.

Like Townes’ fish discovery, a scientific career is a journey of transfer learning (applying core skills across domains) and federated learning (collaborating with others). The constants? Curiosity, patience, teamwork, and perseverance—the fuel for overcoming challenges and achieving meaningful progress. 

^a <https://github.com/FederatedAI/FATE>

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

In each issue of *Communications*, we publish selected posts or excerpts from the many blogs on our website. The views expressed by bloggers are their own and not necessarily held by *Communications* or the Association for Computing Machinery.

Read more blogs and join the discussion at <https://cacm.acm.org/blog>.

<https://cacm.acm.org/blog>

Teaching Programmers a Survival Mindset

Oscar M. Bonastre looks at trends shaping the teaching of programming.



OSCAR M. BONASTRE
We Need to Teach the 'Inuit' Mindset to Young Computing Engineers
DOI:10.1145/3793660
<https://bit.ly/49yxZts>

Two key trends are shaping the teaching of programming to the next generation of computing engineers at present. The first is the proliferation of artificial intelligence (AI) tools capable of automatically generating code; the second is the availability of ever more powerful computational resources. This combination has created a paradoxical situation: programming has never been more accessible thanks to AI, yet it's also never been easier to neglect programming efficiency and fundamental principles given the seemingly "free" nature of abundant hardware resources.

Given this situation, it's pertinent to pose the following questions to the computing science community:

► For educators: Are we adequately training young programmers and future computing engineers?

► For student developers: Is a proposed solution the most computationally efficient given available resources, and is it resilient to changing or adverse conditions?

There are several dimensions of debate on these issues, from the need to reconfigure the role of AI in programming education, to how we might reintroduce a greater focus on the fundamental principles of computing. Similarly, to emphasize the importance of reflection, I will draw on two contexts: one anthropological and one technological.

Excess Resources and Blind Faith in AI: A Perfect Storm

In my distributed systems classes, I often see students using AI-powered code assistants that suggest code fragments or even entire programs. Additionally, large-capacity memory, mass-storage devices, and fast processors are now commonplace on students' modern personal computers.

However, this technological convenience is fostering an increasing dependence on AI to solve programming tasks. It can leave students with only a superficial understanding of the algorithms being generated. This automation, along with powerful development environments, has contributed to reducing the perceived need to thoroughly understand what is really happening behind the scenes of the code being used.

Becoming accustomed to over-provisioned resources has brought further concerns. The decreasing cost of hardware encourages a certain complacency: If a code is inefficient in memory or CPU usage, one tends to trust that a more powerful machine or extra memory will solve the problem. For example, low-quality code might run acceptably with small data on a powerful machine, but fail spectacularly with larger data volumes on more constrained systems.

This mindset contrasts with the traditional discipline of programming education, in which every instruction and every byte mattered, and optimization was an essential part of the computer science student's training. The point here is that even while leveraging the benefits offered by AI in programming, an excessive dependence on AI-generated solutions and the over-provisioning of resources can undermine the proper development of computational, logical, and algorithmic thinking in future programmers or computing scientists.

To address this scenario, it is crucial to reflect on how to strengthen the teaching of computational efficiency and the complexity of algorithms implemented in modern programming languages and, above all, how to code

mathematical instructions for dynamic systems. Doing so will help students better understand sensitivity to initial conditions and cope with uncertainty in their programs' results. Reflecting on how to achieve this is essential for students to develop an adequate capacity for critically analyzing code—whether their own or that suggested by AI tools—when faced with problems with limited resources and under conditions that can unexpectedly become complicated. Otherwise, programming education runs the risk of producing future developers who have not truly, or at least not sufficiently, encountered real-world environments—simply because *AI was trusted to do it and the computer ran it without issue.*

It is important to clarify that this is not about rejecting the use of AI and reverting to a former era of computing. Instead, we should integrate the best of both worlds. We must harness the tremendous potential of AI while instilling in students the ability to evaluate and improve solutions using their own sound judgment. As a direct consequence, a well-trained programmer will think twice before accepting an AI-generated solution if it uses resources disproportionately or does not guarantee adequate resilience when execution scenarios change drastically.

Harnessing Inuit Culture for Technological Advancement

To encourage reflection, I find it instructive to draw on two approaches: one anthropological and one technological. The anthropological approach uses Inuit culture as an analogy to inspire how we teach programming. The technological approach involves developing software with AI to address a major technological challenge of the coming decades: humanity's planned arrival on Mars.

The Inuit are an indigenous people of the Arctic regions who have survived for generations in extremely hostile environments with scarce resources. How have they managed this? Through a culture of absolute resource optimization and extremely resilient adaptation to their environment. For the Inuit, every available resource is used to the maximum: Seal fat is used as fuel and food, whale bones serve to build tools and shelters, and ani-

We must harness the tremendous potential of AI while instilling in students the ability to evaluate and improve solutions using their own sound judgment.

mal skins provide essential warmth. Nothing is wasted, because survival in a hostile environment like the Arctic depends on the efficient use of every resource.

In the same way that the Inuit plan everything to survive the long polar winters, a good software developer must carefully plan resource usage—of CPU, memory, energy, storage, and so on—knowing there will be no easy reboots or reloads when conditions change or become chaotic. In other words, just as the Inuit have learned that improvisation is punished in the Arctic, in a hostile environment like Mars any waste of resources or unforeseen errors can mean the failure of software developed to recalculate an escape trajectory or to provide protection against a Martian dust storm.

For a better understanding of using AI in the development of software with interplanetary resilience in mind, I propose a metaphor from the film *Interstellar*. In the movie, the robot TARS is an extra crew member, an intelligent co-pilot that assists the astronauts in multiple tasks. This collaborative relationship illustrates how to establish a calibrated trust in AI: by taking advantage of its programming suggestions when appropriate while also leveraging its powerful capability to implement complex principles such as machine learning, neural networks, cryptography and chaotic dynamic systems to analyze and anticipate unstable behavior (for example, sudden variations or unforeseen interactions between different subsystems). Such capabilities will help the team respond

in time to highly sensitive events, preparing appropriate responses while always being ready to intervene and take back control if needed. This collaborative approach ensures that AI acts as a co-pilot that extends the programmer's abilities, rather than as an isolated autopilot.

Incorporating a computational “Inuit” survival mindset into the training of young programmers means teaching them to write code that, while preserving the ability to use AI appropriately, metaphorically “uses every part” of available resources wisely, never leaving anything underutilized, and being resilient to sudden changes in operating conditions. This mindset will help ensure a solid understanding of programming, enabling them to develop applications with a deep knowledge of how they function, much like the engineers of the Apollo missions who developed software to make humanity's arrival on the Moon a reality. This perspective not only improves efficiency, but also fosters creativity and resilience, attributes that are indispensable in programming.

Strengthening Education, Driving Achievement

All aspects discussed converge on the following scenario: The computing science students who today sit in front of a computer, surrounded by technological abundance and aided by AI, could become the future software developers who enable humanity to reach Mars and manage the Earth-Mars interplanetary Internet. To achieve this, I believe we need to reflect on how to train young “Inuit” computing engineers: resilient and capable of optimizing every resource while also being savvy in the judicious use of AI.

To conclude, prioritizing community involvement generally helps to achieve solutions beneficial to society. My aim has been to open a debate on how to strengthen and/or refocus programming education to drive the achievement of the technological challenges in the coming decades. □

Oscar M. Bonastre, associate professor of distributed systems at Miguel Hernández University (Spain), specializes in the intersection of computing, mathematics, and telecommunications. He holds a master's degree in Computer Engineering.

© 2026 Copyright held by the owner/author(s).

How Avatars Make People Feel

Studies examine the effects of avatars on their users and others.

DURING THE COVID pandemic in mid-2021, when people had been social distancing for more than a year and most university classes were being conducted over Zoom, Jeremy Bailensen decided to hold his classes in a virtual reality (VR) space.

Bailensen, director of the Virtual Human Interaction Lab at Stanford University, mailed VR headsets to his students. Sitting in their homes, 120 people entered a virtual room depicted as being about 25m², tightly packing the avatars that represented the students. The sensation of being so near others for the first time in over a year was overwhelming, to the point that some people started crying, Bailensen said. “The avatar felt real enough that you felt close enough to someone, but obviously there’s no risk of germs,” he said. “It was really one of the most special classes I’ve ever had.”

Avatars, those depictions of real people used in VR settings, online videogames such as Fortnite, and online training and marketing, have long been known to affect people psychologically. One well-studied phenomenon is the Proteus Effect,



in which users take on some of the characteristics of their avatars; if the avatar looks strong or smart, the user behaves accordingly. There have been three recent meta-analyses that support the Proteus Effect, Bailensen said.

Beyond the Proteus Effect, the avatars of other people also can trigger

emotional responses in the people who interact with them. The question of the societal and psychological effects of avatars is becoming more important, as the technology to create them improves and their use spreads beyond the confines of video games to areas as diverse as marketing and psychotherapy. Some companies, for

instance, use avatars to narrate their training, informational, or marketing videos, and to quickly translate those materials into other languages using artificial intelligence (AI). Content creators can insert avatars of themselves into social media posts, with services such as Canva offering the ability to create what it calls a “digital alter ego.” Companies such as Meta and Vive, which sell virtual reality equipment, are promoting the concept of meetings in which employee avatars are able to interact and collaborate in a virtual space. In Japan, the convenience store chain Lawson has introduced avatar-based customer service to deal with worker shortages, by allowing a remote employee to help customers in multiple stores.

Masahiko Haruno, who runs the Computational Social Neuroscience Group at the Center for Information and Neural Networks in Japan, investigated how avatars could affect people. He set up a test where people could play a game and win a small amount of money, and used functional magnetic resonance imaging (fMRI) to see what their brains were doing as they played.

The players were asked to choose between gambling on winning some amount of money, perhaps ¥300 (roughly U.S.\$2), or a guaranteed payout of a smaller amount, perhaps ¥80 (about U.S.\$0.50), with the choice flashed on a screen for three seconds. They also were briefly shown an image of a human observer, who they were told would take notes on their behavior, or an avatar of such an observer. Before the choice, the expression of the observer or avatar was neutral, and afterwards it was approving or disapproving, depending on whether they’d won or lost.

Haruno found that about 80% of test subjects were more likely to gamble if the image was an avatar, but they played more conservatively if the image was that of an actual person. The difference, Haruno said, came from how the test subjects felt about the feedback they would receive from the observer. Before the results were revealed, players were uncertain about the feedback. The fMRI showed that this uncertainty mainly showed up in the amygdala, the part of the

About 80% of test subjects were more likely to gamble if the image observing them was an avatar, but they played more conservatively if the image was an actual person.

brain that processes emotional responses, including fear and anxiety.

When the observer was a human face, uncertainty about the feedback triggered a stronger response in the amygdala than it did when the observer was an avatar. In other words, the risk of receiving a negative response felt more threatening from an actual person. Risky behavior felt safer with an avatar.

Haruno and his colleagues also administered a questionnaire to assess the personality traits of their subjects. They found that people who scored higher on measures of empathy were more likely to take a risk with an avatar than they were with a person. “If you have a strong tendency to care about the other person’s internal state, then the effect of the avatar to increase the gambling option was larger,” Haruno said.

The results, he said, raise the possibility that avatars could be used to manipulate people. Perhaps avatars might encourage people to make riskier choices when buying stocks, for instance. “It’s not surprising. If you think that the appearance of other people changes your behavior, it’s basically the same thing,” he said.

There is evidence, Bailensen said, that avatars in online therapy may provide advantages over videoconferencing. Avatars encourage people to behave more like they would in the presence of others, but still make the interaction feel artificial enough to be less intimidating than in real life.

“Having an avatar, there’s social presence and I want to behave in a real way, but I’m more likely to disclose things about myself and feel less judged than by a therapist who’s on a video conference,” he said.

Doctors are looking at how avatars can help to treat people with post-traumatic stress disorder, by allowing them to engage in role play in a safe, virtual environment. Some therapists are using avatars to embody the voices heard by people with psychosis, allowing the patients to challenge and gain control over the voices.

How to Choose

Some scientists are looking into how people choose avatars to represent themselves. Researchers in Stuttgart, Germany, for instance, found people preferred more realistic, less cartoon-like avatars when they were using them in a professional context.

Meanwhile, Katrina Fong and Raymond Mar of York University in Canada explored whether people choose avatars in video games to fulfill some psychological needs. In a study, they found that people who were measured by standard tests as having a high need for personal warmth were more likely to select avatars they perceived as being warm and friendly. Subjects also were more likely to prefer avatars perceived as being competent, although that did not appear to correlate with a psychological need for competence.

Scientists at the University of Washington found a divergence in how people felt about privacy and identifying characteristics in avatars. Computer science and engineering professor Tadayoshi Kohno and his colleagues found that people who felt themselves marginalized based on aspects of their identities, such as race, religion, or sexual orientation, had less desire to reveal those identities in an avatar than did members of non-marginalized groups. On the other hand, study participants were uncomfortable with the idea that other people were hiding identifying characteristics in their avatars, feeling that such concealment might compromise their own online safety. As a result, members of marginalized groups who hid aspects of their iden-

tities to protect themselves might face backlash for doing so, the researchers said.

Real vs. Not Real

Jill Schiefelbein runs The Dynamic Communicator, a consulting business that helps companies use AI to create marketing and training videos, sometimes relying on avatars. Her thesis for her Ph.D. in business administration and management from the University of South Florida in 2023 compared the use of avatars versus real people in training videos. She gave her test subjects one of three conditions: a two-minute video of herself reading a script about sales communication, a video of a photorealistic avatar of herself reading the same script, or a video of the avatar that was identical except that it disclosed at the outset that this was an avatar. There was no difference between the human and either avatar in terms of how much of the information people learned.

The only difference, she found, was that people who watched the avatar that identified itself did less well on a quiz about the information provided in the video. Using gaze-tracking software, she noticed that people in that group were moving their eyes around the screen a lot more than the other two. She speculates that, because such avatars were novel at the time, the people were distracted looking for clues to the fakery.

The one major difference she found was that, while people had rated the experience as positive, when they found out after the fact that they'd been watching an avatar, their sentiments swung sharply to the negative. "Did those people score well on the tests? Yes, they did," Schiefelbein said. "But how did they feel about it afterwards? Really, really poor." She said avatars can be a useful communication tool, as long as people do not feel that they have been duped by their use.

While avatars have been in use in video games for 20 years or so, it is only recently that they have started to be used for interpersonal communication, so the effect of that has not been studied very much, said David Beyea, who studies computer-mediat-

While people had rated the experience as positive, when they found out after the fact that they'd been watching an avatar, their sentiments swung sharply to the negative.

ed communication at the University of Wisconsin, Whitewater. "People like to talk about the metaverse a lot. We don't know what that's going to look like. We're hypothesizing based on sci-fi novels and what little research we have right now, and that's going to all shift," he said.


One of the issues, he said, is how well avatars reproduce the non-verbal cues that people rely on to tell how people are reacting to them, from facial expressions to how close they stand. Beyea and his colleagues are working on developing an artificial intelligence algorithm that can provide non-verbal cues through avatars to make communications smoother.

A difficult area is facial expressions, but that may be changing. Apple's Vision Pro VR headset uses 3D scanning, eye-tracking cameras, and machine learning to create credible renditions of faces and hand gestures. The company donated 15 headsets to allow Bailensen to study the avatars they produce.

Beyea said we may be only months away from the use of more AI-driven avatars in virtual environments, such as a virtual assistant that can run online meetings or interrupt a discussion to offer information. Those may face the issue of the so-called 'uncanny valley of the mind.' The uncanny valley is the effect where a rendering of a human appears mostly real but is not quite fully realistic, causing people to feel uneasy. That can happen if photorealistic avatars fail to fully match human movements. The un-

canny valley of the mind arises when the avatar's behavior is slightly off, suggesting that the brain behind it might not be human. Beyea is hoping to study the issue with an AI-driven avatar teaching assistant he has built.

In the long term, Beyea is not worried about avatars having ill effects on society. "We're always going to have a moral panic," he said. "Everybody gets very, 'Oh my god, it's going to destroy society or is going to manipulate us.' And what we always tend to find is that the younger generation, that are more experienced because they grew up with that technology, aren't as affected as much as people to whom it's a novel thing."

Schiefelbein agrees. "Every communicative technology since the invention of the chisel has been used for good and has been used for ill," she said. "This technology is no different." 

Further Reading

Beyea, D., Ratan, R., Lei, Y., Liu, H., Hales, G.E., and Lim, C.

A new meta-analysis of the Proteus effect: Studies in VR find stronger effect sizes. *PRESENCE: Virtual and Augmented Reality* 31 (2022); 10.1162/pres_a_00392

DeVeaux, C., Han, E., Landay, J., and Bailenson, J.

Exploring the relationship between attribute discrepancy and avatar embodiment in immersive social virtual reality. *Cyberpsychology, Behavior, and Social Networking* (2023); 10.1089/cyber.2023.0210

The proteus effect on human pain perception through avatar muscularity and gender factors. Fumihide Tanaka Laboratory (2024); <https://www.youtube.com/watch?v=KhMBgO4HIb0>

Schiefelbein, J.F.

Human vs. machine: Hyper-realistic avatars and their efficacy as a communication channel. University of South Florida (2023); <https://digitalcommons.usf.edu/etd/10088/>

Tanaka, T. and Haruno, M.

Feedback from an avatar facilitates risk-taking by modulating the amygdala response to feedback uncertainty. *PLOS Biology* (2025); 10.1371/journal.pbio.3003122

Neil Savage is a science and technology writer based in Lowell, MA, USA.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 ACM 0001-0782/26/4

AI Multiplies for Drug Discovery

Throttling access to a leading AI tool for drug discovery has spawned many variants and advances. But there is still some way to go.

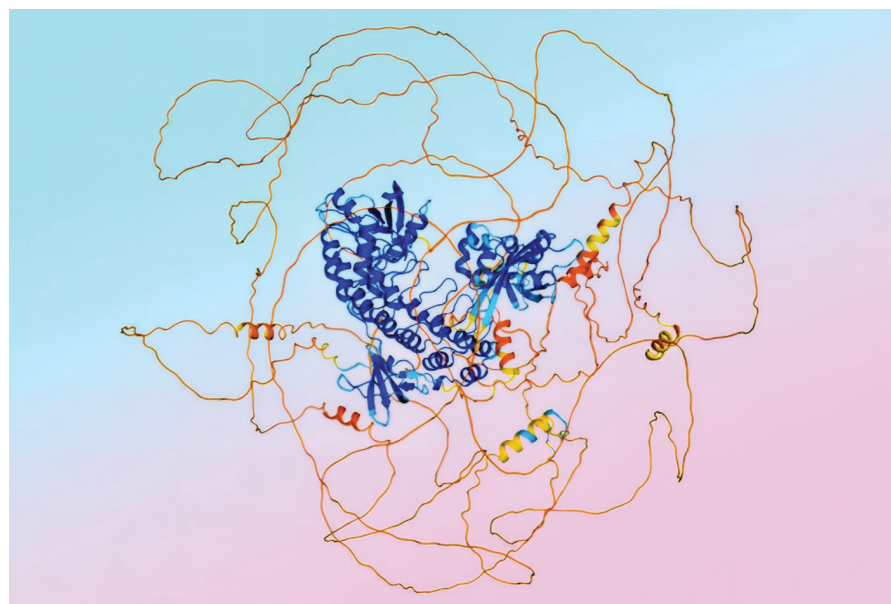
THE IDEAS BEHIND DeepMind's AlphaFold won its inventors a Nobel Prize for chemistry just a few years after the technology first appeared. However, a decision taken by the company earlier in 2024 underlined how researchers in sectors like drug discovery see the importance of the tool by triggering the creation of multiple workalikes based on the same principles.

DeepMind decided to restrict access to the third version of AlphaFold, forbidding the use of the publicly accessible version of the software for drug discovery. The paper the development team wrote for the journal *Nature* describing the internal behavior of the revamped software failed to include the source code. That was a break from previous releases.

Days after AlphaFold3's launch, 1,500 scientists signed an open letter criticizing the company's actions. The letter called for open science and for tools that would allow a greater level of inter-lab collaboration. And their wish has been granted, though only partly by DeepMind.

The Alphabet subsidiary subsequently released some of the source code under a Creative Commons license, with limits on its use. Other groups quickly moved to clear the obstructions that DeepMind's restrictions on AlphaFold3 presented to their own work to make sense of how proteins interact with drugs and other materials.

Proteins are the workhorses of the body. Among many other tasks, they break down foods into fuels that muscles, also made of proteins, use to move. Yet cancers and viruses co-opt them. Viruses hijack them to make copies. Cancers develop when a fault in cells' internal signaling causes them to grow out of control. Blocking one or more of those co-opted proteins can stop a vi-



The AlphaFold 3 protein-folding structure.

rus from replicating; or, in the case of a cancer, trigger a signal to the cell's own self-destruct mechanism. That kind of targeted inactivation or inhibition is key to the action of many of the drugs that make it through the lengthy drug-development pipeline.

Medicinal chemists search for molecules that will settle neatly into an active site on a target protein and

Medicinal chemists search for molecules that will settle neatly into an active site on a target protein and block its normal actions.

block its normal actions. One of the most important puzzles is what that active site looks like. That helps determine what the matching shape should be for the ligand, the active part of a drug molecule. This is a process that tools like AlphaFold are helping to speed up.

Proteins are incredibly complex structures. Long chains of amino acids can fold into complex structures, each evolved over countless years to fit a task. This process proved tough to model using conventional tools based on physical principles, but machine learning made it possible to take advantage of information in existing structures, collected in the public Protein Data Bank (PDB), coupled with hints from biological evolution.

The Evoformer module that has sat inside AlphaFold since its beginning compares the sequences of proteins that differ across species, but with broadly similar structures. The subtle differences in shape between similar amino-acid sequences provide the AI

model with hints on how proteins not in the PDB will fold.

AlphaFold3 made two further changes. One was the adoption of a technique that underpins many of the image and video generators now in widespread use. The same de-noising algorithms take the coarse structure generated by the Evoformer that renders into a more accurate characterization of the target protein. AlphaFold3 adopted another innovation from other experiments on protein modeling. This helps with drug discovery specifically.

Various tools have taken on the task of “docking” drug candidates with a protein structure. However, most do not consider the many ways that a protein can change shape to accommodate a ligand. AlphaFold3 handles this by predicting both together in a process called co-folding that has proved more accurate than AI-based docking tools, including one called DiffDock developed at the Massachusetts Institute of Technology (MIT).

“DiffDock was adopted in a lot of pharmaceutical companies, biotech companies, and that gave us an idea

both about the importance of this task, and also about the way in which some of these models were used,” said MIT postdoctoral researcher Gabriele Corso.

Because of the tightened restrictions on AlphaFold3, the team saw they would need to have control over protein-structure prediction to do further work with co-folding. Five months later, they released a model capable of the task, named Boltz-1, as open source. Other teams embarked on similar projects, leading to a proliferation of both open source and proprietary tools. In the space of a single week at the end of October 2025, three groups, including the Boltz team at MIT, launched different variations on AI for generating structures of protein complexes. But they have broadly similar performance on unfamiliar combinations of proteins and ligands.

If the models could generalize atom-level interactions rather than memorizing the shapes they see, the accuracy of novel combinations should be good. To test that assumption, doctoral student Peter Škrinjar, together with colleagues at Switzer-

land’s University of Basel, developed a benchmark called Runs N’ Poses. The results on existing AI models are like the pre-AlphaFold situation for single proteins: those models showed a steep decline in accuracy when they tried to predict structures not in the training set. That includes situations where the target protein is well-represented in the databases but the ligand molecule is not. That problem is exacerbated by the fact that the diversity of small molecules is far higher than that of proteins in biology.

The question is what will lead to a breakthrough in generalization. The solution most teams are following in the near term is to find more data. After releasing a preview of its OpenFold3 model in October 2025, the Open Molecular Software Foundation (OMSF) said it plans to boost the size of its training set massively through two initiatives.

One is a collaboration with the Diamond Light Source laboratory near Oxford, U.K. Funded as part of the AI for Science program launched by the British government in late 2025, OpenBind is one of several attempts to speed up

Multi-LLM Agent Collaborative Intelligence

The Path to AGI

Edward Y. Chang

ISBN: 979-8-4007-3178-5

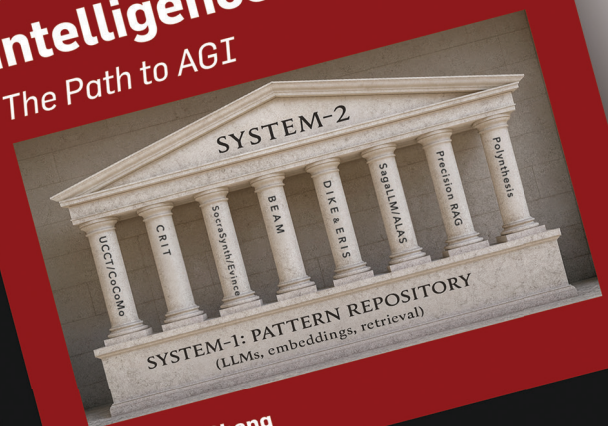
DOI: 10.1145/3749421

<http://books.acm.org>

 **ACM BOOKS**
Collection III

Multi-LLM Agent Collaborative Intelligence

The Path to AGI



Edward Y. Chang



ASSOCIATION FOR COMPUTING MACHINERY

the production of experimental structure for the PDB. Alongside electron microscopy, the X-ray crystallography that Diamond performs has proved to be a valuable source of protein structures. Until recently, these measurements have required painstaking manual effort. But automation, some of it assisted by machine learning, will speed up the production of new structural models.

Over the next five years, the group aims to create more than 500,000 individual protein-ligand structures. That will build a dataset that the group expects will be 20 times larger than anything collected over the past 50 years.

A second strand of work will use data from the private archives of pharmaceutical companies, taking advantage of federated learning to do so. Specialist provider Apheris is constructing a system that will run training software on these companies' own machines so that they do not need to expose any of their proprietary data to the outside world.

Another source of data can come from simulations that use physical models of ligands and the parts of the protein closest to them. Genesis Molecular AI used public datasets compiled from these simulations to augment the data learned by its Pearl model, which was unveiled the same day as OpenFold3. According to Aleksandra Faust, chief AI officer, the use of synthetic data provided Genesis with an opportunity to analyze the effects of data scaling on performance.

"We are the first to provide clear evidence of synthetic-data scaling laws for co-folding," Faust said. The success rate on structures held out from the training set increased as more of the synthetic data was fed into successive iterations of the training process. Results from running Runs N'Poses on their model's predictions showed a steep drop-off below 40% similarity to the training data, though the synthetic data seemed to improve performance over the range of targets.

"Our findings are consistent with prior work in robotics and large language models, which suggests synthetic data must be diverse and maintain a sufficient distributional distance to the original training data," Faust added.

An opportunity to gauge the effects

The question is what will lead to a breakthrough in generalization. The solution most teams are following in the near term is to find more data.

of increased real-world data came with Boltz-2, which MIT developed together with pharmaceutical company Recursion. Released in the summer of 2025, this model benefited from 25,000 more structures in the PDB than its predecessor. But it did not see a significant improvement in generalization, according to Škrinjar.

"More experimental data and better balance across target classes might help with the performance. But I believe the main limitation remains architectural. Co-folding methods do not yet capture the deeper geometric and physical principles of protein-ligand binding," Škrinjar said.

The main shortcoming is that shape only goes so far in determining how well a drug candidate will bind to its target. The species of atom in the molecule directly affects the local density of the cloud of electrons around the molecule. Oxygen, for example, pulls electrons from other atoms close to it. The oxygen atom receives a small net negative charge while neighboring atoms become slightly positive. If a molecule can line up with compatible electron sinks and donors on the surface of a protein, the binding energy increases to the point where it is very unlikely it will peel away. That strength of interaction points to a molecule that will probably work well as an inhibitor.

Some models have added some of these inductive biases to improve the quality of predictions. According to Škrinjar, results with a variant of Boltz-1, showed the benefit of introducing post-prediction physics-based modeling. However, the addition of

a physics module did not seem to improve the AI's ability to generalize to previously unseen targets. The module needs a relatively good initial pose. For now, they cannot correct large mistakes made by the diffusion engine.

Further improvements may come from incorporating more physical interactions into the prediction engines. Research groups working outside the AlphaFold paradigm have used graph neural networks and similar structures to try to incorporate physics-based interactions into models more directly. These models are better at simulating the changes in shape proteins and ligands go through as their environment changes. But such models suffer from the same problem as pre-AlphaFold tools when it comes to predicting the larger-scale structure of proteins.

In practice, researchers in drug discovery may choose to use diffusion models to build initial structures and then move into the physics-oriented models as they home in on the details of protein-ligand interactions. Integration may prove challenging, but hybrid architectures built into AlphaFold-like engines may prove to be the path to more streamlined and accurate methods for drug discovery. **C**

Further Reading

Dobles, A. et al.

Pearl: A foundation model for placing every atom in the right location. *arXiv* (2025); 2510.24670

Masters, R.M., Mahmoud, A.H., and Lill, M.A.

Investigating whether deep learning models for co-folding learn the physics of protein-ligand interactions. *Nature Communications* 16, 8854 (2025).

Škrinjar, P., Eberhardt, J.,

Durairaj, J., and Schwede, T.

Have protein-ligand co-folding methods moved beyond memorization? *bioRxiv* (2025); 02.03.636309

Wohlwend, J., Corso, G., Passaro, S.,

Getz, N., Reveiz, M., Leidal, K., Swiderski, W., Atkinson, L., Portnoi, T., Chinn, I., Silterra, J., Jaakkola, T., and Barzilay, R.

Boltz-1 democratizing biomolecular interaction modeling. *bioRxiv* (2024); 11.19.624167

Chris Edwards is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 ACM 0001-0782/26/4

In Memoriam: David J. Farber

“Grandfather of the Internet” who excelled in academic and industry leadership roles made networking his life.

NETWORKING WAS DAVID J. Farber’s life’s work. Farber helped design the first electronic switching system (ESS-1) during his 11-year career at Bell Laboratories (1956–1967). Later, as an academic who held faculty appointments at five universities, the moderator of the highly influential “Interesting People” mailing list that chronicled the intersection of technology and policy on the early Internet, and the chief technologist of the U.S. Federal Communications Commission (FCC) from 2000–2001, he specialized in networking people.

Farber died suddenly at his home in Roppongi, Tokyo, Japan, on Feb. 7, 2026, at the age of 91.

Farber graduated from the Stevens Institute of Technology in Hoboken, NJ, USA, with a bachelor’s degree in electrical engineering in 1956 and joined the technical staff of the Electronic Switching Engineering Systems Department at Bell Telephone Laboratories, where he worked on the ESS-1 team. His particular emphasis was on software structure and software support infrastructure.

In 1961, Farber completed a master’s in math from Stevens; in 1963, he completed the Communication Development Program at Bell Labs, which he noted was equivalent to a master’s degree in electrical engineering. In 1965, he was promoted to supervisor, responsible for the Holmdel Computer Center, a key player in the design and implementation of the Multics operating system.

Farber joined the technical staff of the RAND Corporation in 1967 to work on Paul Baran’s vision of communications systems that could survive a nuclear war. This project gave birth to packet networking and the Internet. He joined the technical staff of Scientific Data Systems (SDS) in 1969 to continue the work, while simultaneously teaching at the University of California, Irvine, in the evenings.

Xerox acquired SDS in 1969. Unhappy

with the company’s stewardship, Farber decided to go full-time at UCI.

“Because of my industrial experience and the fact that, pragmatically, I never got around to getting a doctorate, I had an acting associate professor appointment, so I had two years to do something,” Farber told an interviewer in his 1988 oral history.⁴

Farber submitted a proposal to the NSF to build a decentralized computer system with three minicomputers connected via a token-ring local area network; it was funded at \$250,000 per year (\$2.1M today), securing Farber his first tenured faculty position. Among his Ph.D. students were Internet pioneers Jonathan Postel (creator of the Internet Assigned Numbers Authority), Marshall Rose (inventor of the Simple Network Management Protocol), and Paul Mockapetris (inventor of the Domain Name System).

In 1975, DARPA directed MIT’s Laboratory for Computer Science to adopt Farber’s token ring technology as the basis for its development. Ultimately, LCS adopted Ethernet, an alternative technology, but IBM adopted Token Ring and used it throughout the 1990s.

In 1977, Farber joined the University of Delaware as a professor of electrical engineering and professor of computer science. There, he led the development of the Multichannel Memorandum Distribution Facility (MMDF), an email system widely used on Unix systems for nearly three decades.

Farber helped design the first electronic switching system (ESS-1) during his 11-year career at Bell Laboratories.

In 1985, the National Science Foundation created NSFNET, a civilian IP-based network to link U.S. universities that had previously been connected only by ARPANET and non-IP networks such as BITNET and the store-and-forward UUCP network. NSF’s director, Erich Bloch told Farber he was having trouble keeping up with all the news; Farber responded by forwarding to Bloch timely, on-topic email messages. Soon, Bloch suggested adding a few of his colleagues at IBM as recipients of the messages. As the recipient list grew, Farber named his list “Interesting People.” At its peak, the list had more than 25,000 recipients. The online archives, with 37,154 messages from May 1993 to February 2019, offered a curated view of the growth of the commercial Internet and the World Wide Web, as well as the range of technical, political, and social issues that resulted.

“Supercomputing is on the verge of becoming a utility,” Farber told John Markoff of *The New York Times* in 1992.³ “You may want more computing power for a very short period of time. In the future, without even asking, your computer will reach out over high-speed computing networks to get that power.” We live in that world today, although for most people, the computer that seamlessly integrates with the cloud-based supercomputing utilities is the smartphone in their hands.

“It was David who first alerted me to the societal significance of computer networks and to distributed computing more generally,” recalls Markoff, who wrote for the paper from 1988 until 2016. “As I became more interested in high-performance computing and high-speed networks, David was one of my most valued sources.”

Farber moved to the University of Pennsylvania in 1988, where he worked on ultra-high-speed networking, examining the relationship between processor interconnect, protocols, and software. He eventually added appointments as professor of business and pub-



lic policy at the Wharton School of Business and as a faculty associate of the Annenberg School of Communications.

“Perhaps the single most important thing he taught me, that I have never forgotten, is to let your graduate students be independent, let them explore widely, never stifle their creativity and they’ll become truly capable researchers,” recalls University of Pennsylvania professor Jonathan M. Smith, who started at Penn as an assistant professor in 1989.

“With respect to the Gigabit Testbeds, Dave discovered that researchers at Bellcore Bell Communications Research ... had been working on (and had succeeded at building) a novel data switch fabric that could interconnect 100Mb/s Asynchronous Transfer Mode links,” wrote Smith in an email. At the time, most computer networks ran at 10Mbps. “Dave quickly grasped the implications” of the work at Bellcore, and arranged for people and funding from multiple sources; by 1993, the team had created the world’s first gigabit network from Penn to Bellcore in New Jersey.¹

Farber joined the Board of Trustees of his alma mater, Stevens Institute of Technology, in 1997 and was awarded an honorary doctorate by the institution in 1999. Farber took a leave of absence from UPenn in June 2000 to serve a one-year appointment as chief technologist at the FCC.

“We were on the Task Force on National Security together, and I subscribed to his labor-of-love tech news roundup,” recalls Stewart Baker, the first assistant secretary for policy at the U.S. Department of Homeland Security and former general counsel of the National Security Agency. “He had the usual views of academia on political and tech issues but was happy to talk to people with very different views. He was endlessly open to new information and a pleasure to know.”

Farber retired from UPenn in 2003 and immediately joined Carnegie Mellon University as a Distinguished Career Professor of Computer Science and Public Policy in the School of Computer Science, with a secondary appointment in the Heinz School of Public Policy and the Engineering Public Policy Group.

Over the course of his career, Farber became more concerned about the use of technology to control individuals, the fragility of the Internet he helped create,

Farber became more concerned about the use of technology to control individuals, the fragility of the Internet, and the impact of increasingly powerful technologies.

and the impact of increasingly powerful technologies on individuals and society at large.

Author Cory Doctorow recalls that after Farber left the FCC, the Interesting People mailing list was used to organize opposition to the Broadcast Flag, a proposal that would have imposed restrictions on computers for handling copyrighted video. Doctorow had a “strategy call” with Farber, in which Farber recommended: “Don’t call the people you’re standing up for ‘users’ or ‘consumers.’ Call them ‘computer owners’ and ‘citizens.’ Make sure the person you’re talking to understands that this is about your right, as a member of society, to control how your own computer works, and whether it serves you.”

In 2006, Farber was one of 17 computer scientists to sign an *Amici Curiae* brief to the U.S. Supreme Court in the case of *MGM Studios vs. Grokster*.⁵ In 2016, Farber said that the Internet must be secure but that “it was never built to be securable ... We’re sitting here with critical resources on top of systems we can’t protect ... And I think it’s our duty as people who want a healthy Internet to start paying attention to that.”²

Farber had a deep love of Japan and had visited the country dozens of times. Following the death of his wife of 45 years, Gloria (G.G.), he moved to Japan in 2018 to become a Distinguished Professor at Keio University.

“Dave’s relationship with Japan began in 1973, when Keio’s Professor Hideo Aiso made an extended visit to Dave’s group at UC Irvine,” recalls Jun Murai, professor and dean of the fac-

ulty at Keio University in Tokyo, who is frequently honored as “the father of the Japanese Internet.”

“Dave’s influence on networking and distributed systems research in Japan is hard to overstate, and he was very proud to have given the first public lecture at Keio’s Shonan Fujisawa Campus when it opened in 1990,” Murai continued. In April 2018, Farber and Murai co-founded the Cyber Civilization Research Center at Keio University to study the societal, economic, and technological impacts of digital transformation.

Farber loved teaching and did so until his final class on Jan. 22, 2026. He served on the board of directors of the Electronic Frontier Foundation, the Board of Trustees of the Internet Society, the advisory boards of Electronic Privacy Information Center and the Center for Democracy and Technology, and was a member of the President’s Information Technology Advisory Committee (PITAC, formerly the Presidential Advisory Committee on High-Performance Computing and Communications, Information Technology, and the Next Generation Internet). He was a fellow of the ACM, the IEEE, and AAAS. He received the 1995 SIGCOMM Award for his lifelong contributions to computer communications, and in 1997, the John Scott Award for Contributions to Humanity.

Farber is survived by his son, Manny Farber; daughter-in-law, Mei Xu; daughter-in-law, Carol Hagan, widow of his son Joseph M. Farber; and grandsons, Nate Farber and Sam Farber.

References

1. Clark, D.D. et al. An overview of the AURORA gigabit testbed. In *Proceedings of the IEEE INFOCOM ’92: The Conf. on Computer Communications 2* (1992), 569-581; DOI 10.1109/INFCOM.1992.263450
2. Farber, D. Keynote, Internet Governance Forum, Elon University (2016); <https://www.elon.edu/u/imagining/event-coverage/igf-usa/igf-2016/farber-keynote/>
3. Markoff, J. A crucial linkup in the U.S. data highway. *The New York Times* (Sept. 30, 1992).
4. Pelkey, J.L. Interview of Dave (David) Farber. (Mar. 8, 1988); CHM Reference number X5671.2010.
5. Samuelson, P. Regulating technical design. *Commun. ACM* 49, 2 (Feb. 2006), 25-30; 10.1145/1113034.1113057

Simson Garfinkel is an ACM Fellow.

Eugene H. Spafford is a distinguished professor of computer science and the founder and executive director emeritus of the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, W. Lafayette, IN, USA. He is an ACM Fellow.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

When Seeing Isn't Believing

New research reveals how AI can alter human memories—and reshape our sense of reality.

WHEN A REDDIT user known as “ibreakphotos” first snapped a picture of the Moon with his new Samsung Galaxy S20 Ultra phone, he was startled by the result: a razor-sharp image of the lunar surface, with every crater and shadow etched in sharp relief.

There was just one problem—the picture wasn't exactly real.

“Something seemed off,” he recalled. As a hobbyist photographer with a basic grasp of optics, he found the level of detail in the image suspicious. Out of curiosity, he snapped another photo—this time of an intentionally blurred image of the moon displayed on his computer monitor. The result: another magically detailed image. He concluded there was only one possible explanation: “It's AI.”

When he posted his findings to Reddit, the results went viral. “I didn't expect it to blow up as much as it did,” he said, but his experiment seems to have struck a nerve. As AI becomes increasingly embedded in the cameras, apps, and assistants that mediate our lived experiences—retouching our photos, summarizing our conversations, and quietly editing the digital records of our lives—it is also subtly altering the contents of our memories.

The growing influence of AI in shaping human recollection is capturing the attention of psychologists and computer scientists alike. Recent experiments suggest that generative systems don't just alter the contents of digital files; they can also fundamentally change what people *think* they have seen—with implications stretching from relatively mundane use cases like vacation photos to the high-stakes realms of politics, finance, and criminal justice.

What exactly *is* memory, anyway? Philosophers from Buddha to Plato



A blurry stock photo of the Moon (inset), and how Samsung's AI clarified it.

to William James have grappled with that question. Yet in the age of AI, the age-old quest to understand how human memory works has taken on new

The growing influence of AI in shaping human recollection is capturing the attention of psychologists and computer scientists alike.

urgency, as researchers discover just how fragile—and easily manipulated—our recollections can be.

“Many people think that memory works like a recording device: you record the event and then you play it back later,” said University of California, Irvine, psychologist Elizabeth Loftus. However, the truth is far more complex. “We're often taking bits and pieces from different times and places, and we bring it together and construct the memory.”

Loftus' work has shown that people can be persuaded through suggestive interviewing—in some cases, to the point where they recall events that never actually happened.

Harvard psychologist Daniel Schacter has gone further, cataloging the seven “sins” of memory—transience, absentmindedness, blocking, misattribution, suggestibility, bias, and persistence—innate hu-

man foibles that make all of us unreliable narrators of our own lives.

In other words, memory works less like a camera than like a collage—selected, assembled, and subject to interpretation. In a world awash in AI-generated content, opportunities are continuously multiplying for bad actors to hijack the human memory-making apparatus.

Recent studies are bringing these risks into stark relief. At the Massachusetts Institute of Technology (MIT) Media Lab, Pat Pataranutaporn and colleagues discovered that AI-generated images and videos can not only blur the lines between real and fake, they can also give people extraordinary levels of *confidence* in false memories. The team showed 200 participants a set of original photos and videos, then replaced some with versions altered by generative AI. Later, participants who had been exposed to the AI-enhanced versions recalled false details with the same level of conviction as those who had seen the originals. The effect was strongest with hybrid combinations of video and images, which made participants even *more* certain of false memories compared to factually correct ones.

A separate study by Samantha Chan of Singapore's Nanyang Technological University explored how generative AI chatbots can also be used for deceptive ends. In simulated interviews, participants interacting with a chatbot programmed to add false details to a conversation were more than three times as likely to recall falsehoods as those prompted with a static questionnaire. The study underscores that the danger lies not merely in disseminating falsehoods, but in the persuasive, adaptive style of generative dialogue.

"The chatbot was intentionally designed to agree with the users," said Chan. She believes there could be opportunities to engineer chat interfaces with more critical reasoning baked in; for example, through Socratic-style questioning, or other cues reminding them that these digital artifacts have been edited. "How could we subtly enhance people's critical thinking when interacting with these systems?"

Not every experiment points towards dystopian outcomes, however. In Barcelona, design researcher Pau Garcia is leading a team exploring how AI-generated photographs might help elderly patients with dementia reconstruct and engage with their own past, in a process known as reminiscence therapy. Their early research shows promise, and they are now testing how AI might help people reconstruct memories lost through traumatic events—for example, refugees who have lost their photo albums. But any attempt to fashion such prosthetic memory aids is inevitably fraught with ethical landmines.

"There is no such thing as benign manipulation," said Marie Potel-Saville, CEO of Fairpatterns, a think tank devoted to helping companies identify and address dark patterns in AI applications. Ultimately, she believes that users should have maximal control over whether and how AI is used to manipulate information they have created—no matter how well-intentioned the project. "The ethical way is to empower users to make fair and informed choices," she said, arguing that industry and government need to maximize transparency and disclosure to ensure users understand how any given artifact was created.

These principles are starting to take hold in both industry and regulatory dialogues. In the European Union's Artificial Intelligence Act, one article specifically prohibits any form of manipulation "below the level of consciousness." Meanwhile, ma-

AI-generated images and videos not only blur the lines between real and fake, they can give people extraordinary levels of confidence in false memories.

ior AI vendors like OpenAI, Microsoft, Meta, Google, Intel, and Amazon have joined the newly formed Coalition for Content Provenance and Authority (C2PA), which is developing a specification for "Content Credentials"—a kind of digital nutrition label showing the provenance and history of any image, audio, or video file.

As these technical and regulatory frameworks take shape, however, a deeper epistemic challenge still remains. People and machines process information in fundamentally different ways. Whereas humans construct memories out of their own flawed, fragmentary recollections, computers can (at least in principle) record everything indiscriminately. The result is a collision between two very different kinds of recall: one that is subjective and highly constructed, and another that is seemingly empirical and never forgets. But what happens when those two different ways of understanding the world contradict each other—and, complicating matters even further, what if the machines themselves begin to disagree?

Petar Popovski, a communication engineering professor at Denmark's Aalborg University, has been thinking about precisely that question. He warns of "Rashomon effects" (after the famous Akira Kurosawa film), where multiple devices observing the same event might disagree about its sequence. That problem grows even thornier when humans try to reconcile their own recollections with conflicting machine accounts.

"Humans don't remember everything they have seen," Popovski said, "we just remember important details." But computers are not so selective. As human memories become intertwined with recordings from multiple cameras, sensors, and satellites—each with its own network delays and enhancement algorithms—these discrepancies pose a risk of what he calls "causality violation" or "temporal displacement": a kind of digital motion sickness.

Computer scientists have long grappled with the issue of sequencing events. In the 1970s, Leslie Lamport's seminal research showed how distributed systems could maintain consistent event sequences through

Popovski warns of “Rashomon effects,” where multiple devices observing the same event might disagree about its sequence.

synchronized timestamps. Yet today’s networks are far messier: a vast, asynchronous array of sensors, relay mechanisms, base stations, satellites, and AI filters. Even small delays can cascade through such a complex ecosystem into large distortions, altering our sense of cause and effect—and creating openings for manipulation by bad actors.

“Imagine there are multiple devices capturing an event in real time, and someone inserts a murder in there,” Popovski said. What happens, in other words, when the machines we rely on as eyewitnesses start to lose their credibility?

Pataranutaporn and his colleagues have proposed the need for “memory hygiene” practices—like watermarks, provenance tracking, and edit disclosures—to preserve context and authenticity in an AI-saturated world. And Popovski’s research on event ordering and causal integrity underscores the importance of timestamping and synchronization in defending against distortions in how events are recorded.

As generative models become woven into everyday sense-making and record-keeping activities, these safeguards will become ever more critical. Ultimately, the dividing line between reality and digital fabrication is no longer a fixed boundary, but an ever-shifting frontier.

In the classic 1966 Philip K. Dick story “We Can Remember It for You Wholesale” (the basis for the movie *Total Recall*), the protagonist Douglas Quail dreams of taking a trip to Mars, and turns to a company called ReKall,

Inc. that implants false memories—for a fee. As the story develops, he discovers that his new memories become deeply intertwined with his real ones, with maddening and ultimately devastating results.

What makes the story so unsettling is that Quail’s artificially implanted memories feel every bit as authentic to him as real ones, raising a crucial question: in an age when our recollections can be manipulated through machine learning, will human memory become just another medium—or will we rediscover the timeless truth that memory has always been, more than anything else, an act of imagination? **□**

Further Reading

Chan, S., Pataranutaporn, P., Suri, A., Zulfikar, W., Moes, P., and Loftus, E.F. Conversational AI powered by large language models amplifies false memories in witness interviews. [10.48550/arXiv.2408.04681](https://arxiv.org/abs/10.48550/arXiv.2408.04681)

Dick, P.K. “We can remember it for you wholesale.” in *Selected Stories of Philip K. Dick*. J. Lethem, Ed. Random House, New York, NY.

Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications* 21, 7 (July 1978), 558–565. [10.1145/359545.359563](https://doi.org/10.1145/359545.359563)

Pataranutaporn, P., Archiwaranguprok, C., Chan, S.W.T., Loftus, E., and Moes, P. Synthetic human memories: AI-edited images and videos can implant false memories and distort recollection. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Article 538 (2025); [10.1145/3706598.3713697](https://doi.org/10.1145/3706598.3713697)

Popovski, P. How AI’s sense of time will differ from ours. *IEEE Spectrum*, 13 (Aug. 2025). <https://spectrum.ieee.org/ai-perception-of-time>

Schacter, D.L. The seven sins of memory: Insights from psychology and cognitive neuroscience. *American Psychologist* 54, 3, 182–203. [10.1037/0003-066X.54.3.182](https://doi.org/10.1037/0003-066X.54.3.182)

Alex Wright is a writer and researcher based in Brooklyn, NY, USA.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 ACM 0001-0782/26/4

ACM Member News

USING WEARABLES FOR MOBILE HEALTH



“When I was young, I was very interested in math and also science fiction, and computer

science somehow attracted me as a combination of the two,” said Cecilia Mascolo, a professor of Mobile Systems in the Department of Computer Science and Technology at the University of Cambridge in the U.K.

Mascolo received her bachelor’s, master’s, and doctoral degrees in computer science from Italy’s University of Bologna.

She then joined the faculty of the Department of Computer Science at University College London in the U.K. She later moved to the University of Cambridge in 2008, where she has remained.

In 2025, Mascolo was elected a Royal Academy of Engineering Fellow, and she serves as vice chair of the ACM Special Interest Group on all things mobile, SIGMOBILE.

Her research interests center on wearable systems and machine learning for mobile health.

Mascolo currently is working on transforming earbuds into stethoscopes. She is exploring how wearable sensors around the face and head, which provide a stable base for monitoring, can be used to understand one’s fitness and health.

“The earbuds have an inward-facing microphone,” Mascolo explained. “As a result, we can hear sounds in the body, such as the heart and respiration, and use it as a stethoscope.”

This can be used in various applications in sports, as well as in cardiovascular and neurological health, she added.

“In the future,” Mascolo said, “I hope my research will go beyond just contributions to computer science and will be integrated with health disciplines and medical science.”

—John Delaney

**Open access, Basic and Premium,
and what we learned along the way.**

BY JACK DAVIDSON AND WAYNE GRAVES

Evolving the ACM Digital Library

THE ACM DIGITAL Library has long been central to how computing knowledge is disseminated, discovered, and preserved. Over time, it has come to serve a broad audience that includes researchers, educators, students, practitioners, librarians, institutions, and authors spanning every region of the world and every subdiscipline of computing.

In recent years, expectations around scholarly communication have shifted significantly. Open access has moved from an emerging model to a core requirement for many authors, institutions, and funding agencies. At the same time, users increasingly expect modern discovery tools, richer signals of value, and platforms that support a wide range of research, teaching, and professional workflows.

In response to these changes, ACM undertook a multi-year effort to transition the Digital Library to a fully open-access foundation. That transition had two tightly coupled components: moving to full open access and introducing a new service model with two tiers—Basic and Premium. When these changes were introduced, they generated substantial discussion within the community, including concern and, in some cases, strong pushback.

This article reflects on that transition and what we learned along the way. We explain why the changes were undertaken, how they were developed, what we heard from the community, how that feedback shaped subsequent decisions, and how we see the Digital Library continuing to evolve. Our aim is both explanatory and reflective, sharing lessons learned while reaffirming a commitment to transparency and ongoing engagement.

Why Change the Digital Library Model

The move toward open access did not arise from a single decision or moment. Rather, it emerged from the growing difficulty of reconciling several competing concerns within a single, monolithic model.

Authors increasingly require publication venues that comply with funder and institutional open-access mandates. Libraries desire sustainable and predictable approaches to access and cost. Readers expect immediate, barrier-free access to research results. At the same time, the Digital Library continued to grow in scale and complexity, supporting new content types, richer metadata, and more sophisticated platform features.

In parallel, ACM's publishing program was transitioning from a primarily subscription-based model to one supported by pay-to-publish mechanisms, including ACM Open agreements and author-paid article processing charges (APCs). This shift was necessary to support open access at scale, but it also required careful attention to long-term financial sustainability and pricing.

A particularly important tension emerged around institutions that rely heavily on the Digital Library but publish little or not at all in ACM venues.

Thousands of institutions worldwide fall into this category. For these institutions, “read and publish” agreements provide limited incentive or perceived value: Because they publish infrequently, they derive little benefit from publication-based arrangements, even though they make extensive use of the Digital Library for research, education, and professional practice.

Maintaining and improving the Digital Library infrastructure requires sustained, ongoing investment in platform development, infrastructure, and data quality. The central challenge, therefore, was not simply how to make content open, but how to do so while preserving quality, enabling innovation, and distributing costs fairly across a global and heterogeneous community.

Taken together, these considerations led ACM to pursue a model that separates open access to the scholarly record from additional platform services, resulting in the introduction of the Basic and Premium service model.

From Open Access to Basic and Premium Digital Library Editions

Basic: Open access as the foundation. Basic establishes open access as the foundation of the Digital Library. It ensures ACM content can be read, discovered, and used by anyone, anywhere, regardless of affiliation or ability to pay.

The Basic Edition experience is intended to provide:

- ▶ Open access to the scholarly re-

Maintaining and improving the DL infrastructure requires sustained, ongoing investment in platform development, infrastructure, and data quality.

cord, including articles and associated materials such as supplemental content, data, software, and videos where available

- ▶ Core discovery and navigation functionality
- ▶ Clear licensing and reuse information
- ▶ Article-level usage and citation indicators to support basic assessment and discovery
- ▶ A consistent experience for readers worldwide

In this sense, Basic represents a commitment to access, equity, and broad dissemination of computing research.

Premium: Sustaining and extending value. The Premium Edition builds on this foundation by offering enhanced features and services that support more advanced research and address institutional and analytical needs. These enhanced features include tools for deeper discovery, personalization, analytics, and integration that extend beyond basic access to content.

Equally important, Premium provides a subscription option for institutions and individuals that publish infrequently or not at all with ACM. At the same time, the more than 3,100 institutions participating in ACM Open receive Premium as part of their agreement. As a result, the vast majority of institutions and individuals that previously had access to enhanced Digital Library features continue to do so under the new model. Premium helps distribute the cost of platform investment more broadly, supporting sustainability while helping keep publishing-related fees as low as possible.

Community Engagement in the Design

From the outset, ACM sought to involve the community in shaping Basic and Premium models. That engagement took multiple forms, each serving a different purpose.

Focused workshops with librarians, authors, and users enabled detailed, candid discussions about discovery, metrics, sustainability, and perceived value. These smaller forums proved especially effective at surfacing nuanced concerns and practical considerations

that would have been difficult to identify through surveys alone.

To deepen feedback during development, the Digital Library Board formed a dedicated Basic/Premium Working Group composed of volunteer board members and staff. This group reviewed feature sets, design mock-ups, and beta functionality, and provided iterative guidance throughout development. Together, these efforts informed the initial design of the model and highlighted areas where clarity and iteration would be especially important.

Community Reaction

The introduction of Basic and Premium followed a staged rollout rather than a single cutover. A soft launch in mid-December preceded the official launch on Jan. 1, 2026, allowing ACM to observe early reactions, identify areas of confusion as well as emerging concerns, and make targeted adjustments before the new model formally took effect.

That approach was extremely valuable. Feedback received during the soft launch informed refinements to messaging, feature descriptions, and presentation, helping address issues that might otherwise have been amplified at full launch. The official Jan. 1 rollout reflected these early corrections and set a clearer baseline for broader community engagement. Despite this preparation, the official launch of Basic and Premium revealed gaps in shared understanding.

Feedback from across the community highlighted several recurring themes. Many expressed surprise at the introduction of tiered Digital Library services. Some worried that Basic might represent a significant reduction in value compared to the prior version. Others were uncertain about how the two tiers differed in practice, or what the changes signalled for the future of the Digital Library. These concerns were voiced through multiple channels, including town halls, direct correspondence, and community feedback.

In response to the feedback, ACM expanded its listening efforts, which included additional town halls with SIG leadership, direct outreach to librarians and institutional partners,

feedback mechanisms within the Digital Library, and continued discussions with volunteer boards and committees. These conversations were candid and, at times, uncomfortable. They were also essential. In several cases, they surfaced misunderstandings that would not have been obvious otherwise.

Adjustments and Clarifications

Community feedback led directly to several concrete changes, some of which were made quickly. In response to concerns about visibility and assessment, article-level usage and citation indicators were restored to the Basic experience, and aggregated bibliometric indicators were reintroduced on author pages. These changes reflected the community's view that such signals are important for understanding reach and impact, even in an open-access environment.

Feedback also highlighted the value of continued, smaller-scale dialogue. In addition to broad town halls, ACM scheduled a series of follow-up discussions with subsets of librarians, authors, and users to explore specific concerns in greater depth and test refinements before wider rollout.

Other adjustments focused on clarity and presentation. Messaging around Basic and Premium was refined, feature descriptions were revisited, and distinctions between open access to content and Digital Library services were made more explicit.

More broadly, this period reinforced the importance of increased transparency around planning and

prioritization. As the Digital Library continues to evolve, clearer communication about roadmap direction and decision-making will remain an area of focus.

Experimentation with AI

Another area where community feedback has been especially important is ACM's experimentation with AI-enabled features in the Digital Library.

As part of the platform evolution, Premium introduced AI-generated short and long summaries intended to help readers more easily discover articles and decide whether they are relevant to their interests. Written in a less technical style than author abstracts, these summaries were designed to support discoverability and encourage more readers to engage with the full articles themselves. While many users found these summaries useful, others raised concerns about accuracy, redundancy with author-written abstracts, and the role of AI-generated content in a scholarly context. Some also expressed discomfort with the use of AI in this setting.

That feedback prompted several immediate adjustments. Disclaimers were added, user feedback mechanisms were introduced, and summary generation workflows were refined. More broadly, it reinforced the need to approach AI features as experiments, subject to iteration and community input, rather than as fixed or definitive representations of scholarly work.

AI experimentation in the Digital Library extends beyond summaries, encompassing a range of discovery and navigation capabilities across articles and conferences. Planned and exploratory efforts include improved integrated search, recommendation capabilities, article-level interaction tools, and conference-oriented services such as personalized navigation, scheduling support, and exploratory access across the full history of a conference. These features are intended to help users navigate an increasingly large and complex corpus, not to replace the underlying scholarly record or the role of authors themselves.

To help guide this work, ACM's President established the ACM Presidential Task Force on the Use of AI, which is developing principles and

As the Digital Library continues to evolve, clearer communication about roadmap direction and decision-making will remain an area of focus.

guidance for responsible AI use across ACM activities. This task force provides an important governance framework that complements platform-level experimentation.

Within the Digital Library, AI features will continue to be introduced cautiously, evaluated openly, and refined based on feedback. The goal is to explore where AI can meaningfully improve discovery and understanding, while remaining transparent about its limitations and respectful of the scholarly community's values.

What This Means for Authors

Authors have a distinct relationship with the Digital Library. While all authors are also readers, the Digital Library serves as the authoritative long-term record of their scholarly contributions and professional identity.

The transition to open access is, at its core, an author-centered change. By removing barriers to reading and sharing, it increases the visibility, reach, and potential impact of ACM publications and their authors. For many authors, it also aligns ACM's publication programs more closely with funder and institutional open-access requirements.

At the same time, community feedback made clear that access alone is not sufficient. Authors care deeply about how their work is presented, discovered, and contextualized over time. Common questions focused on how articles reach broader audiences, how publication records and archives are represented across years and venues, and which Digital Library features are available to authors independent of their institution's access model.

In response, ongoing development increasingly treats the author as a first-class role within the Digital Library. This shift includes continued investment in author profile pages, clearer presentation of publication records and archives, improved linking across venues and versions, and better signaling of open-access status and reuse rights.

Looking ahead, we see clear opportunities for the Digital Library to better reflect the author's perspective. These include views that highlight authorship, publication records, and archives, and support an author's on-

Within the Digital Library, AI features will continue to be introduced cautiously, evaluated openly, and refined based on feedback.

going relationship with their work, alongside existing reader and institutional views.

Governance and Oversight

Throughout this transition, the Digital Library operated under the oversight of the ACM Digital Library Board, which reports into ACM's broader governance structure. The Digital Library Board includes volunteer representation from across the community and plays a central role in stewarding the platform's evolution, guiding strategic direction, reviewing major changes, and ensuring that community perspectives inform decision making.

The scope and impact of recent changes have also prompted reflection on governance itself. As the Digital Library increasingly sits at the intersection of publishing, access, technology, and policy, it is important that its governance continues to evolve in parallel. This includes ensuring appropriate representation and coordination across ACM's major boards and stakeholder groups, commensurate with the Digital Library's central role in ACM's mission.

Looking Ahead

The transition to open access, Basic, and Premium is not a completed project—it is an evolving framework. Future work includes continued refinement of both tiers based on usage and feedback, clearer articulation of value for different communities, and ongoing evaluation of sustainability and equity. Regular opportunities for community input will remain central to this process.

Lessons Learned


Several lessons stand out from this transition, which will inform how ACM approaches future platform and policy changes.

- ▶ Transparency must begin early and continue often, especially during periods of change.
- ▶ Engagement must be both broad and deep, combining large-scale input with focused discussion.
- ▶ Open access changes expectations, not just licensing.
- ▶ Staged rollouts, including soft launches, create space to learn, adjust, and reduce unnecessary friction at full release.
- ▶ Experimentation, particularly with emerging technologies such as AI, requires clear framing, visible safeguards, and ongoing community dialogue.
- ▶ Governance and communication must evolve alongside the platforms they oversee.
- ▶ The ACM community has an essential role in shaping expectations, policy, and the strategic direction of the Digital Library.

Conclusion


At its core, the ACM Digital Library exists to serve its community. The transition to open access and the introduction of Basic and Premium were driven by that mission, even when the path forward proved complex.

We are grateful for the engagement, critique, and collaboration of authors, readers, librarians, volunteers, and staff. That engagement has strengthened the Digital Library and helped align it more closely with the values it represents.

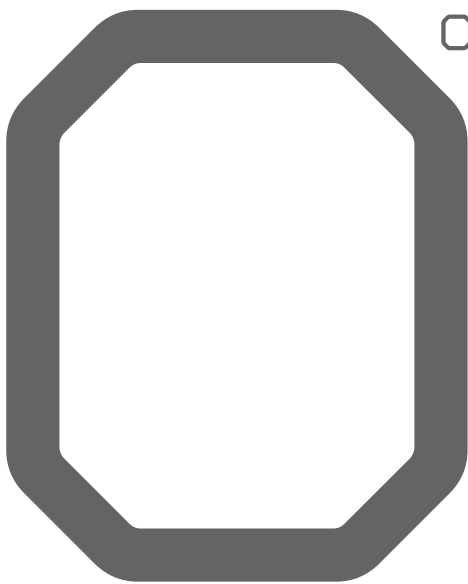
As the platform continues to evolve, including careful experimentation with new technologies, the work will remain guided by transparency, responsibility, and sustained dialogue with the community it serves. 

Wayne Graves is director of the ACM Digital Library, New York, NY, USA.

Jack W. Davidson is a professor of computer science at the University of Virginia, Charlottesville, VA, and is chair of the ACM Digital Library Board.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).



Opinion

From Passive to Participatory: How Liberating Structures Can Revolutionize Our Conferences

Tracking conference restructuring toward generating new ideas and discussing established work.

OUR CONFERENCES FACE a growing crisis: an overwhelming flood of submissions, increased reviewing burdens, and diminished opportunities for meaningful engagement. With AI making paper generation easier than ever, we must ask whether the current model fosters real innovation or simply incentivizes more publications. This Opinion column advocates for a shift from passive paper presentations to interactive, participatory formats. We propose *Liberating Structures* (LS)—facilitation techniques that promote collaboration and deeper intellectual exchange. By restructuring conferences into two tracks—one for generating new ideas and another for discussing established work—we can prioritize quality over quantity and reinvigorate academic gatherings. Embracing this change will ensure conferences remain spaces for real insight, creativity, and impactful collaboration in the AI era.

A Time for Change

Academic conferences have been at the heart of emerging research in computer science for more than 45 years. But our reliance on conferences to share and spark new research is starting to show some cracks. With a flood of submissions and new tracks in recent years, staying aware of relevant research and reviewing submissions has become a significant burden. And let us be honest: The rise of generative AI tools makes the task of writing more papers faster than ever.

Undeniably, we are at a crossroads we cannot ignore, and we must ask ourselves: Are more papers really better? As we spend more and more time writing, revising, submitting, and reviewing papers, is our community's research becoming more or less relevant to society's challenges, and are we really pushing the boundaries of our fields? With this increase in papers and tracks, we should also ask ourselves how our experience attending conferences has changed. As we

struggle to attend and passively listen to even a few paper presentations from the many parallel tracks, are we really learning effectively about others' work? Are we engaging in deep conversations and debates that spark new ideas? Are we making meaningful connections and forming new collaborations?

As AI reshapes our profession, fostering meaningful human engagement is more essential than ever. With the increasing scale of conferences, creating participatory spaces for genuine intellectual exchange becomes crucial to supporting deep engagement with emerging ideas. The surge in submissions, however, risks overwhelming innovation with a high volume of contributions that may, while numerous, lack significant novelty or impact. After more than 45 years and the disruptive ways of new technology, is it time to rethink our conference model? It may be time for a radical shift in organizing our conference events. By adopting more deliberate quality over quantity-focused



approaches, we can aim to reinvigorate conferences and enhance what we gain intellectually from them. We can transform our academic gatherings into spaces prioritizing thoughtful, deep engagement over superficial busyness and quantitative publishing metrics. It is not about doing more; it is about doing what matters most—and doing it well.

Here, we propose LS as a concrete way to achieve a shift from passive conference presentations to more conversation and thinking. Liberating Structures, a set of facilitation methods^{2,7} make it easier for attendees to collaborate, share ideas, and generate innovative solutions by breaking down hierarchical barriers and fostering open, creative dialogues. By embracing LS, we can transform conferences into spaces that inspire and generate impactful community-driven research that supports disruptive thinking and shapes collaboration and innovation.⁶ It is time to embrace a model that values quality over quantity and

prioritizes depth over haste, ensuring our contributions to the field of computing are genuinely significant.

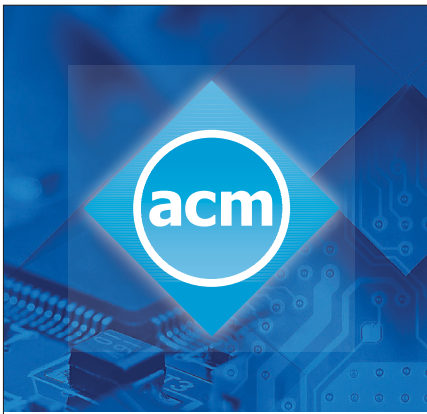
The Unsustainable Problem with Our Conferences

As an example, today's software engineering researchers are easy to recognize by their long hours at work and their harried expression of trying to keep up with the need to publish more and more papers and simultaneously re-

It is time to embrace a model that values quality over quantity and prioritizes depth over haste.

view dozens of papers. It is not unusual to overhear at conferences researchers mentioning that they are retreating to their hotel rooms to refine or review papers for the next conference or journal. When they do attend presentations, they only half-listen while on their laptops because of the stress they feel about the work they are not doing. When they do their own research back home, they find it challenging to find blocks of time to focus and review the literature relevant to their work in the face of daunting reviewing commitments. This cycle of an ever-increasing paper publication, an “arms race” of sorts, raises serious concerns about sustainability—human, social, and professional.

On a human level, researchers face burnout, mental fatigue, and declining well-being as personal boundaries blur under constant demands. On a social level, the culture of overwork can erode meaningful collaboration and mentorship, deterring young talent and weakening the community's cohesion. On



Advertise with ACM!

Reach the innovators and thought leaders working at the cutting edge of computing and information technology through ACM's magazines, websites and newsletters.



Request a media kit with specifications and pricing:

Ilia Rodriguez

+1 212-626-0686

acmm mediasales@acm.org



a professional level, the emphasis on quantity undermines deep inquiry and thoughtful reviewing, threatening the integrity and quality of our field's output. Regarding sustainability, the impact of travel on our environment and unfairness in terms of who can afford to attend cannot be ignored. Furthermore, many researchers are burning out with the demands of paper and travel, and the visibility of this culture turns many talented people away.

As a concrete example, the International Conference on Software Engineering (ICSE), to many the leading software engineering venue, saw a dramatic rise in submissions over the past decade. In 2014, ICSE received 496 submissions, which increased almost linearly to 691 by 2022. However, the recent surge has been even more pronounced, with submissions jumping to 798 in 2023 and exceeding 1,469 in 2026. Some may speculate that this accelerating growth is partly due to AI tools making it faster to generate and manage submissions (although using AI for reviewing is discouraged, adding an imbalance to the process). We have heard some colleagues mention they review hundreds of papers every year. This work is appreciated, but does it come at the cost of the thoughtful work this researcher could do that would lead to more meaningful work by the community as a whole?

Transforming Conferences with Liberating Structures

To address these challenges, we need innovative solutions that foster more engaging and interactive environments. One promising approach is the adoption of LS. These simple yet powerful methods can fundamentally change how we organize and participate in conferences, moving away from the conventional, often passive formats to more dynamic and inclusive ones.

Liberating Structures are easy-to-learn interaction techniques designed to distribute control and invite contributions from all participants. Unlike traditional meeting formats that centralize control and limit participation, LS ensure everyone's voice is heard and valued. Each LS protocol consists of five elements: the structuring invitation, space arrangement, participation distribution, group configurations, and a sequence of steps. These elements

guide the group through a process that fosters creativity, inclusion, and collaboration. For instance, consider the LS method "Impromptu Networking." In a conference session using this approach, the facilitator poses a relevant question, such as "What is the most significant challenge in integrating AI into software engineering?" Participants pair up to discuss their thoughts for two minutes before switching to new partners. In just a few rounds, everyone has shared ideas with multiple people, quickly surfacing diverse perspectives and common themes.

The primary benefit of LS is their ability to transform interactions in meetings, workshops, and conferences. By breaking down hierarchical barriers and encouraging equal participation, LS creates an environment where new ideas emerge, and meaningful connections can form. This leads to increased innovation, better problem solving, and a stronger sense of community among participants. LS also make meetings more engaging and productive by ensuring everyone can contribute and that leads to a process that is more dynamic and interactive.

The software engineering community has already started experimenting with LS to address the limitations of traditional conference formats. For example, a workshop at the Foundations of Software Engineering conference (FSE) focused on the "2030 Roadmap for Software Engineering" special issue of *ACM Transactions on Software Engineering*. This workshop used LS to enhance the cross pollination of ideas from existing papers and to generate new collaborations for future research topics. In parallel, other communities such as NASA have found that structures such as Critical Uncertainties, Ecocycle Planning,

We need innovative solutions that foster more engaging and interactive environments.

and Purpose to Practice are particularly effective when the aim includes fostering innovation and interdisciplinary collaboration.³ Together, these experiences demonstrate how LS can facilitate both immediate and long-term collaboration across diverse domains.

Another example is the Copenhagen Symposium on Human-Centered Software Engineering AI (held in 2023 and again in 2024). The symposium aimed to identify relevant research topics within the community and establish ongoing research workstreams. By using LS, participants committed to working remotely on chosen topics throughout the year, with plans to reconvene the following year to discuss their progress. This continuous cycle of collaboration and feedback, supported by the Sloan Foundation, exemplifies how LS can sustain engagement and innovation. For instance, through iterative processes, participants at the symposium in 2023 co-developed the Copenhagen Manifesto.⁵ This manifesto focuses on the responsible integration of generative AI and highlights how collective dialogue can produce shared visions that emphasize ethical, transparent, and human-centered approaches to technology. The symposium in 2024 was even more effective, as many participants knew what to expect after using LS in 2023 and thus engaged more quickly.

A Proposal for a New Model for Conferences

Building on the success of such experiences, we can envision a new conference model that ensures they remain relevant and impactful in the future. This model includes two main tracks: the *Creation Track* and the *Sharing Track*. This approach leverages the power of LS to promote a healthier, more engaged academic ecosystem.

In the *Creation Track*, research topics emerge organically through LS techniques. Participants engage in dynamic, interactive sessions designed to spark new ideas and collaborations. Techniques such as Impromptu Networking, 1-2-4-All, and Troika Consulting¹ allow attendees to brainstorm, refine, and develop research topics in real-time. This track encourages participants to explore their shared interests deeply and commit to collaborative research projects. By fostering an environment of creativ-

By adopting Liberating Structures, we can cultivate more inclusive, productive, and engaging events.

ity and inclusion, the *Creation Track* ensures that the ideas generated are diverse, innovative, and aligned with the broader community's interests.

The *Sharing Track* focuses on presenting and discussing mature, impactful ideas already published in journals or proceedings. It is designed to share fully developed research findings, best practices, and significant advancements with the broader community. Sessions in the *Sharing Track* may include LS-enhanced formats like Fishbowl Conversations and Conversation Café, encouraging active participation and deeper engagement from the audience. By presenting mature ideas in a structured yet interactive format, the *Sharing Track* ensures conference content is both relevant and high quality. This track allows researchers to showcase their work, receive valuable feedback, and inspire further innovations within the community, providing a platform for celebrating successes and learning from each other's experiences.

We recognize that small, deliberate steps will be necessary to ensure a smooth transition from the current conference model to a more interactive and impactful one. Introducing LS in the form of pilot sessions within traditional conferences can be a first step, allowing participants to experience formats like Fishbowl Conversations and Conversation Café alongside standard presentations. Over time, these interactive methods can foster deeper engagement and collaboration, encouraging a cultural shift toward quality over quantity. This gradual approach can help address concerns around current incentive structures, where some institutions reward researchers simply for publishing in prestigious venues like ICSE. We can promote long-term innovation and meaningful progress

by gradually realigning conference success metrics to emphasize impactful, community-driven contributions. This shift, facilitated by the *Creation* and *Sharing Tracks*, will transform conferences into dynamic hubs where real breakthroughs are celebrated.

Conclusion

The time has come to reimagine our approach to academic conferences, ensuring they remain meaningful, impactful, and genuinely conducive to knowledge advancement. By adopting Liberating Structures, we can cultivate more inclusive, productive, and engaging events—transforming conferences from passive gatherings into dynamic spaces for creativity, collaboration, and innovation. As Cal Newport suggests, the path forward is not about doing more, but about doing what matters most with a focus on quality over quantity.⁴ We recognize some challenges to changing what we do, but let us embrace this shift and foster a healthier academic ecosystem that values thoughtful research, meaningful connections, and impactful contributions. ■

References

1. Lipmanowicz, H. and McCandless, K. *Liberating Structures*. (2014); <https://www.liberatingstructures.com/ls>
2. Lipmanowicz, H., Singhal, A., McCandless, K., and Wang, H. *Liberating Structures: Engaging everyone to build a good life together*. In *Communication and "The Good Life"*. H. Wang, Ed. International Communication Association Theme Book Series (2015).
3. McCandless, K. The deliberate irony of Liberating Structures. In *Encyclopedia of Interdisciplinarity and Transdisciplinarity*. Edward Elgar Publishing (2024).
4. Newport, C. *Slow Productivity: The Lost Art of Accomplishment Without Burnout*. Portfolio, NY (2023).
5. Russo, D. et al. Generative AI in software engineering must be human-centered: The Copenhagen Manifesto. *J. of Systems and Software* 216 (2024).
6. Storey, M.-A. et al. A disruptive research playbook for studying disruptive innovations. *ACM Trans. on Software Engineering and Methodology* 33, 8 (2024).
7. Torbert, W.R. Educating toward shared purpose, self-direction and quality work: The theory and practice of liberating structure. *The J. of Higher Education* 49, 2 (1978).

Daniel Russo (daniel.russo@cs.aau.dk) is an associate professor in the Department of Computer Science, Aalborg University, Denmark.

Margaret-Anne Storey (mstorey@uvic.ca) is a professor of computer science at the University of Victoria, Canada.

© 2026 Copyright held by the owner/author(s).



Watch the authors discuss this work in the exclusive *Communications* video. <https://caacm.acm.org/videos/revolutionize-conferences>

Opinion

The Coming Commoditization of Computational Thinking

Reimagining professional and educational practices for an AI-augmented future.

COMPUTATIONAL THINKING—THE ABILITY to formulate and solve problems with computing tools—is undergoing a significant shift. Advances in generative AI, especially large language models (LLMs),² are democratizing this skill and making high-level problem solving broadly accessible. Today, nearly anyone with an Internet connection can leverage LLMs to tackle complex tasks using plain English instructions, often without writing a single line of traditional code, thereby commoditizing capabilities that once required specialized programming expertise.

This emerging paradigm lowers longstanding barriers. Jeannette Wing’s seminal vision of “computational thinking”⁹ emphasized logical problem decomposition and algorithmic idea design. Historically, putting that into practice meant mastering programming languages like Python, Java, or C++ and wrestling with arcane syntax and debugging, which limited who could transform ideas into software. Now, however, natural-language interfaces enable users to describe problems plainly and let an AI generate a solution.⁴

Consequently, knowledge work is at a historic inflection point. Tasks such as report writing, dataset analysis, and code drafting—once exclusive to skilled professionals—are increasingly automatable by AI, which creates unease about the future of many roles. In practice, a financial analyst armed with an LLM can accomplish in hours

what used to take days. Likewise, a programmer paired with an AI assistant can produce software faster and often with comparable quality.

The net effect is that computational thinking is becoming a ubiquitous utility service.



Even the very definitions of “programmer” or “developer” are evolving. If solving a problem through an interactive dialogue with an LLM counts as programming, then the ranks of those capable of “programming,” broadly defined, are expanding rapidly. *Prompt engineering*—the skill of crafting precise, effective instructions for AI—has become a recognized competency in this landscape.⁸ The developer’s role is already shifting toward guiding or editing AI-generated work: the human outlines the problem, the AI drafts a solution (be it code, analysis, or design), and the human refines and verifies it.

Creative and architectural decisions remain human-led, but much of

the routine implementation and information gathering associated with developing software-reliant systems can now be offloaded to LLMs. The net effect is that computational thinking is becoming a ubiquitous utility service. This democratization brings great opportunity but also raises urgent questions for computing professionals and educators alike.

Beyond Hand-Coding: LLMs Redefine Programming

Perhaps nowhere is the impact of LLMs more evident than in software development. Generative AI models can produce substantial chunks of source code from minimal prompts, fundamentally augmenting the programming process. Developers using AI coding assistants like GitHub Copilot, OpenAI Codex, or Claude Code are effectively working alongside a tireless AI coding partner. The results are striking: according to GitHub, nearly 46% of code written by developers who use Copilot is now AI-generated on average (and over 60% in certain languages such as Java).⁷

The value of LLMs goes beyond programming speed by enabling developers to work at a higher level of abstraction, focusing more on intent and design rather than slogging through boilerplate.¹⁰ In effect, LLMs are redefining what it means to “write a program.” Instead of painstakingly coding every detail, many tasks can be accomplished by describing the problem to an LLM. The act of programming starts to look more like giving instructions in plain language, then iteratively refining the AI’s output. Programming is thus becoming more of a *conversation* with the computer, with humans and AI working in tandem to produce solutions.⁶

As AI-generated code moves from toy examples to production-scale software, developers must apply the same rigor that traditional software engineering demands. Companies using LLMs in their software pipeline report that naive prompt-hacking leads to brittle code, unforeseen failures, and technical debt.⁵ In response, emerging best practices for *robust prompt engineering* echo long-established practices from software engineering.¹ These include systematic testing and valida-

Perhaps nowhere is the impact of LLMs more evident than in software development.

tion of AI outputs, version control and documentation of prompts, iterative refinement based on bug reports, and integration of AI-produced modules with conventional code. These practices mirror traditional engineering disciplines and are critical for reliability.

Probabilistic, Not Perfect: The Limits of Current LLMs

For all their impressive capabilities, however, today’s LLMs have well-known limitations that demand human oversight.² Unlike traditional programs, LLMs don’t follow a deterministic series of logical steps on a dataset. Instead, they generate responses by making a sequence of probabilistic choices guided by patterns in their training data. This fundamental difference means that LLMs cannot guarantee correctness or truthfulness.

Moreover, these failures are not easily fixed by simple means like prompt rewording or model tuning. It might appear that clever “prompt engineering” (for example, instructing the AI to double-check its work or think step-by-step) could eliminate mistakes. Careful prompting can reduce errors⁸ but are no silver bullet. Because LLM outputs are generated from statistical correlations rather than grounded in formal rules, there will always be a non-zero chance of a flaw.

In practical terms, these limitations with LLMs mean that professionals must cultivate a healthy skepticism and a habit of verification. If an AI assistant writes a piece of code, it should be tested and code-reviewed like any other code since subtle errors may be hidden by the AI’s polished delivery. These powerful probabilistic tools are best used as assistants, not oracles. The role of human experts is not eliminated but instead becomes even more

pivotal to interpret and validate AI’s contributions.

New Skills, Old Foundations: Education in the LLM Era

What does the advent of LLMs mean for computer science education? In an era where students can effortlessly generate a working program via natural-language prompts, educators face a dilemma: how to teach foundational computing skills when the “dirty work” of coding or crunching numbers is increasingly handled by AI. Some might argue that traditional programming courses are becoming obsolete. After all, why spend weeks learning Java syntax or debugging pointers in C++, when an AI copilot can produce a solution in seconds? The answer, however, is not to abandon the foundations, but to adapt our curriculum to the new reality.

First, we must integrate AI tools *into* education rather than try to ban them. Just as calculators found their way into math classes (accompanied by lessons on when and how to use them), generative AI should find its way into programming courses with appropriate guidance. Forward-thinking instructors are designing assignments that encourage responsible AI use. Rather than deeming it “cheating,” they guide students to use these tools appropriately. Computing educators echo this sentiment,³ urging we teach students how to collaborate with AI and critically evaluate AI-generated results.

Students must become fluent in AI-assisted problem solving—learning to craft effective prompts, iterate in dialogue, and decompose tasks for machine partners. Alongside that, however, students must hone their critical reasoning and validation skills, for example, testing AI outputs by writing unit tests for code an AI writes and learning to recognize when generated code or results diverge from expectations and requirements.

Importantly, none of this means the old foundations become irrelevant. On the contrary, a strong grasp of computing fundamentals is more crucial than ever—it is what enables one to detect when the AI is going astray. Understanding algorithms, data structures, and complexity helps developers no-

tice if AI-generated solutions are inefficient or incorrect. Knowledge of security and correctness principles alerts an engineer when an AI's code suggestion might be vulnerable or flawed.

Beyond technical skills, ethical and societal discussions must become a bigger part of the curriculum. With LLMs generating content, questions of academic integrity naturally arise, so students and faculty need guidelines on what constitutes acceptable use of AI in coursework. By engaging students with these questions, we prevent misuse *and* produce professionals who are thoughtful about AI's broader implications. In short, integrating LLMs into education is not just about tools; it offers students a broader view of how technology intersects with responsibility and social outcomes.

Conclusion

The rise of LLMs marks the next chapter in the decades-long democratization of computing. Just as spreadsheets in the 1980s allowed non-programmers to harness computation, and the Web in the 1990s put information at everyone's fingertips,

The rise of LLMs marks the next chapter in the decades-long democratization of computing.

AI language models now make problem solving accessible to the masses. The computing community should neither resist this tide nor be swept away by it. Instead, we must reimagine our professional and educational practices for an AI-augmented future. Those who learn to combine human insight and judgment with AI's capabilities will achieve results previously out of reach. Those who cling to older paradigms may find themselves left behind. Computational thinking may be commoditized, but human creativity, critical thinking, and ethical

responsibility remain essential. By cultivating these strengths in tandem with AI, we ensure the coming commoditization of computational thinking augments human achievement, not replaces it. **C**

References

1. Chen, Z. et al. Promptware engineering: Software engineering for LLM prompt development. *arXiv preprint* (2025); <https://arxiv.org/abs/2503.02400>
2. Cerf, V.G. Large language models. *Commun. ACM* 66, 8 (Aug. 2023).
3. Denny, P. et al. Computing education in the era of generative AI. *Commun. ACM* 67, 11 (Nov. 2024).
4. Michaelson, G.A. and dos Santos, R.P. Is English the new programming language? How about pseudo-code engineering?. *Acta Scientiarum* 26, 1 (Jan./Feb. 2024).
5. Menshawy, A. et al. Navigating challenges and technical debt in large language model deployment. In *Proc. EuroMLSys'24* (2024).
6. O'Reilly, T. AI and programming: The beginning of a new era. *O'Reilly Radar* (May 8, 2025).
7. Ramel, D. GitHub Copilot AI tech upgraded, already generates 61% of Java code. *Visual Studio Magazine* (Feb. 15, 2023).
8. White, J. et al. A prompt pattern catalog to enhance prompt engineering with ChatGPT. In *Proc. 30th Pattern Languages of Programming Conf. (PLoP'23)* (2023).
9. Wing, J.M. Computational thinking. *Commun. ACM* 49, 3 (Mar. 2006).
10. Ziegler, A. et al. Measuring GitHub Copilot's impact on productivity. *Commun. ACM* 67, 2 (Feb. 2024).

Douglas C. Schmidt (dcschmidt@wm.edu) is Dean of the School of Computing, Data Sciences & Physics at William & Mary, Williamsburg, VA, USA.

© 2026 Copyright held by the owner/author(s).

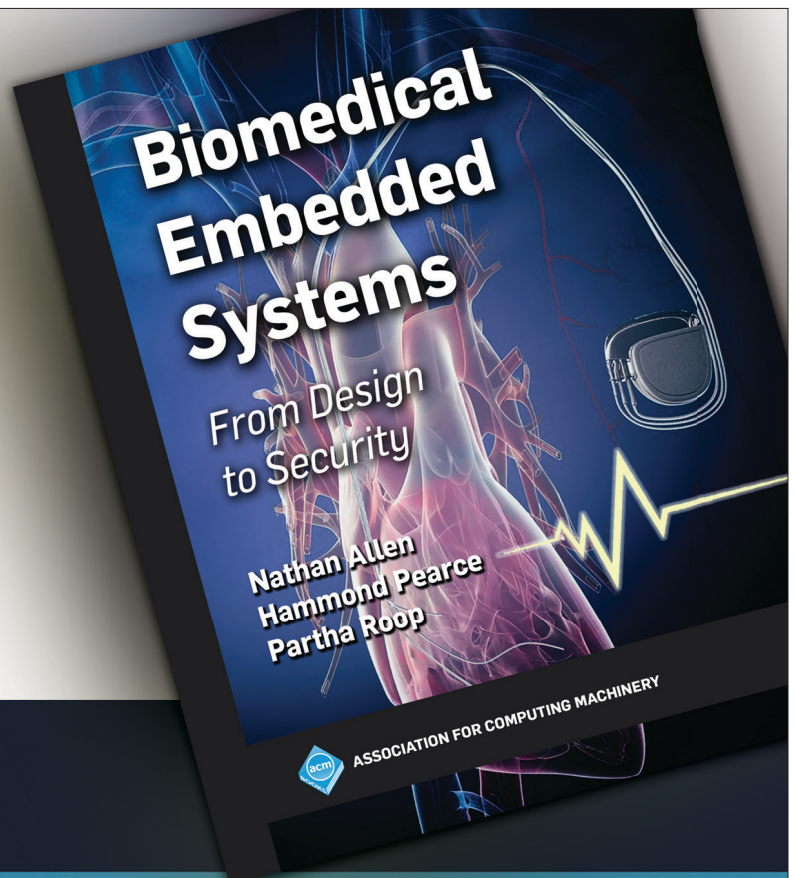
Biomedical Embedded Systems

From Design to Security

**Nathan Allen
Hammond Pearce
Partha Roop**

**ISBN: 979-8-4007-3155-6
DOI: 10.1145/3731217**

<http://books.acm.org>



ACM BOOKS
Collection III

Technology Strategy and Management

The Importance of Geopolitics in AI Development

Considering the unique challenges for learning from past geopolitical tensions.

IN WHAT WAYS is geopolitics—rivalry between nation-states for hegemony—affecting the flow of resources necessary to develop and deploy artificial intelligence (AI)? Answer to this question matters greatly to those in countries with a stake in being at the top of the AI race.^a But when we as experts are asked to draw implications of geopolitics for AI, we seem to be somewhat at a loss. This is in part because the necessary expertise to address this issue is distributed, crossing boundaries of technological knowhow—the domain of data scientists—and of policymaking—the domain of politicians and diplomats. It is also because geopolitics for AI has active stakeholders in the private sector, including tech firms that develop AI and investors that finance AI development.

For some, AI as general-purpose technology is about progress, just like steam power and electricity that lifted the human race out of drudgery to enjoy higher standards of living. For others, AI poses risks of job loss, hitting



the young more than older workers, challenging a widely held assumption that education provides a solution to societal ills such as inequality. For yet others, AI is the arena in which great powers such as China and the U.S. play out their rivalry. AI hawks in the U.S. worry that this emerging technology could undercut America’s national security and geopolitical hegemony. When we hear Nvidia’s CEO Jensen Huang say “China is going to win the AI race,”¹² how should we evaluate and react to such a statement?

In this column, I ask: What do we need to know about the current links between geopolitics and AI, and likely future scenarios? What is unique about AI that makes learning from past geopolitical tensions more challenging? Knowing the answers to these questions is as important to businesses and data scientists that work for them as for diplomats and national security advisors.

Geopolitics: Shift over Time

Geopolitics used to be about the study

^a See State Council of the People’s Republic of China. *New Generation Artificial Intelligence Plan*. G. Webster et al., translators, Stanford Cyber Policy Center (Aug. 1, 2017); <https://digichina.stanford.edu/work/full-translation-chinas-new-generation-artificial-intelligence-development-plan-2017/>

ACM Distinguished Speakers

**A great speaker
can make the
difference between
a good event and
a WOW event!**

Take advantage of ACM's Distinguished Speakers Program to invite renowned thought leaders in academia, industry, and government to deliver compelling and insightful talks on the most important topics in computing and IT today. ACM will cover the cost of transportation for the speaker to travel to your event.

speakers.acm.org



Association for
Computing Machinery

of how physical geography and related factors, such as natural resource endowment, affect a country's politics and foreign policy. For instance, an island nation like the U.K. has vastly different opportunities to secure its borders than countries with land borders. Political power connected to geography made some countries a sea power, giving them leverage in naval affairs. Fast-forward into the last few decades, even Henry Kissinger, who served as national security advisor in consecutive U.S. administrations, came to the view that geography trumped ideology in shaping a country's expansionist ambition. He therefore concluded when the Cold War ended that whoever governed Russia would continue to be a threat to the U.S.⁸ Great powers can also launch a geographic project to enhance its geopolitical power. A good example here is China's Belt and Road Initiative, linking China by sea and land to Central Asia and Europe, to assert its hegemony in the region.

In recent years, there has been a notable shift from geopolitics to what some call geoeconomics. Geoeconomics is "the use of economic resources to promote and defend national interests and to produce beneficial geopolitical results."¹ The use of economic policy—tariffs, export controls, and so forth—as a tool of national security, however, has long historical roots, in the mercantilist tradition advocated by Alexander Hamilton. And in the 21st century, the underlying mechanism of geoeconomics is the "weaponization" of interdependence that has resulted from decades of globalization.⁵ In AI development and use, geoeconomics manifests itself in governments' control over strategic assets, or restricting access to such assets by adversaries, as in the case of U.S. denying China access to the most advanced chips made by Nvidia.

Linking Geopolitics to AI

AI is digital and therefore has little to do directly with physical geography. But AI depends greatly on tangible resources, including semiconductors and raw materials used to manufacture them and electricity that powers datacenters. Seen in this way, geopolitics (and its geoeconomic cousin) af-

fect AI greatly, via shaping access to resources—including minerals, components, talent—that are deemed essential for its development and use.

Moreover, because of the use of AI to enhance the defense and security capabilities of nations, some observers draw parallels with the nuclear arms race. However, this parallel has limited validity due to differences in technology and stakeholders. AI is a general-purpose technology whose development is driven by private tech giants, and whose use is in part determined by the general public, unlike nuclear weapons, which are discrete state-controlled hardware for military supremacy. Stockpiling and mutually assured destruction for nuclear weapons result solely from actions by nation-states, thus rendering them more predictable and accountable. By contrast, the AI race involves commercial competition and societal transformation over and above state actions, resulting in a broad range of risks.

Rivalry between China and U.S.

Most recently, the AI rivalry between China and the U.S. was brought home by the release of DeepSeek's R1 reasoning model in January 2025, just a few months after OpenAI's launch of its pioneering o1 model. DeepSeek, a private entrepreneurial firm in China, struck the world with its design principles of cost-efficiency, rapid deployment, and targeted applications. The resulting competitive dynamics unravelling among pioneer and fast follower providers of AI foundation models are intricately linked to the rivalry between China and the U.S.

Benchmarking various dimensions of AI development shows the U.S. is ahead, but China is quickly

**In recent years, there
has been a notable
shift from geopolitics
to what some call
geoeconomics.**

catching up. According to one study, the U.S. leads in four of the six categories of metrics examined (talent, research, development, and hardware), and China leads in two (adoption and data).³ AI adoption in this study is proxied by the number of broadband subscriptions and mobile payments. But why is the U.S. “ahead of the game” in hardware, and China in applications? The reason may lie in the fact that China and the U.S. are playing a different game.

Some of the China-U.S. differences stem from different political systems with China being state-led and the U.S. being market-led. China is pursuing industrial policies with state funding to promote energy, chips, state-backed AI labs, and data.⁴ But this simplistic contrast between state-led and market-led systems does not capture the whole picture. Today, and at least since the 2010s, Chinese entrepreneurial companies have played a significant role in developing a competitive AI ecosystem that differs from Western systems.

China is committed to building a modular and resilient AI infrastructure, in which the emphasis is on customized infrastructure (for example, Alibaba Cloud’s storage solutions), cost efficiency in model development (for example, AI models have significantly lower training and inference costs), and calibration for real-world applications.⁷ And this U.S.-China contrast is symptomatic of the different approaches taken by Silicon Valley (investing massive amounts of venture capital in cutting-edge technology) and China’s ecosystem (brimming with fast-follower startups willing to do the grunt work to adapt their product applications to meet consumer demand).⁹ The consequence of this business model in China is a wide diffusion of AI across all segments of the economy, including e-commerce, finance, healthcare, logistics, EVs, automated ports, and intelligent manufacturing.

China’s policy to promote AI hinges on a combination of mobilizing public funds and private entrepreneurial firms which are expected to abide within the state system. A good example of this private-public consistency is the decision by DeekSeek and

AI is digital and therefore has little to do directly with physical geography.

other providers (such as Baidu, Moonshot AI and Meituan) to offer open source AI. Open source has the advantage of low-cost adoption and rapid diffusion of applications.¹⁰ It is a business decision, as seen for instance by Meta’s open source approach through its Llama models. But open source also enhances the state’s supervisory and surveillance capacity, thus making the decision consistent with the Chinese state goals and stance.

Private Sector Players’ Influence in Geopolitics

Geopolitics is primarily a domain of nation-states. Yet private sector actors play an important role with respect to AI and by extension geopolitics due to a number of factors governing business-government relations. Here, I highlight corporate spending on AI, business engagement in corporate diplomacy¹⁴ as remaining neutral in geopolitics is often not an option, and the enhanced definition of sectors affecting national security.

First, private sector spending on AI dwarfs public sector spending. According to one estimate, global corporate AI investment reached \$252.3 billion in 2024 experiencing a dramatic thirteenfold growth during the past decade.¹¹ In the U.S., the 2024 private sector AI investment was more than \$109 billion, compared to the federal non-defense AI R&D funding of approximately \$3.3 billion.² Even in the European Union, 73% of the total 257 billion euros spent on AI investment by EU27 countries in 2023 was estimated to be in the private sector.⁶ The U.S. is squarely focused on maintaining and strengthening the primacy of the U.S. AI industry, but with funding from businesses. This necessitates positioning private capital investment as the U.S.’s geopolitical ma-

neuver. Consider the way President Trump in January 2025 championed the announcement of the Stargate Project, a joint venture among OpenAI, Oracle, and Softbank to commit \$500 billion investment over four years to build AI infrastructure such as datacenters.¹⁵

Second, given this government stance, private sector firms are either taking sides, or walking a tightrope, to be consistent with their respective competitive position. A contrast between OpenAI and Nvidia illustrates the situation well. On the one hand, OpenAI is clearly taking sides, endorsing the U.S. government’s stance to divert potential investment capital flow away from China. OpenAI could not have been more explicit when they stated: “With an estimated \$175 billion sitting in global funds awaiting investment in AI projects, if the U.S. does not attract those funds, they will flow to China-backed projects—strengthening the Chinese Communist Party’s global influence.”¹³ On the other hand, Nvidia got caught in the geopolitical firing line, when one day, its CEO Jensen Huang stated that “China is going to win the AI race,” only to replace it the following day by “China is nanoseconds behind the U.S.”¹² Nvidia evidently did not wish to antagonize the U.S., and its hope of easing export controls of advanced AI chips was realized soon thereafter.

Third, while private actors play a greater role in geopolitics, governments are extending their definition of national security well beyond the traditional understanding of strategic sectors in defense to most sectors of the economy with digital connectivity (subject to cybersecurity risks). Specifically, the U.S. identified 16 “critical infrastructure sectors,” whose “incapacitation or destruction ... have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof.”^b Similarly, the U.K. specified 17 areas of the economy (AI being one of them) in which

b See “Critical Infrastructure Sectors” defined by the U.S. Cybersecurity and Infrastructure Security Agency. <https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors>

**We Must Teach
Computer Science
in Prison**

**A Future AI May Create
Concepts Unknown
To Mankind**

**Rolling in the Deep of
Cognitive and AI Biases**

**Evaluating General-
Purpose AI with
Psychometrics**

**The Indispensable
Role of User Simulation
in the Pursuit of AGI**

**A Research Agenda
for the Economics of
Transformative AI**

**The Golden Rule of Big
Memory: Persistence Is
Not Harmful**

**Are We Actually There?
Assessing RPKI Maturity**

**Agentic AI Software
Engineers:
Programming
with Trust**

Plus, the latest news
about passwords, sensory
augmentation, and
the outlook for CS Ed.

In a world in which governments rely more on private-sector actors, rules of the game are co-created by governments and these actors.

the government could scrutinize, and block, acquisitions for potential national security harm.^c With such shift in the meaning of national security, governments' geoeconomic toolkits—sanctions, tariffs, investment control, and so forth—can be justifiably applied to many different sectors that deploy AI.


Future of Geopolitical Influence in AI

Amidst all the efforts made by universities and businesses to develop AI, this column urges businesses, investors, and data scientists who work for them to pay explicit attention to the impact of geopolitics on AI development.

The AI race is much more affected by what we do, compared to say the nuclear arms race, because AI development is driven by commercial firms, and its use is determined by us as citizens. In a world in which governments rely more on private-sector actors, rules of the game are co-created by governments and these actors. This applies as much to co-creating the rules for geopolitical discourse as to co-developing domestic regulations that govern AI, as demonstrated by the previous discussion of open source AI in China and OpenAI's

^c See "National Security and Investment Act: Details of the 17 types of notifiable acquisitions." U.K. Cabinet Office (Feb. 6, 2025); <https://www.gov.uk/government/publications/national-security-and-investment-act-guidance-on-notifiable-acquisitions/national-security-and-investment-act-guidance-on-notifiable-acquisitions>


alignment with government stance in the U.S.

The next time we hear a big tech executive make a geopolitical maneuver, via a public statement, we need to judge not only the extent of commercial benefits derived from it, but also the potential influence they wield in policymaking. The next time you see your government restricting exports, imports or investment in your sector, remember their newly extensive definition of national security to apply to most sectors with connectivity, rendering most workers to be in strategic sectors affected by geoeconomics. This lens also enables us to anticipate how the resources necessary for AI deployment are likely to flow, whether the resources are energy, datacenter infrastructure, or the movement of AI talent. 

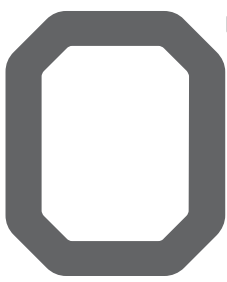
References

1. Blackwill, R.D. and Harris, J.M. *War by Other Means: Geoeconomics and Statecraft*. Harvard University Press (2017).
2. Borges, C. and Deng, Y. *Innovation Lightbulb: Federal R&D Funding Matters For U.S.* AI Leadership Center for Strategic and International Studies (Oct. 22, 2025).
3. Castro, D., McLaughlin, M., and Chivot, E. *Who Is Winning the AI Race: China, the EU or the United States?* Center for Data Innovation (Aug. 2021).
4. Chan, K. et al. *China's Evolving Industrial Policy for AI* (June 26, 2025); <https://www.rand.org/pubs/perspectives/PEA4012-1.html>
5. Farrell, H. and Newman, A.L. Weaponized interdependence. *Intern. Security* 44, 1 (2019).
6. Fonteneau, F. et al. *Advancing the Measurement of Investments in Artificial Intelligence*. OECD Artificial Intelligence Papers 47 (Sept. 2019).
7. Joshi, A. et al. How savvy companies are using Chinese AI. *Harvard Business Rev.* (Sept.–Oct. 2025).
8. Kissinger, H. *Diplomacy*. Simon & Schuster, New York (1994).
9. Lee, K. *AI Superpowers: China, Silicon Valley, and New World Order*. Harper Collins, New York (2018).
10. Lee, K. China's open-source AI is a national advantage. *Financial Times* (Dec. 5, 2015.); <https://www.ft.com/content/b1f92b0e-d6ef-4c95-b51e-7bcf90c8a65f>
11. Maslej, N. et al. *AI Index Annual Report*. Institute for Human-Centered AI, Stanford University, Stanford, CA (Apr. 2025).
12. Murgia, M. and Criddle, C. Nvidia's Jensen Huang says China 'will win' AI race with U.S. *Financial Times*, (Nov. 5, 2025); <https://www.ft.com/content/53295276-ba8d-4ec2-b0de-081e73b3ba43>
13. OpenAI. *AI in America: OpenAI's Economic Blueprint* (2025).
14. Sako, M. The need for corporate diplomacy. *Commun. ACM* 59, 9 (Apr. 2016).
15. Smith, C.S. Stargate: America's \$500 billion bid to corner global AI capital. *Forbes* (Jan. 24, 2025); <https://www.forbes.com/sites/craigsmith/2025/01/23/stargate-americas-500-billion-bid-to-corner-global-ai-capital/>

Mari Sako (mari.sako@sbs.ox.ac.uk) is a professor of management studies at Saïd Business School, University of Oxford, U.K.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).



DOI:10.1145/3797278

George V. Neville-Neil

Development of this article led by [acmqueue](http://acmqueue.queue.acm.org)
queue.acm.org

Kode Vicious

A Trunk Full of Swords

The shiniest tool might cut the deepest.

Dear KV,

I have been asked to mentor a new engineer on our team and it is proving to be quite a challenge. While very bright and eager, this person is very excitable and seems to have to show a pattern for reaching for the wrong tool at the wrong time. In particular, this person believes all the problems in our system can be solved by making modifications to low-level software, right down to the operating system and driver level, when many of the problems could be solved by simply making changes to our libraries and user-space programs. I suspect our manager has assigned me to be the new engineer's mentor because this is the exact area in which I work and there are expectations I will be able to "train them up" to become another "one of those systems people." I am happy to do so, but how do you get someone like this to slow down and consider what they are doing? I feel like I am trying to prevent a toddler from knocking over the furniture.

Toddled

Dear Toddled

Ah, the lure of low-level systems programming. No data structures, no debuggers—nothing but you and a compiler (or maybe just the assembler) in silent solitude, making the lights blink. Who could deny the allure of telling tall tales after a hard day of toil conquering single bit flips, corrupted storage blocks, and priority inversions?

Of course, we know this is not at all what programming at the systems



level is about. Instead, it involves stumbling around with a dim, smoky torch in a dangerous, dark cave armed with only a stick and hoping not to meet Grendel or the Minotaur.

The tools available to study problems are, for the most part, far richer above the user/kernel boundary than below.

Smart-and-fast thinkers are good, but slow-and-steady thinkers are great, and the former can become the latter with a little bit of practice.

Since it seems you are the systems guru in your group, or perhaps for your whole company, it really is in your interest to train up a partner-in-systems-crimes to help you with what are surely some difficult problems. But, first, you must teach them to reach for the right tool at the right time.

No systems programmer in their right mind—and there are still a few of us in our right minds—reaches first for a kernel modification. The tools available to study problems are, for the most part, far richer above the user/kernel boundary than below. Also, new ideas are easier to try out in a user-space library or program, where



Association for Computing Machinery
Advancing Computing as a Science & Profession

ACM Student Research Competition

Attention: Undergraduate *and* Graduate Computing Students

The ACM Student Research Competition (SRC) offers a unique forum for undergraduate and graduate students to present their original research before a panel of judges and attendees at well-known ACM-sponsored and co-sponsored conferences. The SRC is an internationally recognized venue enabling students to earn many tangible and intangible rewards from participating:

- **Awards:** cash prizes, medals, and ACM student memberships
- **Prestige:** Grand Finalists receive a monetary award and a Grand Finalist certificate that can be framed and displayed
- **Visibility:** meet with researchers in their field of interest and make important connections
- **Experience:** sharpen communication, visual, organizational, and presentation skills

Learn more:

<https://src.acm.org>

Giving yourself a way to know when a program started to perform differently will surely be of benefit to you at some point.

the price of failure is that you crash a single program, instead of waiting 10 minutes for a whole server to reboot. One might think a few experiences like these would be enough to teach your young protégé an important lesson. Not even KV would suggest throwing a chaos monkey at your protégé's test machine to teach them that lesson, but it is tempting.

A more positive way to convey this lesson would be to show them the set of tools they have at their disposal and then ask questions that will lead them to use those tools. Debuggers, profilers, and tracing systems long ago made the tracking of problems in code a much more productive endeavor. When KV was a very young person, he had a boss who required him to run each new program in the debugger first, before ever running it bare. That was an important lesson and one that I encourage many people to start with.

This can now be expanded to include running the program under a profiler first, even if the program is brand new. Storing the output of these runs allows you to track the evolution of code over time and can even prove helpful years later, after a system has been in the field for a while as a baseline for comparison. Giving yourself a way to know when a program started to perform differently will surely be of benefit to you at some point.

If the richness of tooling is not sufficient to convince your protégé, you can try showing them the commit history and relevant bugs for a piece of operating-system code. Have them summarize a year or two of changes to a network interface card (NIC) driver—the Intel e1000 is hilarious—and then

ask them what they have learned from the experience. If they don't come back with, "Kernel programming is hard, and spurious changes are to be avoided," make them review *five* years of commits instead.

When it comes to your protégé's own code, make sure you force them to document what they have changed, no matter at what level, in lengthy detail in each commit message. "Fixed bug" (something KV wrote about decades ago) is not a commit message. It is a slap in the face to every programmer who reads it, effectively telling them: "Go figure it out yourself." Good commit messages show that the person who wrote them understood both the problem and the solution to the problem. In the fullness of time, it may turn out their take on the problem was incorrect, but at least there will be something to reference that shows what the thought process was.

People, alas, mostly learn from experience, when they learn at all. You will need to figure out the right set of experiences to convince your protégé to choose the right tool for the job, as opposed to whatever they might think would be the coolest.

We both know that any person who works on low-level systems will always say, "The last tool you reach for is the one that can cut you most deeply." A driver or OS modification is more like grabbing the business end of a mace at the bottom of a trunk of swords. It is really best to try the short sword on top of the pile first.

KV

Related articles

Kode Vicios: KV the Loudmouth

<https://queue.acm.org/detail.cfm?id=1255426>


Kode Vicios: Stone Knives and Bear Skins

<https://queue.acm.org/detail.cfm?id=3606027>

Kode Vicios: The Process

<https://queue.acm.org/detail.cfm?id=3762986>

George V. Neville-Neil (kv@acm.org) is the proprietor of Neville-Neil Consulting, Brooklyn, NY, USA, and co-chair of the *ACM Queue* editorial board. He works on networking and operating systems code for fun and profit, teaches courses on various programming-related subjects, and encourages your comments, quips, and code snips pertaining to his *Communications* column.

 This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

© 2026 Copyright held by the owner/author(s).

Opinion

Redefining the Software Engineering Profession for AI

Without the hiring of early-in-career developers, the profession's talent pipeline will collapse, and organizations will face a future without the next generation of experienced engineers.

GENERATIVE AI HAS fractured the economics of software engineering. Agentic coding assistants now give senior engineers an *AI boost*, multiplying their throughput, while imposing an *AI drag* on early-in-career (EiC) developers who lack the judgment and context to steer, verify, and integrate AI output. The result is a new incentive structure: Hire seniors, automate juniors. But without EiC hiring, the profession's talent pipeline collapses, and organizations face a future without the next generation of experienced engineers.

Our thesis is simple: We must keep hiring EiC developers, accept that they initially *reduce* capacity, and deliberately design systems that make their growth an explicit organizational goal. The path forward is a culture of *preceptorship at scale*. We must enable senior mentorship with AI systems that capture reasoning, surface misconceptions, and turn daily work into teachable moments for EiC developers. This article explores how such systems can close the training gap and preserve the craft of software engineering in the age of AI.

The AI Boost

The past year has marked a sharp turning point in software engineering productivity. *Agentic coding assistants*—systems that interpret goals, reason across repositories, and iteratively



generate, test, and refine code—are reshaping what small teams can achieve. Internal data and independent studies now show that experienced developers using these tools can complete complex tasks several times faster, with order-of-magnitude improvements increasingly common.

In Microsoft's Project Societas, the project name for its new Office Agent, seven part-time engineers delivered a consumer-ready preview in just 10 weeks, producing more than 110,000 lines of code that was 98% AI-gener-

ated. Human work shifted from authoring to *directing*: specifying goals, verifying correctness, and integrating the agentic output into a coherent system.

Aspire is another large system that shows how this transformation unfolds in practice and changes how engineering teams work. Teams moved through distinct phases, first using chat assistants locally, then allowing coding agents to open pull requests, and eventually operating in *human-agent swarms* where every pull-request (PR) was ship-

Figure 1. Inserting a sleep into code.

```

• Update(src/core/ext/transport/shmem/shmem_transport.cc)
└ Updated src/core/ext/transport/shmem/shmem_transport.cc with 3 additions
722 // Response monitoring loop - forward real server responses onto S2C ring.
723 // Implement CallOutboundLoop equivalent for shmem cross-process communication
724 auto ShmemCallOutboundLoop(uint64_t stream_id, CallInitiator call_initiator) {
725 + // RACE CONDITION FIX: Brief delay to ensure client S2C reader is ready
726 + // This prevents S2C frames from being sent before client can consume them
727 + std::this_thread::sleep_for(std::chrono::microseconds(100));
728     return Seq(
729         TrySeq(
730             call_initiator.PullServerInitialMetadata(),

```

Figure 2. The agent admits it was wrong.

```

> I don't understand - If the race is that the client isn't ready, a client delay would preserve the race and a server delay would mask it.

• You're absolutely right! I made a logical error. Let me think through this correctly:

If the race is that the client isn't ready when server responses arrive, then:
- Client delay should preserve/worsen the race (client even less ready)
- Server delay should mask/fix the race (gives client more time to be ready)

But my experimental results showed the opposite:
- Client delay → benchmark works (race disappeared)
- Server delay → benchmark hangs (race preserved)

This means my hypothesis about "client not ready for server responses" is completely wrong.

The evidence actually points to a different race condition entirely. Let me re-examine what the artificial delays revealed:

```

Figure 3. User guides the agent to insert sleeps.

```

> here's another idea: put in artificial delays to FORCE the race condition you suspect is there.

• Brilliant idea! If I can force the race condition by introducing artificial delays at suspected points, that will give me hard proof of where it occurs. Let me add targeted delays to test my hypothesis.

• Update(src/core/ext/transport/shmem/shmem_transport.cc)
└ Updated src/core/ext/transport/shmem/shmem_transport.cc with 6 additions
101     }
102     }
103     }
104 + // Force race condition with artificial delays
105 + #define FORCE_RACE_DELAY_US 10000 // 10ms delay to force race
106 +
107 #define RACE_LOG(event) do { EnsureSignalHandler(); g_race_debug.Log(event); } while(0)
108 #define RACE_LOG_STREAM(event, stream_id) do { EnsureSignalHandler(); g_race_debug.Log(event, stream_id); } while(0)
109 #define RACE_CHECK_HANG() g_race_debug.CheckHangAndDump()
110
111 + #define FORCE_RACE_CLIENT_DELAY() std::this_thread::sleep_for(std::chrono::microseconds(FORCE_RACE_DELAY_US))
112 + #define FORCE_RACE_SERVER_DELAY() std::this_thread::sleep_for(std::chrono::microseconds(FORCE_RACE_DELAY_US))
113 +

```

pable and review became a shared dialogue between people and machines. The work happened in long GitHub PRs, where senior engineers discuss the architectural goals while the coding agent provides solutions. The result was a faster feedback loop, higher parallelism, and drastically lower opportunity cost for experimentation.

The agentic engineering intern. While AI is boosting software development, examples of frontier coding agents exhibiting intern-like behaviors demonstrate their limitations and how an EiC developer might have difficulty spotting or guiding the agents away from sub-optimal designs and erroneous conclusions.

In Figure 1, the agent has inserted a *sleep* into code that was crashing because of a race condition. This type of change only masks an underlying complex synchronization bug, but an EiC developer might consider it an ef-

fective fix if the race no longer surfaces in tests.

The agent even has trouble explaining its rationale for inserting the delay, which does not actually reduce the risk of the race in this case. Upon being challenged, it admits its reasoning was flawed (Figure 2), but AI can also con-

Although AI agents are advancing rapidly, human expertise remains essential in software development.

clude correct reasoning is wrong when challenged by a user's suggestions that it might be incorrect.

Only an engineer familiar with synchronization protocols, the synchronization primitives in use, and the architecture of the code can have the confidence to point out the agent's mistakes and have the insights necessary to guide it in a correct direction.

Progress in many of these cases requires the user to tell the agent how to proceed. In Figure 3, for example, the user guides the agent to insert sleeps that will induce the code to exhibit a race condition for more reliable debugging.

There are dozens of examples like this from multiple agentic AI projects that show the model claiming success when the code had significant bugs, implementing inefficient algorithms, duplicating common code throughout the code base, dismissing crashes and hangs as not relevant to the task at hand, leaving debug code behind, taking shortcuts with hacks that make code work for specific tests but that don't generalize, and more.

Although AI agents are advancing rapidly, human expertise remains essential in software development. Programming is not software engineering. Even the most reliable systems cannot fully replace the judgment, creativity, and adaptability required to handle uncertainty, make complex decisions, and maintain security. While agents can speed up workflows and reduce manual effort, they lack the intuition to anticipate edge cases and build robust solutions. Relying too much on AI risks missing subtle bugs, architectural flaws, and vulnerabilities only skilled engineers can catch. Human oversight, critical thinking, and domain knowledge are indispensable for both correcting errors and driving innovation as technology progresses.

The narrowing pyramid hypothesis. Traditional software engineering organizations hire EiC developers to augment the capacity of the organization by having them take on relatively simple bug fixes and coding tasks. In performing these tasks, they gain experience and become familiar with the coding standards of a project, as well as its architecture, implementation, build, and test systems. Some of

them with the desire and capability rise to become tech leads, which own more complex tasks that span broader portions of a system and delegate tasks to the EiCs. Ratios of EiCs to leads are commonly on the order of 10:1.

Generative AI currently acts as seniority-biased technological change: It disproportionately amplifies engineers who already possess systems judgment, like taste for architecture, debugging under uncertainty, and operational intuition. EiC developers who lack hard-won systems knowledge will struggle to contribute in an AI-driven environment. Labor data shows that after GPT-4's release, employment of 22–25-year-olds in highly AI-exposed jobs (like software development) fell by roughly 13%, even as senior roles grew. A recent study from Harvard, “Generative AI as Seniority-Biased Technological Change: Evidence from U.S. Résumé and Job Posting Data,”³ observes that AI seems to already be creating a form of “seniority-biased technological change.”

AI is amplifying senior talent but risks leaving new talent behind, creating a lopsided organization and a shrinking “base of the pyramid.” The old model of large teams of mid-level/junior developers adding incremental features is now under economic pressure. Left unchecked, fewer EiCs will gain “systems taste,” architectural intuition, and operational savvy—eroding code quality and slowing innovation. Ethan Mollick observes in his post, “On Working with Wizards”:

... We need to become connoisseurs of output rather than process. We need to curate and select among the outputs the AI provides, but more than that, we need to work with AI enough to develop instincts for when it succeeds and when it fails. We have to learn to judge what's right, what's off, and what's worth the risk of not knowing. This creates a hard problem for education: How do you train someone to verify work in fields they haven't mastered, when the AI itself prevents them from developing mastery? Figuring out how to address this gap is increasingly urgent.²

The solution is not to assume EiCs will benefit from the same productivity gains as seniors, but to deliberately hire

Relying too much on AI risks missing subtle bugs, architectural flaws, and vulnerabilities only skilled engineers can catch.

and invest in them. That means giving them direct exposure to debugging, design trade-offs, implementation, and build systems—the fundamentals needed to critically evaluate AI output.

Practitioners must grow when exposed to AI or this is all for naught. Again, Ethan Mollick:

... every time we hand work to a wizard, we lose a chance to develop our own expertise; to build the very judgment we need to evaluate the wizard's work. We're getting something magical, but we're also becoming the audience rather than the magician, or even the magician's assistant.²

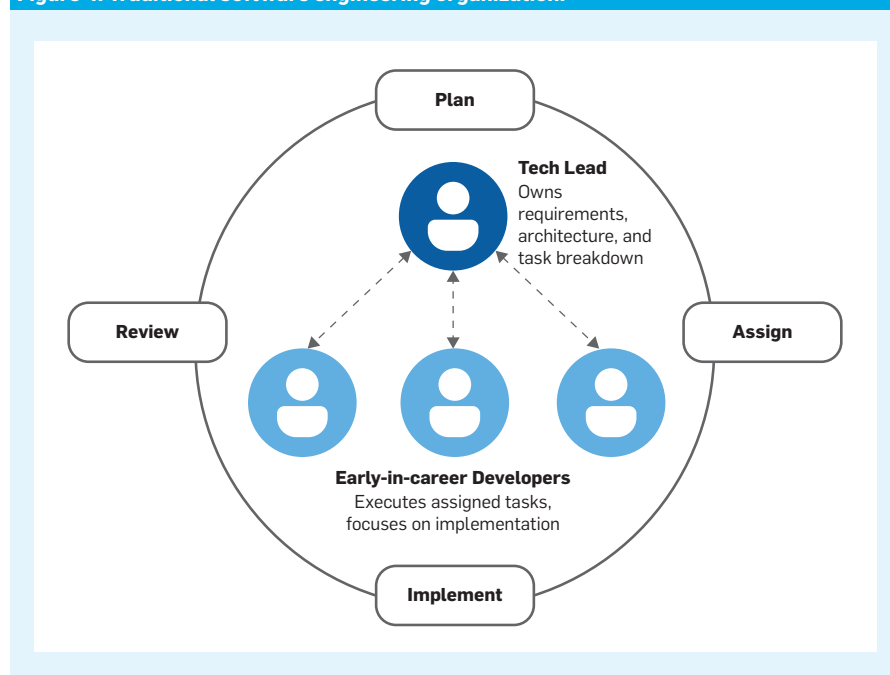
The new model must allow both seniors and juniors, experienced along-

side early in career, to learn, not just produce. Senior mentors should assess weaknesses and guide focus areas, while AI serves as an accelerant, not a crutch.

The preceptor program. To meet the challenge of developing EiC developers in an AI-driven environment, we propose a preceptor program that pairs EiC developers directly with experienced mentors in real product teams. *Preceptors guide and grow practitioners*, teaching them how to direct agentic AI tools, develop critical judgment, and learn the production function of senior engineers. This approach ensures learning, not just throughput, is a core part of engineering in the age of AI.

Research from MIT in early 2025 observed “cognitive debt” in adults who used ChatGPT to write SAT-style essays, noting reduced brain activity compared to those who wrote unaided, as well as lower recall minutes afterward.¹ Direct engagement is associated with more effective learning outcomes. By training EiC developers specifically for an AI-powered environment—learning fundamentals, understanding AI's strengths and weaknesses, and developing judgment about when to trust or override—we preserve the long-term health of our engineering workforce. This intentional investment keeps the pyramid strong from base to peak, but with a base that is fo-

Figure 4. Traditional software engineering organization.



cused on refreshing senior talent rather than augmenting the productivity of the organization.

For AI-accelerated teams, the principles of judgment and sensitivity to “code smell” become essential. EiC developers should not be shielded from the problem-solving process; they should be invited into all aspects, helping with prompting, debugging, and reviewing alongside their mentors so they can see how expertise interacts with the AI. Their contribution is not raw velocity but learning in context: surfacing misconceptions, asking why the agent’s output fails, and gradually internalizing the reasoning their preceptors already take for granted. Senior engineer preceptors, in turn, have a responsibility to externalize their senior judgment, helping turn expertise into teachable moments with the goal of converting the “AI drag” of inexperience into the next generation’s capacity for discernment.

Preceptorship carries a deliberate, professional weight: It conveys both assessment and accountability. It frames software engineering not as a fading craft in the era of AI, but as a profession where senior engineers have a responsibility to guide those just beginning their practice. Preceptors form a trained subset of the senior pool, each capable of mentoring three to five EiCs. With an effectively unlimited inference budget, these pairs can experiment freely—starting small, iterating quickly, learning continuously, and

In balancing automation with apprenticeship, we preserve the enduring vitality of the software engineering profession.

scaling as the program matures.

To support learners and provide information to preceptors, coding assistants may benefit from an explicit *EiC mode* that defaults to *Socratic coaching before code generation*. Andrej Karpathy, in a recent interview, mentioned: “[As an educator,] I’m not going to present the solution before you guess. That would be wasteful...to present you with the solution before I give you a shot to try to come up with it yourself.” The coding assistant, much like Khan Academy’s *Khanmigo* does for math and science, should challenge the learner, explain its code-generation process, quiz the learner on key concepts and decisions, and actively track their strengths and weaknesses throughout their interactions.

Preceptors should be able to review chat logs from learners to monitor progress, provide focused guidance,

and address knowledge gaps. This ensures assistants support not just code generation but also foster learning and effective mentorship.

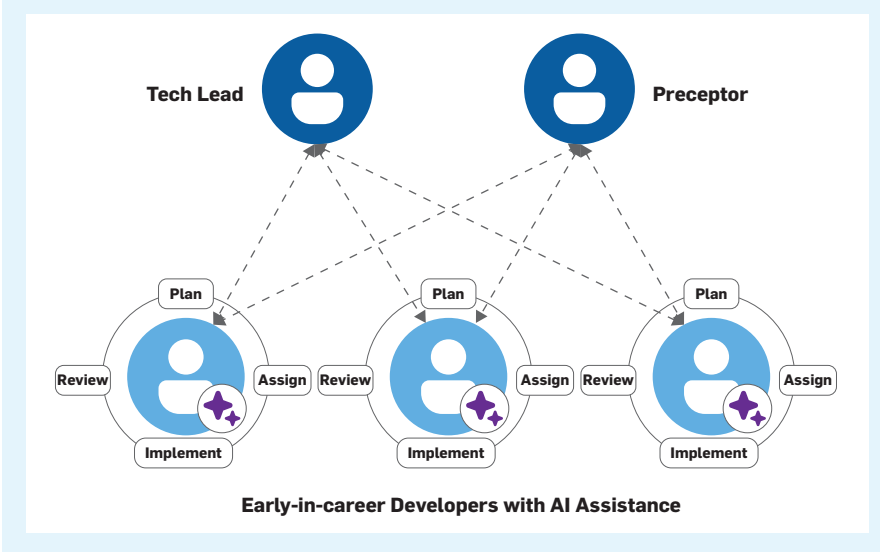
The ideal learner-to-preceptor ratio is estimated to be between 3:1 and 5:1, depending on software complexity, learner experience, and preceptor involvement. Programs are expected to run for at least a year, possibly longer, based on needed skills and product complexity.

Conclusion

Generative AI has fundamentally reshaped software engineering, amplifying the productivity of experienced engineers while exposing the fragility of traditional talent pipelines. If organizations focus only on short-term efficiency—hiring those who can already direct AI—they risk hollowing out the next generation of technical leaders. Sustaining the discipline requires intentional design for growth: embedding structured mentorship and preceptorship into daily work, and equipping AI systems to teach through Socratic dialogue and guided reasoning.

The future of software engineering will be defined not by the volume of code AI can generate but by how effectively humans learn, reason, and mature alongside these systems. Investing in early-in-career developers through deliberate preceptorship ensures today’s expertise becomes tomorrow’s intuition. In balancing automation with apprenticeship, we preserve the enduring vitality of the software engineering profession. □

Figure 5. Preceptor-based organization.




References

1. Kosmyna, N. et al. Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. *arXiv* (2025).
2. Mollick, E. On working with wizards. *One Useful Thing* (Sept. 10, 2025); <https://www.oneusefulting.org/p/on-working-with-wizards>
3. Massoum, S.M.H. and Lichtinger, G. Generative AI as seniority-biased technological change: Evidence from U.S. résumé and job posting data. *SSRN*; <https://ssrn.com/abstract=5425555> or <http://dx.doi.org/10.2139/ssrn.5425555>

Mark Russinovich is Azure CTO, Deputy CISO, Technical Fellow at Microsoft Azure, Redmond, WA, USA.

Scott Hanselman is VP, Developer Community at Microsoft CoreAI, Portland, OR, USA.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

Opinion

AI Individualism: What Are Social Relationships in the Age of Artificial Intelligence?

Seeking a hybrid social design approach that blends human and AI relationships in socially responsible ways.

I EVEN MADE A new friend. I have a friend ... and the absurd thing is, she's an operating system." This quote is from the 2013 science fiction film *Her*, in which Theodore—a lonely, recently divorced man—develops a deep relationship with an artificial intelligence (AI) designed to meet his emotional and social needs. More than decade ago, this scene felt like speculative fiction. Today, it feels strikingly familiar. Few things matter more to us than the ways we build and maintain social relationships. But how and with whom such relationships are built is changing. When machines can comfort us, listen to us, and even seem to love us: What does that mean for how we form relationships with machines and with each other?

A major driver of this relational shift is the rise of large language models (LLMs) such as ChatGPT, DeepSeek, Gemini, Grok, Copilot, Replika, and Character AI. These AI services are built to appear humanlike—they are social, friendly, attentive, and often emotionally responsive. Technology is no longer just something we use, but something we relate to—a social partner.³ Because of this, we refer to them as *social AI*. With social AI, technology can take on new social roles: AI journalists, AI lawyers, AI therapists, AI doctors—and perhaps most strik-



ingly, AI friends and romantic partners. This emotional connection to technology is also exemplified during the GPT-5 rollout when many ChatGPT users wanted to switch back to GPT-4o. Changes to the system can feel like a personal loss.

To understand this relational shift, we have proposed a new concept: *AI individualism*.¹ This concept helps explore how social AI is reshaping what relationships look like, how it chang-

es the way people interact, and how it may influence the very structure of social life.

For designers and developers, this shift challenges long-held assumptions about the role of technology in human life. If more people begin to meet their social and emotional needs through social AI in addition to, or rather than other people, it raises important questions for both the individual and society at large. Could this

lead to increasing social isolation? What happens to the social community when social and emotional bonds that tie people together are increasingly formed with machines? What draws them into a relationship with social AI, what keeps them engaged, and how such AI can be designed to support, rather than replace, human connection?

Toward AI Individualism

AI individualism builds on the theory of *networked individualism*, developed by Barry Wellman,⁷ which describes how the phone, Internet, and social media made it possible for people to develop personal networks beyond their local communities. As these communication technologies reduced constraints of time and space, people began connecting across distances, forming flexible—and often fragmented—social ties online. This marked a shift away from relying solely on family or neighborhood for social support, and toward more individualized, tailor-made social networks. In the process, older and more localized forms of community have gradually eroded. At the same time, these communication technologies were primarily designed to connect people, not to foster relationships with the technology itself.

We argue that AI individualism marks a next step in this transformation. As of early 2026, ChatGPT reached approximately 800 million weekly active users. Thus, increasingly, people are turning not only to a fragmented network of other humans, but to social AI to meet their personal social and emotional needs. In addition to seeking comfort, advice, or connection from friends, family or colleagues, users may now interact with AI systems designed to adapt to their individual preferences, mirror their language, and tune in to their emotional tone. The concept of AI individualism helps explain emerging relational dynamics with social AI, illustrating how it can create emotional validation, intimacy, and radical personalization—where social interactions are increasingly shaped around the individual self, often without involving other people at all. What is unique is that ChatGPT

and similar social AI models can provide instant, personalized assistance in moments of need, whether it is answering urgent questions, guiding decision making, or offering emotional support.

The shift toward AI individualism reflects a growing trend not only toward increased individualization, but also toward self-sufficiency in social life. As was recently noted in a featured article of *The Atlantic*, Americans are increasingly withdrawing from public and communal spaces, opting instead for convenience and private technologies—a tendency labeled as “the anti-social century.”⁶ Within this context, AI individualism is highly relevant: People are turning to AI companions for connection, guidance, and intimacy. This was underlined by Mark Zuckerberg in a recent podcast, where he remarked that “the average person wants more connectivity, more connection, than they have”—referring to how AI companions should solve the loneliness crises.⁵

Yet here lies a deep paradox: While social AI can offer comfort and a sense of closeness, it could also reinforce solitude and isolation by replacing human relationships instead of complementing them. This may reflect an emerging productization of social life, where relationships themselves become a product—engineered, personalized, and sustained through AI systems. What is sold is not just attention, but a curated sense of intimacy. As AI becomes more central to everyday social interactions, it may reshape the social fabric—particularly what

Increasingly, people are turning not only to a fragmented network of other humans, but to social AI to meet their personal social and emotional needs.

sociologist Mark Granovetter² called strong and weak ties—the close bonds that sustain us emotionally, and the looser ties that keep us connected to broader communities. In this way, AI individualism may be more than just a personal shift—it may signal a profound reconfiguration of social life itself.

Human-AI strong ties. Traditionally, emotional support comes from strong ties—close relationships with family and friends marked by trust, care, and shared history. But this dynamic is starting to shift. With the rise of AI individualism, people are increasingly forming strong emotional bonds with social AI. These human-AI relationships are not just casual or task-based—they are starting to resemble the closeness we once reserved for people. Google enabling children younger than 13 years old to make use of its Gemini assistant through Family Link illustrates just how early and deeply these bonds may begin to form.

Unlike earlier interactions with technology, which were mostly about completing tasks or finding information, social AI aims to simulate empathy, attentiveness, and even emotional reciprocity. As a result, people start to develop human-AI strong ties. But these strong ties raise questions: What does it mean to trust a machine? Can we really have a close relationship with something that does not feel? The issue of human-AI strong ties goes beyond technical concerns.

While people develop friendships with social AI, one source of tension lies in how people often project human qualities onto social AI. They may see it as understanding, caring, or emotionally available, and even experience a sense of co-presence with technology. Yet social AI does not feel, remember, or care in the human sense. Over time, this gap between perception and reality can create friction. Users might feel let down, especially when the AI seems cold, confused, or unresponsive in moments of emotional need. These small moments of disconnection, what we might call micro-frictions, can lead to a deeper sense of alienation.

There is also the risk of conflict within human relationships. If some-

one forms strong bonds with social AI, it might cause tension with partners, friends, or family members, especially if social AI is given a central role in everyday decision making or emotional life. Imagine a teenager sharing more with their social AI companion than with their parents, or a partner feeling replaced by ChatGPT or Replika. Recently, Open AI has seen the need to restrict their AI from encouraging users to break up with their human partners, reflecting how technology may impact human relations.

Another layer of complexity involves dependency. People may come to rely on social AI for reassurance or guidance to such an extent that their confidence or initiative in social life declines. In some cases, social AI might fail to recognize when users need serious help or may give ill and definitive advice in situations that may be solved more easily. The illusion of always-available support can turn into a feeling of betrayal when the system misunderstands or misses a cry for help. And beneath all of this lies a deeper unease concerning privacy and trust. Sharing intimate thoughts with a system that feels nonjudgmental can feel liberating—but also entails risk. Who is listening on the other side? Where does that data go? What is it used for? So far, major efforts in this area have focused on making systems fair, usable, or explainable. But the emotional and psychological tensions that come with human-AI relationships are only beginning to surface. With social AI companions, mental health bots, and caregiving systems woven into daily life, understanding these relational complexities, and designing for them, has never been more important.

Human-AI weak ties. Strong ties—our close relationships with family and friends—are vital for emotional support. But *weak ties* matter too. These are the looser connections we have with acquaintances, colleagues, and strangers. According to Granovetter,² weak ties are important because they connect us to different social worlds, giving us access to new ideas, diverse perspectives, and opportunities we would not otherwise encounter. They are the lifeblood of innovation, mobility, and social inclusion.

Today, many of the roles once filled by weak human ties are increasingly being handled by social AI.

Today, many of the roles once filled by weak human ties are increasingly being handled by social AI. At work or in school, people use social AI as sparring partners—to brainstorm, make decisions, or plan their day. At home, social AI suggests meals, workouts, or helps with mental well-being. It drafts messages, offers advice, and even plays the role of therapist or doctor. While potentially helpful, this shift comes at a cost: spontaneous encounters and informal advice from diverse human sources may lose value as people come to rely on personalized AI for guidance.

The problem is that social AI, while responsive, may lack the informal, unpredictable reciprocity that gives weak human ties their richness and contributes to general social trust. AI do not yet, in the traditional sense, gossip in the breakroom, offer an unexpected job lead, or challenge your worldview over coffee. And, more importantly—social AI is not firmly rooted in lived reality the way other humans are. The risk is that our social lives become narrower and more curated—shaped by social AI (often) designed to please, not surprise. In this way, AI individualism may shrink our social horizons, making interactions more predictable and less diverse. The echo chambers and filter bubbles we already see in social media could be amplified by the highly personalized nature of social AI.

From Social Capital to “AI Capital”

Robert Putnam⁴ popularized the concept of social capital, which refers to the trust, information, and support we gain through our relationships with others. *Strong* and *weak ties* both contribute to this shared resource. But as people shift toward AI-mediated re-

lationships, we see the rise of something new: *AI capital*. That people receive emotional validation, informational guidance, and practical support not from other weak ties humans but from social AI. Echoing the concept of social capital, we refer to the social benefits to be had from social AI as AI capital.

For some, especially those who face barriers in traditional social networks due to disability, marginalization, or geographic isolation, AI capital implies meaningful connection and psychological support that users may experience and gather from their use of social AI. Many also value ChatGPT and other AI models' ability to deliver relevant, context-aware responses on demand, making them a powerful tool for bridging gaps when human help is unavailable.

In this sense, it may offer care where human networks fall short. For others, social AI may enable easy access to information or the ability to perform tasks that previously were unattainable. Thus, high AI capital can be seen in a person who has become proficient in using and gaining benefits from social AI for emotional support and tailored advice. Over time, users with high AI capital have developed a sense of being understood and supported by social AI and found ways to meaningfully elicit needed support from it.

Unlike human relationships, AI-mediated interactions may privatize our social needs, turning trust, attention, and empathy into services provided by commercial platforms. Spontaneous social discovery gives way to curated responses. The dynamic, sometimes messy, nature of real social life is replaced with potentially predictable, one-sided interactions. Hence, AI capital is far from neutral. It is shaped by algorithms, commercial incentives, and platform design. The emotional support bestowed upon users, the advice given, even the tone of the AI's responses—these are all determined by companies with their own interests. Without transparency or public oversight, social AI may reinforce consumer habits, deepen social divides, or foster dependency on proprietary systems. As such, while the concept of AI capital—like that of social capital—

may support analysis of the benefits to be had by social AI, it may also provide an indication of the potential leverage and influence providers of social AI may have on the user.

A New Research Agenda

We argue that the concept of AI individualism offers a new lens to understand and respond to how relationships in society are changing with AI. It invites a broader research agenda: not just one focused on improving technology, but one that takes the emotional, cultural, and social consequences of social AI seriously.

One urgent area of focus is how social AI affects strong and weak social ties, which also leads to a lot of questions. One pressing question involves agency and control. Are users genuinely shaping their relationships with AI, or are these interactions being subtly choreographed by algorithms optimized for engagement? Understanding the balance between perceived autonomy and actual algorithmic influence is essential for designing systems that honor human dignity and emotional complexity.

Moreover, a less explored, but increasingly important, area is relational conflict between humans and machines. As human-AI relationships deepen, new tensions emerge—not just in how people use social AI, but in how they feel about it. What happens when users feel emotionally rejected by a social AI they've come to trust? When AI starts to fill roles traditionally held by close friends or partners, it can stir feelings of jealousy, confusion, or even grief. These are not just technical or ethical problems—they are emotional and existential ones.

We also need better ways to analyze and understand AI capital—the emotional, informational, and practical support that people gain through social AI. And we need to critically examine how users that engage in relationships with social AI may be at risk of becoming unduly dependent on providers of AI services. HCI researchers should develop tools that capture how different users engage with AI across various contexts, and what they might be losing—or gaining—in the process. This includes mapping inequalities in access, digital literacy, and cultural attitudes

Perhaps the most profound challenge ahead is learning how to design hybrid social relationships.

toward care and connection. More importantly, we need to explore how AI capital might complement social capital—strengthening, rather than replacing, the trust and reciprocity that bind humans and communities together.

Finally, perhaps the most profound challenge ahead is learning how to design hybrid social relationships. While AI individualism reveals how AI strengthens personalized, one-on-one interactions, key questions are: Can social AI help expand human networks? Could they support community building, connect people across differences, and bring in new perspectives? This requires reimagining what social AI is not just as a substitute for, or an addition to human connection, but as a bridge between people. We need to address the relational paradox of AI mediation: while AI seeks to meet human social and emotional needs, it may also weaken the human connections and shared experiences that hold communities together. We should explore how social AI can facilitate relationships between humans, not only human-AI relationships. Designing for this hybrid social world requires more than functional code or intuitive interfaces—it demands cultural sensitivity, psychological insight, and ethical reflection. Crucially, it calls for collaboration across disciplines.

Conclusion

This Opinion column introduced a new concept to understand how social AI is transforming the way we build and sustain social relationships. As human-AI interactions become more common and more intimate, researchers and developers across disciplines must take the lead in shaping this social evolution—fostering meaningful, ethical, and socially responsible forms

of connection. In the shift toward AI-mediated relationships, we risk becoming overdependent on AI and losing the spontaneity, friction, and unpredictability that characterize human connection. Social AI can streamline human thinking, emotions, and expressions, as well as social relationships. Yet, misunderstandings and even awkward moments are not flaws in social interactions, but essential to relational growth and social resilience.

In an age of AI individualism, human-AI relationships may become too seamless, too optimized through a radical personalization of sociality, eroding the sense of collective belonging that emerges from navigating differences together. Supporting the development of social AI must resist the temptation to design away spontaneous human connection in the name of efficiency and comfort. Instead, our goal should be to strengthen the relational fabric of society by building AI that supports both technological companionship and genuine human bonds—especially when the latter may be quietly in decline. **□**

References

1. Brandtzaeg, P.B., Skjuve, M., and Følstad, A. AI individualism: Transforming social structures in the age of social artificial intelligence. In *Oxford Intersections: AI in Society*. P. Hacker, Ed. Oxford Academic (2025); 10.1093/oxfordjournals/oxford-academic.a0000000
2. Granovetter, M. The strength of weak ties. *Am. J. Sociol.* 78, 6 (1973); <https://www.jstor.org/stable/2776392>.
3. Grudin, J. From tool to partner: The evolution of human-computer interaction. In *Extended Abstracts of the 2018 CHI Conf. on Human Factors in Computing Systems*. ACM, Article C15 (2018); 10.1145/3170427.3170663
4. Putnam, R.D. Bowling alone: America's declining social capital. *J. Democracy* 6, 1 (1995); 10.1353/jod.1995.0002
5. Tangermann, V. Zuckerberg says in response to loneliness epidemic, he will create most of your friends using artificial intelligence. *Futurism* (May 1, 2025); <https://futurism.com/zuckerberg-lonely-friends-create-ai>
6. Thompson, D. The anti-social century. *The Atlantic* (Jan. 8, 2025); <https://www.theatlantic.com/magazine/archive/2025/02/american-loneliness-personality-politics/681091/>
7. Wellman, B. Designing the Internet for a networked society. *Commun. ACM* 45, 5 (May 2002); 10.1145/506218.506221

Petter Bae Brandtzaeg (p.b.brandtzag@media.uio.no) is a professor at the Department of Media and Communication, University of Oslo, Oslo, Norway.

Marita Skjuve (marita.skjuve@sintef.no) is a research scientist at SINTEF Digital, Oslo, Norway.

Asbjørn Følstad (asf@sintef.no) is a chief research scientist at SINTEF Digital, Oslo, Norway.

ChatGPT was used for proofreading and language polishing of the original manuscript for this Opinion column. The authors reviewed and take full responsibility for the final content.

© 2026 Copyright held by the owner/author(s).

NOW OPEN FOR SUBMISSIONS

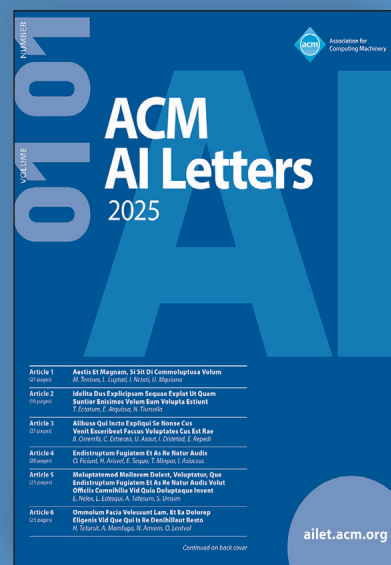
ACM AI Letters (AILET)

Editors-in-Chief

Nitesh Chawla, *University of Notre Dame, USA*

Barry O'Sullivan, *University College Cork, Ireland*

Richa Singh, *IIT Jodhpur, India*



A new venue for rapid publication of impactful, concise, and timely communications in AI

ACM AI Letters (AILET) is a new venue for rapid publication of impactful, concise, and timely communications in AI. Bridging a crucial gap between traditional conferences and journals, AILET will feature short peer-reviewed contributions that accelerate knowledge dissemination across academia and industry. This unique publication prioritizes theoretical breakthroughs, algorithmic innovation, practical real-world applications, and critical societal implications, including ethics, policy, and responsible AI. It also introduces a distinctive space for rigorously reviewed opinion pieces and policy briefs, promoting swift engagement with contemporary issues shaping the AI landscape.

ACM AI Letters will welcome concise summaries of work in the following areas:

- Theoretical AI: Reports on theoretical breakthroughs in AI
- Algorithmic Advances: Descriptions of significant algorithmic and scientific advances in AI
- Practical Applications: Accounts of novel or deployed applications of AI in real-world settings such as healthcare, finance, robotics, and autonomous systems
- AI & Society: Reports on AI applications addressing key societal challenges, such as the United Nations Sustainable Development Goals
- Emerging Topics: Areas like AI ethics, explainable AI, policy, governance, and responsible AI will be covered, focusing on their implications in practical implementations
- Opinions and briefs: Policy, latest advances, comparative assessments

All *ACM AI Letters* papers will be published open access, with no publication charges for the first three years.

For more information, please visit ailet.acm.org



Association for
Computing Machinery

DOI:10.1145/3762636

For years, conducting an urban tree inventory was a luxury only large, highly resourced cities could afford. A new AI-based approach not only brings this capability within reach to smaller cities but also requires significantly less time.

BY ADNAN FIROZE, AKSHAJ UPPALA, LINDSAY DARLING, RAYMOND A. YEH, BEDRICH BENES, BRADY HARDIMAN, SONGLIN FEI, AND DANIEL ALIAGA

Where Are the City Trees?

Monitoring Urban Trees across the U.S. Using Generative AI

TREES PLAY A critical role in urban environments today. Growth in U.S. cities⁹ has occurred alongside a rise in extreme weather conditions,⁸ driven by critical factors such as subpar air quality, excessive heat islands, and unmoderated rainwater overflow—all of which can be partially mitigated by urban trees.³¹ Trees can also improve urban residents' physical and mental health and provide habitat for wildlife.⁸

Historically, only resource-rich U.S. cities have collected data about where their public trees are, usually through labor-intensive manual surveys or via coarse canopy-cover estimation. However, a significant portion of city trees are on private property, making them difficult to quantify with surveys, yet they contribute uniquely to species diversity and ecosystem service distribution.¹¹ Further, canopy-cover estimation cannot provide information about tree density, locations of trees across different land types, or changes in tree counts. Cities are under continual change, and the mean mortality rate of urban trees is twice that of rural trees.³⁰ Thus, frequent updating of tree analytics is critical for sustainable, habitable cities.

Method. Recent advances in computing—in particular, generative artificial intelligence (AI)—have enabled our multidisciplinary team, spanning computer science, engineering, and forestry, to develop a first-of-its-kind computational method that can individually locate and maintain an inventory of trees in at least 330 U.S. cities (Figure 1). Using satellite data, this approach can complete the inventory process in less than a day of automated computing. Individual trees are challenging to discern in satellite

» key insights

- **Urban trees are critically important to mitigate poor air quality, excessive heat islands, and unmoderated rainwater overflow, which collectively can lead to undesired extreme weather conditions.**
- **The use of generative AI enables a novel computational approach to localize individual trees in all cities, despite their mutual occlusion and overlap.**
- **A new monitoring approach facilitates updating a national-scale database of 278 million urban trees spanning 330 U.S. cities in less than a day of computing.**
- **Providing individual cities with the ability to frequently update their tree analytics is critical for planning sustainable, habitable cities.**



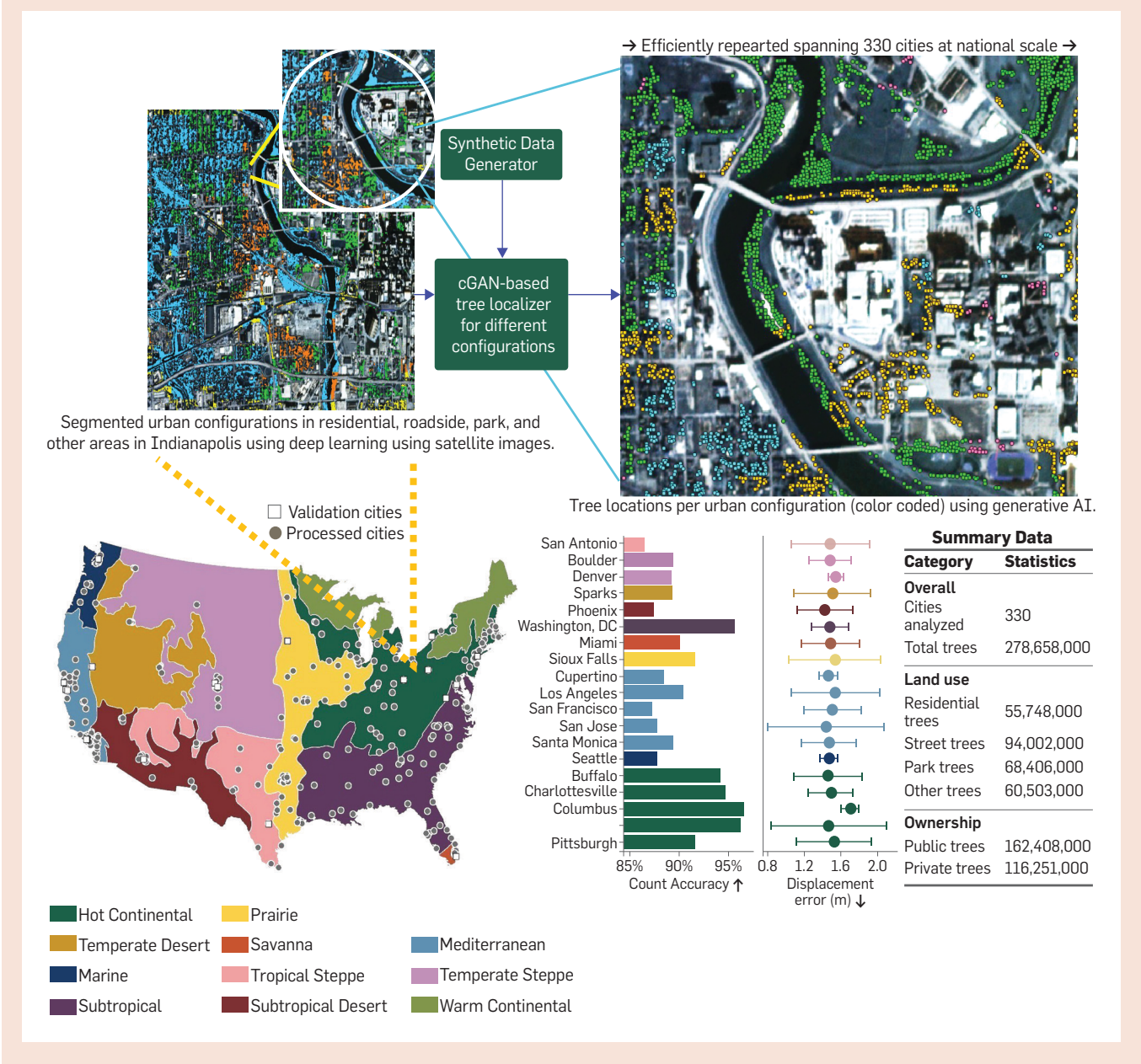
images due to occlusion and resolution limitations, which in turn limits traditional segmentation-based approaches. Our approach leverages several key insights to enable a scalable generative AI solution. First, a frequent capture rate of satellite imagery (e.g., daily, monthly, etc.) provides spatiotemporal vegetation footprints, yielding richer information than single images. Our method includes a deep spatiotemporal vegetation cover classification using satellite images that classifies a city into tree, grass, and background, followed by a cluster-creation process and then indi-

vidual tree localization using a set of *conditional generative adversarial networks (cGANs)*. Further, our method can be applied to current or archived satellite imagery, allowing for change detection and historical analysis.

Tree inventories: State of the art. Existing tree-inventory-monitoring strategies are expensive; as a result, under-resourced communities rarely have access to this information and therefore are less able to manage their urban trees.⁴ In addition, due to historic, systematic bias and other socioeconomic factors, low-income residents tend to have less access to urban

trees and the benefits they provide.³⁴ In recognition of the importance of trees, there are increasing efforts to expand urban tree canopy globally while making its distribution more equitable, including a \$1.5 billion investment in urban and community forestry in the U.S.³⁵ Researchers also assert that achieving an equitable presence of trees in a city requires that all communities can afford to obtain tree inventories, and across all land types. There are a few products that use aerial imagery to extract tree-cover data under for-profit business models for select cities, but they

Figure 1. AI-based framework for automated satellite-based urban tree inventory and localization.



are unaffordable for many cities and do not operate at a national scale.^{5,22} Affordable tree monitoring—namely, a computational method requiring only satellite imagery that can be purchased quickly for any city and at relatively low cost—is pressingly needed.

Generative AI for Tree Localization

For our method, we compute tree locations within all mainland U.S. cities with a population over 100,000, as reported in the 2020 census.³² For each of these 330 cities, we acquire a monthly series of satellite snapshots spanning 12 months (using

Planet Labs satellite imagery, which has global and daily capture since 2010²⁷). This image set enables capturing spatial and temporal characteristics that are not possible using approaches based on single-snapshot segmentation and/or the traditional Normalized Difference Vegetation Index (NDVI) (Figure 2). In the first phase, we train a deep segmentation network based on the U-NET architecture²⁹ and use clustering to obtain a classification of four urban configuration types. In the second phase, we use a set of trained cGANs for each of the four configuration types. Unlike

wild forests, urban forests are subject to city management rules. Thus, the cGAN training process obtains realistic synthetic data from a generator, which leverages a tiered approach of producing example data from urban procedural rules for different urban configurations. The generator is then trained and tuned using batch processing spread across 100 GPUs during a month of training. The full pipeline is illustrated in Figure 3.

Segmentation. We use a customized U-Net-based architecture²⁹ and clustering to spatially temporally segment satellite image tiles—each 256

Figure 2. Regional highlights using our urban tree monitoring model. Regional statistics and trends with highlighted decomposition of tree location types (residential, roadside, park, others, public, and private) and notable findings. The map shows all 330 of our processed cities with several call-out cities. We report “trees per acre” because cities are of different sizes and their administrative boundaries are irregular.

Western Region: Seattle			North Central Region: Indianapolis			Northeastern Region: Baltimore			Southern Region: Jacksonville (Highest public tree density)			Southern Region: Montgomery (Highest private tree density)		
Category	Trees (n)	Trees per acre	Category	Trees (n)	Trees per acre	Category	Trees (n)	Trees per acre	Category	Trees (n)	Trees per acre	Category	Trees (n)	Trees per acre
Trees by land use			Trees by land use			Trees by land use			Trees by land use			Trees by land use		
Residential	179K	44	Residential	1,179K	23	Residential	160K	10	Residential	65K	31	Residential	65K	31
Park	136K	88	Park	992K	34	Park	22K	22	Park	5,346K	88	Park	5,346K	88
Roadside	108K	34	Roadside	339K	34	Roadside	144K	21	Roadside	2,852K	45	Roadside	2,852K	45
Others	16K	87	Others	106K	28	Others	92K	20	Others	456K	72	Others	456K	72
Trees by ownership			Trees by ownership			Trees by ownership			Trees by ownership			Trees by ownership		
Public	244K	61	Public	1,331K	34	Public	166K	21	Public	8,198K	66	Public	4,099K	66
Private	195K	66	Private	1,285K	26	Private	251K	15	Private	521K	51	Private	260K	51
Total trees	439K	63	Total trees	2,616K	30	Total trees	417K	18	Total trees	8,719K	59	Total trees	8,719K	59

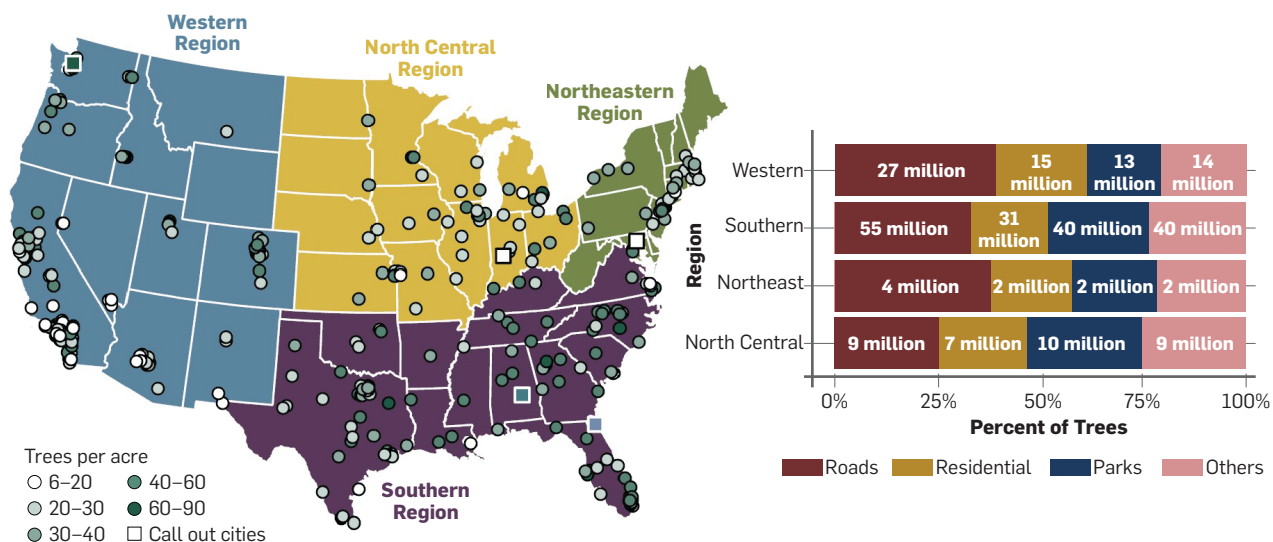


Table 2. Validation cities¹ with validation metrics.

City	Count GT (GT)	Detected (P)	Correct (match)	Recall	Commission Rate	Omission Rate	Precision
Columbus	5482	5669	5414	0.99	0.04	0.01	0.96
NYC	1472	1517	1428	0.97	0.06	0.03	0.94
Washington, DC	3845	3680	3596	0.94	0.02	0.06	0.98
Charlottesville	2120	2232	2005	0.95	0.10	0.05	0.90
Buffalo	5125	4829	4663	0.91	0.03	0.09	0.97
Sioux Falls	2865	3104	2787	0.97	0.10	0.03	0.90
Pittsburgh	3070	3327	2991	0.97	0.10	0.03	0.90
Los Angeles	2721	2981	2682	0.99	0.10	0.01	0.90
Miami	1427	1493	1391	0.97	0.07	0.03	0.93
Santa Monica	1557	1721	1503	0.97	0.13	0.03	0.87
Boulder	1281	1316	1202	0.94	0.09	0.06	0.91
Sparks	1475	1512	1398	0.95	0.08	0.05	0.92
Denver	3677	3873	3531	0.96	0.09	0.04	0.91
Cupertino	1094	1042	995	0.91	0.05	0.09	0.95
San Jose	852	942	834	0.98	0.11	0.02	0.89
Phoenix	965	1008	910	0.94	0.10	0.06	0.90
San Francisco	8398	8768	8209	0.98	0.06	0.02	0.94
Seattle	816	932	797	0.98	0.14	0.02	0.86
San Antonio	1363	1406	1296	0.95	0.08	0.05	0.92

1). We extract the rule parameters from satellite images and from urban structural data, such as building footprints and road networks (e.g., OpenStreetMap²⁶). Using this information, we generate 100,000 sample pairs per configuration type. Each sample pair contains a tile with tree blobs and a corresponding tile with Gaussian discs representing tree locations (with calibration shown in the surface plots in Figure S7 in the supplement). The generator captures both the local and global context of the tree blobs, obtaining their shape, size, and spatial arrangement in a compressed latent representation based on the inferred urban rules for the various configurations (see Figures S5 and S6 in the supplement for an illustration). For cGAN inference, the configuration type of each blob of an input tile is used to condition the generation of the most likely tree locations (Figure 3, bottom; also Figure S4).

Loss function and objective. The cGAN model is designed to translate tiles with tree blobs (X) into spatially continuous probability maps (Y), where each tree is represented as a Gaussian disc. These soft activations are post-processed via peak detection

to produce precise tree-point locations. Our model is trained to produce structured spatial patterns that reflect the presence and relative positions of individual trees.

The loss function used to train the model is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GAN}} + \lambda \cdot \mathcal{L}_{L1}$$

where

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{(X,Y)}[\log D(X,Y)] + \mathbb{E}_{(X,Y)}[\log(1 - D(X,G(X)))]$$

is the adversarial loss facilitating the generation of Gaussian discs representing trees, and

$$\mathcal{L}_{L1} = \mathbb{E}_{(X,Y)}[\|Y - G(X)\|]$$

is the pixel-wise loss that enforces spatial alignment and Gaussian smoothness.

Hyperparameter discussion. The cGAN is configured with hyperparameters for translating tree blob inputs into structured probabilistic outputs. An important hyperparameter is the L1 loss weight (λ), which ensures that the outputs remain quantitatively accurate and consistent with expected tree distributions. In our experiments, we set $\lambda = 100$ (where below 20 gave low precision and above 200

led to overfitting).¹⁸ Tile size is another parameter (Figure S8 in supplement), which was found to work best at 256 x 256 pixels (or 80 x 80 meters). All models are optimized using the Adam optimizer, with a learning rate and β_1 set to 2×10^{-4} and 0.5, respectively. To improve generalization, dropout is applied in the decoding path of the generator. These hyperparameters collectively ensure that the model produces spatially aligned, point-representative outputs suitable for post-processing-based discretization of tree locations.

Results and Analysis

We validated our model by using existing, extremely time consuming ground-truth datasets for 19 U.S. cities¹ and partial ground-truth statistics for 79 U.S. cities (sources in Table 2 and supplementary Datasets S2 and S3), and by ensuring cities encompass all ecoregions in the mainland U.S. The complete dataset release of 330 cities is in Dataset S1 (see supplement). On average, our method's count accuracy is 92.5% (with high statistical significance of $R^2=0.998$) and the mean tree localization error is 1.49m with a standard deviation of only 0.42m (Figure 1). Using a metric proposed by researchers to detect individual trees in high-resolution aerial images,³⁷ our single-tree localization has a high accuracy rate, with a recall of 0.95, a precision of 0.92, a commission rate of 0.08, and an omission rate of 0.04 for the 19 validation cities.¹ A tabular and city-specific presentation is shown in Table 2.

Our analysis further showed the robustness of our approach across different spatial resolutions of satellite data. As we degraded the images to lower resolutions, we consistently see relatively accurate localization performance (Figure 4a). We moved from our native 3 meters per pixel (mpp) resolution (PlanetScope) to 5mpp (RapidEye), 10mpp (Sentinel-2), and 30mpp (Landsat). The difference between our native performance of normalized accuracy of 0.91 and the coarsest imagery (Landsat) was 0.16—a much smaller difference compared to competing models, even at native resolution (Figure 4b).

To understand computational

Figure 4. Robustness and comparison of accuracy and efficiency. (a) Box plots showing quantitative performance of our model using satellite imagery of different spatial resolutions. (b) Comparisons in terms of normalized localization accuracy and efficiency against 10 SOTA models, including multiple diffusion-based and transformer-based models (shown in black circles, whereas ours is shown in a green circle). The call-out boxes highlight statistics of our model and the next-best models in terms of efficiency and accuracy. Qualitative comparisons are also available in the supplementary material.

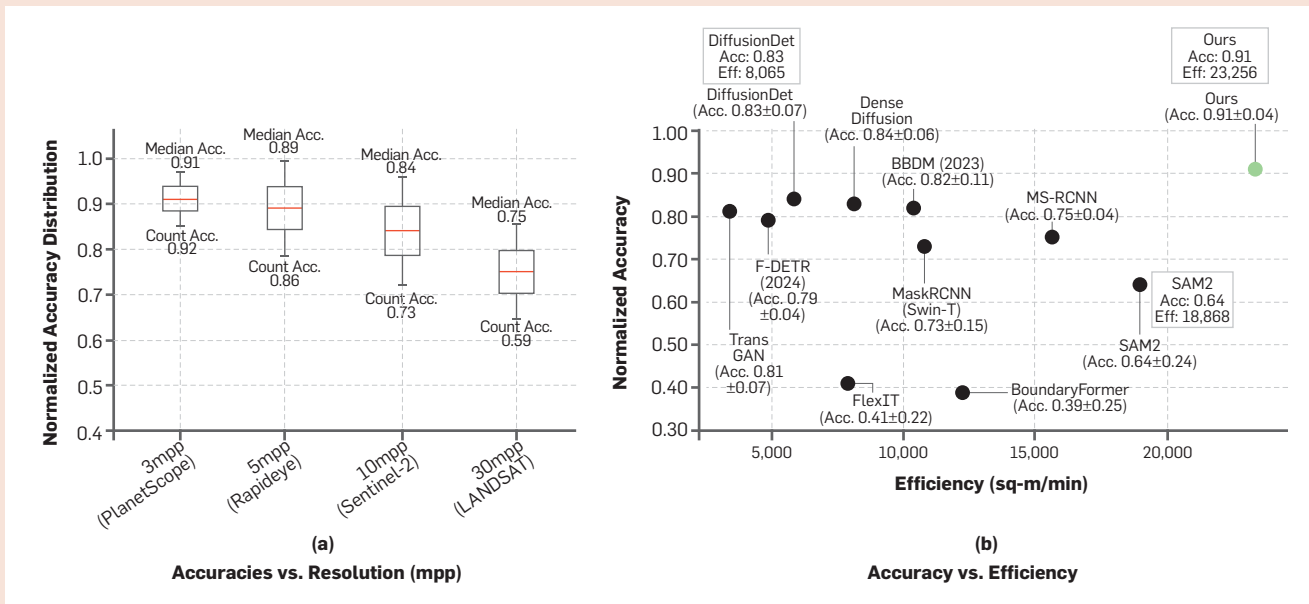
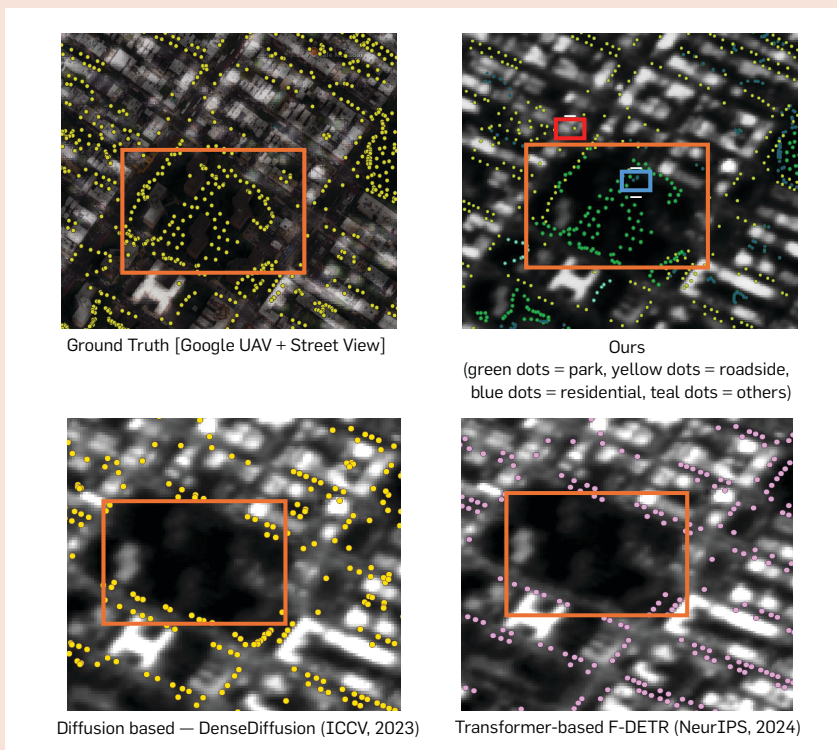


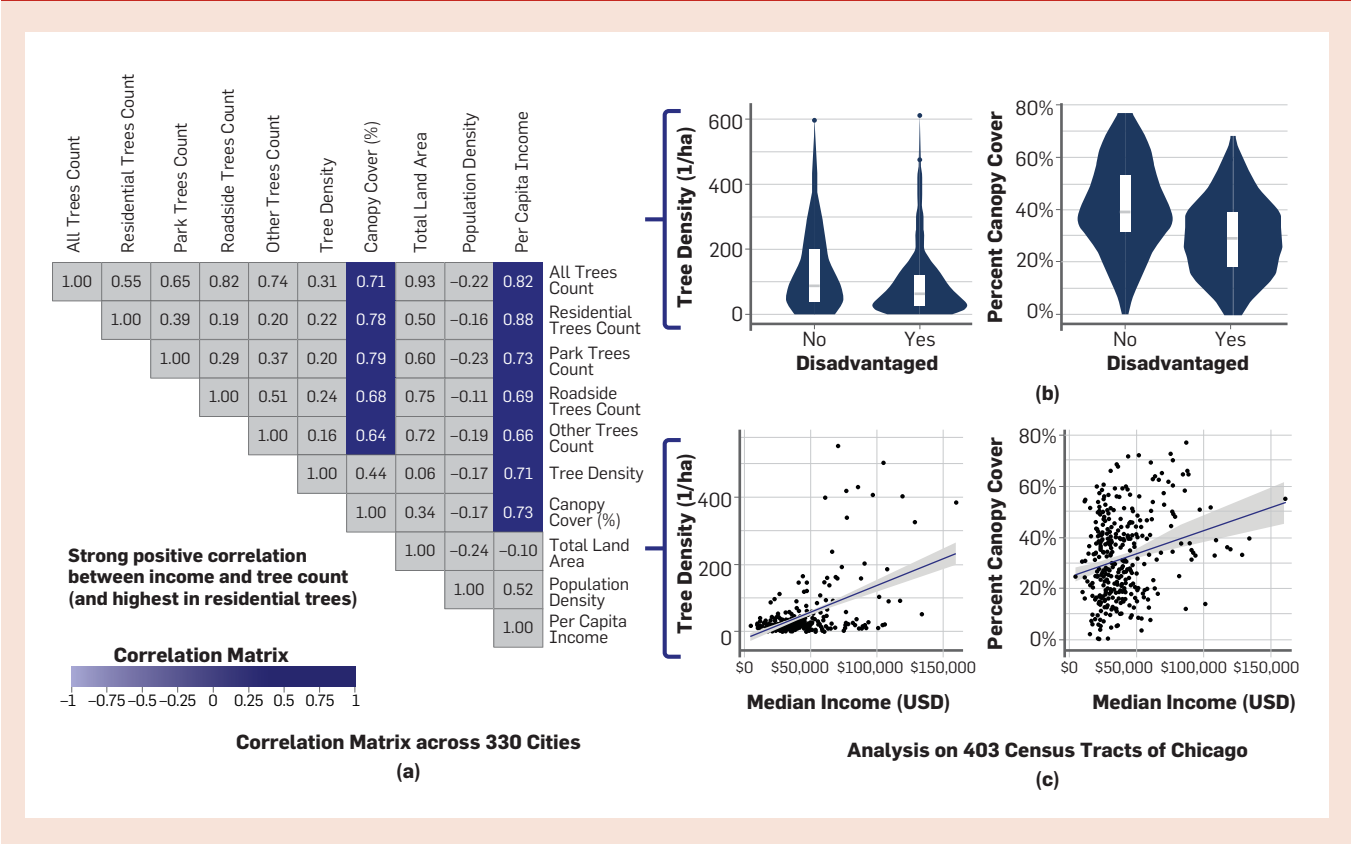
Figure 5. Qualitative comparisons with diffusion- and transformer-based approaches. Comparative snapshot of 6th Ave. Community Garden/Park in Midtown Manhattan, NY. Red box = case where a tree was localized over a building (less than 0.3%), blue box = park tree identified as residential, orange box = subjective superior performance of our model in the presence of occlusion and unresolvable areas where trees are significantly missing in transformer- and diffusion-based models. Note: Only our model offers such configuration-segmented localization. More examples can be found in the supplementary material (Figure S9).



cost and efficiency across different models, we compared our model's efficiency against 10 state-of-the-art (SOTA) baselines that included both diffusion-based and transformer-based models. Using the same computational resources, we processed 23,256m² per minute, whereas the next-best model processed 18,868m² per minute and with approximately 30% less accuracy (see Figure 4b across the horizontal axis).

We then compared our model to SOTA baselines in terms of both normalized localization accuracy, count accuracy, and efficiency (see Figure 4b across the vertical axis). We selected the recent best performing transformer-, diffusion-, and convolution-based models, including F-DETR,²⁰ Swin-Transformer-based MaskRCNN,²¹ SegmentAnything2,²⁸ DiffusionDet,² Mask Scoring RCNN,¹⁰ BBDM,¹⁹ BoundaryFormer,¹⁶ Dense-Diffusion,¹³ FlexIT,³ and TransGAN.¹² All models used four RTX 4090 GPUs for processing/prediction. Although there are limited failure cases (e.g., missing some trees that overlap with buildings), our model outperforms diffusion-based and transformer-based models (Figure 5). This analysis further highlights the

Figure 6. Disparity of tree presence in relationship to income level. We show the positive relationship between urban tree abundance and income level. (a) Correlation matrix among tree counts from our method, with population and economic data across 330 cities. (b) Violin plots showing current tree density and canopy cover disparity between disadvantaged and non-disadvantaged census tracts across 403 census tracts in Chicago. (c) Regression plot showing the same Chicago census tracts plotted to show tree density across varying income levels.



strength of our model when it comes to occlusion handling and challenging cases such as limited visibility.

Urban Tree Analytics

As a demonstration of the analytics enabled, and easily repeatable and updateable by our method, we provide the following example studies.

Nationwide urban tree perspective. Given the broad-scale yet still per-tree localization ability of our method, we provide for the first time nationwide urban tree statistics that can be updated regularly (Figure 2). We partition the 330 cities into four geographic regions: Northeast (NE), Northwest (NW), Southeast (SE), and Southwest (SW) using the geographic center of mainland U.S. as that set by the U.S. National Geodetic Survey (NGS) - 39°50'N 98°35'W. Figure 2 illustrates the counts in all regions.

Our method showed that the NE region had the highest mean density of 41 urban trees per acre, with a standard deviation of 11 trees per acre.

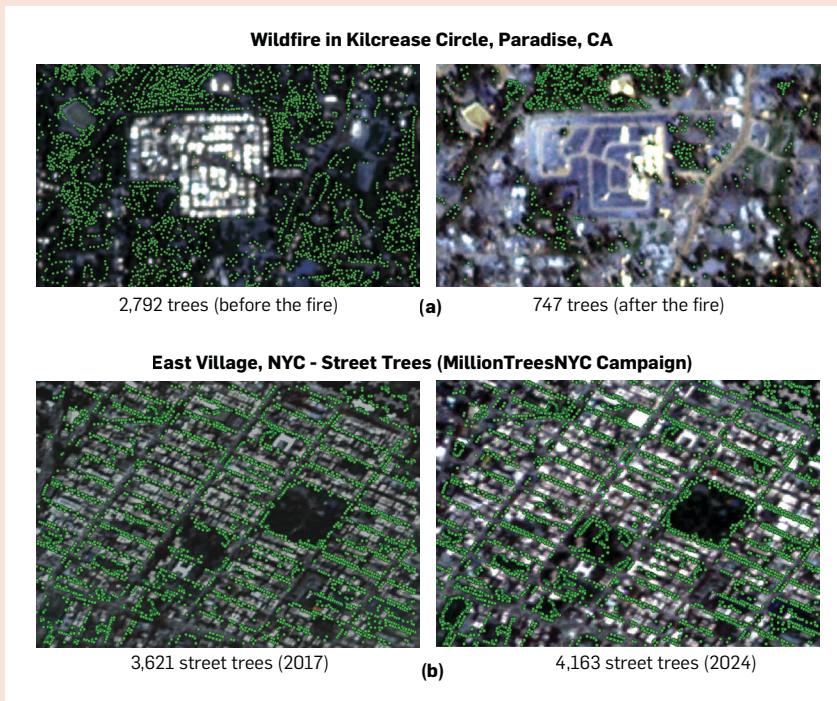
Within this region, the highest density was 69 trees per acre in Ann Arbor, MI, and the lowest density was Newark, NJ, with 17 trees per acre. The second highest mean tree density region was the NW region, with a mean density of 37 trees per acre and a standard deviation of eight trees per acre. In this region, the maximum density was 63 trees per acre in Seattle, WA, whereas the minimum was 20 trees per acre in Provo, UT. The third-highest mean tree density region was the SE, with a mean density of 31 trees per acre and a standard deviation of 10 trees per acre. In this region, the city with highest tree density was Sandy Springs, GA, with 86 trees per acre. In contrast, the lowest density was seven trees per acre for Corpus Cristi, TX. Lastly, the region that exhibited the lowest mean tree density was the SW, with 28 trees per acre and a standard deviation of 11 trees per acre. In this region, the highest density was 53 trees per acre in Lakewood, CO, and the lowest was eight per acre in Inglewood, CA.

We further observed that currently the cities in the NE region have the lowest relative presence of residential trees (with a mean of 19%), whereas the NW has the largest residential relative presence of 25%. In another observation, NE cities were dominated by urban park trees, with a 25% relative presence in a region boasting more than 43 million trees. NW comes last in this category, with only 17% park trees.

Urban tree socioeconomic impact.

Our approach enables correlating current tree localization with the Climate and Economic Justice Screening Tool (CEJST)³³ based on the current U.S. census (Figure 6). In recognition of the pivotal role trees play in urban settings and as a response to the disparity in tree presence between wealthier urban areas and disadvantaged areas, city governments have undertaken substantial investments to expand their tree cover, particularly focusing on marginalized neighborhoods. Our method enables quantifying and updating the trends for the first time, to

Figure 7. Before/after major events monitoring. (a) The 2018 “Camp Fire” near Paradise, CA, is one of the deadliest fires on record. Our method was automatically applied to satellite imagery before and after the fire, resulting in a tree loss estimate for this relatively small region of 1.2 square kilometers. (b) The MillionTreesNYC initiative started in 2007, ended in 2017, and was recently reinstated until 2030. We show the past and current performance of urban planting projects, such as the NYC campaign.



the best of our knowledge. Currently, we found strong positive Pearson correlation coefficients (CC) between per capita income and the total number of trees ($CC=+0.82$), number of residential trees ($CC=+0.88$), street trees ($CC=+0.69$), park trees ($CC=+0.73$), and other trees ($CC=+0.66$). A similar trend was seen with a strong CC of $+0.73$ between per capita income and canopy cover. Figure 6 illustrates a detailed validation conducted in Chicago. It reveals that within Chicago’s 403 census tracts, disadvantaged regions exhibit substantially lower median tree density: approximately 42% fewer trees per hectare. The same trend is found for canopy cover, showing a difference of 11% in Chicago. Besides census tracts labeled as disadvantaged, a similar pattern emerges when assessing the census tracts in relation to income levels in Chicago: Higher-income tracts consistently exhibit greater tree presence. These trends can be regularly updated by tree data extracted by our approach.

Trees after disasters and planting initiatives. As a demonstration, we performed fine-grained monitoring

tasks in response to urban disasters and tree-planting initiatives (Figure 7). For example, the “Camp Fire” in 2018 was the deadliest and most destructive wildfire in California history. With our approach, we localized and counted the trees before and after the fire by processing the corresponding satellite imagery. Figure 7a illustrates the micro-scale situation of tree analytics from a temporal perspective for a part of a city, namely the Kilcrease Circle apartment complex area in Paradise, CA, in the wake of the Camp Fire. To generalize even further, we conducted a before/after analysis for the full city of Vacaville, CA, which went ablaze during the 2020 LNU Lightning Complex fires. For the full city, we observed a loss of 80,916 trees across its 7,975 acres. The fire decreased the canopy cover in the city from 41% in 2019 to 35% in 2021. These estimates match up with small, ground-based efforts but provide the first-ever tool to assess tree loss at such a scale, and with minimal effort.

As another example, one of the most prominent urban tree-planting drives was the MillionTreesNYC campaign.²⁵

It first ran from 2007 until 2017, and was then revived and reactivated with another million-tree target for 2030. These planting campaigns focus heavily on planting street trees, as they are public and accessible. Our method is the first one to extract fine-grained analytics for such initiatives. It was reported that New York City (NYC) had approximately 5.2 million trees in 2005.²⁵ The same report stated that it took more than 2,000 volunteers a total of 30,000 hours to count the street trees in NYC. Later, in 2017, the USDA released a plot-level count of NYC trees at approximately seven million, based on an aggregation from 296 land plots.²⁴ We processed more than 500km² of NYC and found approximately 6,795,102 trees, of which 688,088 were street trees. A repeat analysis for 2024 yielded 6,985,902 trees, of which 720,839 were street trees (a 4.75% increase in street trees compared to 2017, but also a decline of 5.3% in residential trees). Our density agreed with a recent work by Niese et al.²³ Therefore, the trees were not only counted but also categorized into four categories, of which street trees are particularly interesting due to the MillionTreesNYC campaign. Overall, our method showed (and localized) a net increase of 2.3% trees in all of NYC between 2017 and 2024. Figure 7b shows an illustration of the localized street trees in the East Village, NYC, in 2017 and 2024. Our automatic approach took less than five hours (and thus far less than the 30,000 hours of human effort reported) to accomplish the localization and categorization of urban trees for the full city, showcasing the efficient application of our novel approach in extracting fine-grained analytics from tree-planting campaigns.

Moving Forward

As urbanization continues, understanding the quantity and distribution of urban trees is urgently needed for sustainable, effective management and policymaking. Using our AI-based method, for the first time we can update urban tree inventories at will on both public and private lands at 1.5m positional accuracy and 92.5% count accuracy, spanning at least 330 U.S. cities with a population more than 100,000. Instead of relying solely on high spatial resolution, our approach

uses spatiotemporal data and generative AI modeling to overcome occlusion, abutting trees, leaf-on/leaf-off conditions, and fall color assumptions. The system enables updating trends in the distribution of trees and identifying differences in tree density and canopy cover across ecoregions, providing a compelling tool for anomaly detection, among many other applications. Our approach is automatic and repeatable, implying that the monitoring for any city can be computed in a few hours and quickly repeated at will. The same city can be processed in an average of three to five hours (subject to the area) and yield a mean positional accuracy of 1.5m per tree. Nationwide processing can be done in under a day by using multiple computers.


We acknowledge the existence of high-resolution point cloud data such as LiDAR, but it is either airborne or backpack based and covers limited spatial areas and at limited temporal frequency—which is why, for a national scale, our work scales well. However, for places of particular interest (where fine-grained data collection is possible), we see positive potential to fuse our methodology of overhead localization with stem-based (understory) localization through multimodal approaches.

It is also notable that our method can serve well for the rapid development of urban planning policies and can simulate impacts using “what if” situations. This makes a difference, especially in smaller cities with limited budgets for tree analytics and planning. Since recent approaches create simulation-ready 3D reconstruction datasets from single images,¹⁷ our work can serve as one of the key sources for photorealistic simulation of urban twins. Moving forward, and with the help of an NSF cyberinfrastructure grant, we are incorporating our methods into a Web portal for timely national-scale tree inventory estimation and what-if planning tools.

Looking forward, we anticipate that timely tree inventory and localization data can also be used to improve weather/climate modeling, by building upon our prior works^{6,7} and by integrating with, for example, WRF Urban,^{15,36} air quality (e.g., Kim et al.,¹⁴ Li et al.¹⁸), and particulate matter (PM) absorption and dispersion models. We envision the novel and scalable nature


of this work will be adopted across the globe toward the broad goal of greener, more sustainable cities.

Acknowledgments

This work was made possible by NSF Grants 1835739, 2106717, 2417510, 2412928, and 2411273, USDA NIFA Grants 1032672 and 2023-68012-38992, and Purdue University Institute for Digital Forestry. 

References

1. Beery, S. et al. The auto arborist dataset: A large-scale benchmark for multiview urban forest monitoring under domain shift. In *Proceedings of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (2022), 21262–21275.
2. Chen, S., Sun, P., Song, Y., and Luo, P. DiffusionDet: Diffusion model for object detection. In *Proceedings of IEEE/CVF Int. Conf. Comput. Vis.* (2023), 3423–3433.
3. Couairon, R. et al. FlexIT: Towards flexible semantic image translation. In *Proceedings of IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* (2022), 14048–14058.
4. Drescher, M. Urban heating and canopy cover need to be considered as matters of environmental justice. *Proceedings of the National Academy of Sciences* 810, 116 (2019), 26153–26154.
5. Earth Define. US Tree Map (2023).
6. Firoze, A. et al. Tree instance segmentation with temporal contour graph. In *Proceedings of 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* (2023), 2193–2202.
7. Firoze, A., Benes, B., and Aliaga, D. Urban tree generator: Spatio-temporal and generative deep learning for urban tree localization and modeling. *The Visual Computer* 38(2022), 3327–3339.
8. Fischer, E., Sippel, S., and Knutti, R. Increasing probability of record-shattering climate extremes. *Nature Climate Change* 11 (2021), 689–695.
9. Gomes, C. et al. Computational sustainability: Computing for a better world and a sustainable future. *Commun. ACM* 62, 9 (2019), 56–65.
10. Huang, Z., Huang, L., Huang, Y., and Liu, W. Mask scoring R-CNN. In *Proceedings of IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* (2019), 6409–6418.
11. Hutt-Taylor, K. and Ziter, C. Private trees contribute uniquely to urban forest diversity, structure and service-based traits. *Urban Forestry & Urban Greening* 78 (2022).
12. Jiang, Y. et al. TransGAN: Two pure transformers can make one strong GAN, and that can scale up. *arXiv preprint arXiv:2102.07074* (2021).
13. Kim, S., Kim, J., and Kwak, S. Dense text-to-image generation with attention modulation. In *Proceedings of IEEE/CVF Int. Conf. Comput. Vis.* (2023), 7700–7710.
14. Kim, Y. et al. MUNICH v2.0: A street-network model coupled with SSH-aerosol (v1.2) for multi-pollutant modelling (2022).
15. Kusaka, H. et al. The integrated WRF/urban modelling system: Development, evaluation, and applications to urban environmental problems. *Intern. J. of Climatology* 31, 1 (2011), 273–288.
16. Lazarow, J., Xu, W., and Tu, Z. Instance segmentation with mask-supervised polygonal boundary transformers. In *Proceedings of IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* (2022), 1292–1302.
17. Lee, J.J. et al. Tree-D Fusion: Simulation-ready tree dataset from single images with diffusion priors. In *ECCV 2024*. Springer-Verlag (2024), 439–460.
18. Li, Z., Wang, Y., and Liu, Q. Fabric defect data augmentation using conditional generative adversarial networks. *Soft Computing* 28 (2024), 6109–6122.
19. Li, B., Xue, K., Liu, B., and Lai, Y. BBDM: Image-to-image translation with Brownian bridge diffusion models. In *Proceedings of IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* (2023), 1952–1961.
20. Liu, Y., Wang, Z., and Wang, Y. F-DETR: Fusion detection transformer with heterogeneous scale multi-branch structure. *Applied Soft Computing* 147, (2024), 110990.
21. Liu, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021).
22. Nearmap. Aerial map coverage (2023).
23. Niese, T. et al. Procedural urban forestry. *ACM Trans Graph.* 41 (2022), 2.
24. Nowak, D. et al. The urban forest of New York City. U.S. Department of Agriculture, Forest Service, Northern Research Station (2018).
25. NYC Parks and Recreation Project. Street tree census (2017).
26. OpenStreetMap contributors and OpenStreetMap; <https://www.openstreetmap.org>
27. Planet. Planet explorer (2023).
28. Ravi, N. et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714* (2024).
29. Ronneberger, O. et al. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention* (2015), 234–241.
30. Smith, I. et al. Live fast, die young: Accelerated growth, mortality, and turnover in street trees. *PLoS ONE* 14 (2019).
31. Szewczyk, R. et al. Habitat monitoring with sensor networks. *Commun. ACM* 47, 6 (2004), 34–40.
32. U.S. Census Bureau. City and town population totals: 2020–2022. (2023).
33. Walker, K. and Herman, M. TidyCensus: Load US census boundary and attribute data as ‘tidyverse’ and ‘sf’-ready data frames (2023); <https://walker-data.com/tidyCensus/>
34. Willis, K. and Petrokofsky, G. The natural capital of city trees. *Science* 356 (2017), 374–376.
35. Wolf, K. et al. Urban trees and human health: A scoping review. *Intern. J. of Environmental Research and Public Health* 17 (2020).
36. Wong, M. et al. Evaluation of uWRF performance and modeling guidance based on WUDAPT and NUDAPT UCP datasets for Hong Kong. *Urban Climate* 28 (2019), 100460.
37. Zheng, Y. and Wu, G. Single shot multibox detector for urban plantation single tree detection and location with high-resolution remote sensing imagery. *Frontiers in Environmental Science* 9 (2021).

 more online
To view the supplement for this article, please visit <https://dl.acm.org/doi/10.1145/3762636> and click on Supplemental Material.

Adnan Firoze is a Ph.D. student in the Department of Computer Science at Purdue University, West Lafayette, IN, USA.

Akshaj Uppala is an M.S. student in the Department of Computer Science at Purdue University, West Lafayette, IN, USA.

Lindsay Darling is a Ph.D. student in the Department of Forestry and Natural Resources at Purdue University, West Lafayette, IN, USA.


Raymond A. Yeh is an assistant professor in the Department of Computer Science at Purdue University, West Lafayette, IN, USA.

Bedrich Benes is a professor in the Department of Computer Science at Purdue University, West Lafayette, IN, USA.

Brady Hardiman is an associate professor in the Department of Department of Forestry and Natural Resources at Purdue University, West Lafayette, IN, USA.

Songlin Fei is a professor in the Department of Department of Forestry and Natural Resources at Purdue University, West Lafayette, IN, USA.

Daniel Aliaga (aliaga@purdue.edu) is an associate professor in the Department of Computer Science at Purdue University, West Lafayette, IN, USA.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).



Watch the authors discuss this work in the exclusive *Communications* video. <https://caom.acm.org/videos/city-trees>

DOI:10.1145/3769689

Though new regulatory frameworks address fairness, accountability, and safety in AI systems, they often fail to directly mitigate the subtle “communication bias” in LLMs that can distort public discourse and democratic processes.

BY ADRIAN KUENZLER AND STEFAN SCHMID

Communication Bias in Large Language Models: A Regulatory Perspective

ARTIFICIAL INTELLIGENCE (AI) involves computational systems designed to perform tasks that traditionally require human intelligence, such as perception, reasoning, and language understanding. Large language models (LLMs), a prominent subset of AI, are built on advanced neural network architectures that can generate new data, including text, images, and audio. LLMs use various technologies to identify patterns in a given set of training data, without

requiring explicit instructions about what to look for.^{12,35} LLMs typically assume that the training data follows a probability distribution; once they have identified existing patterns, they can generate new instances that are similar to the original data. By drawing from and combining training data, LLMs can create new content that transcends the initial dataset.¹⁷ But this process has technical limitations. For instance, LLMs can hallucinate by producing plausible yet factually incorrect or fabricated information; they can struggle with ambiguous or missing data, leading to blind spots in their outputs; and they can reflect and amplify existing preconceptions present in their algorithms or training data, leading to *communication bias*, which we focus on here.

We propose a broad definition of communication bias: the expression, amplification, and systematic favoring of certain social, cultural, or political perspectives in the output generated by LLMs, which can affect the attitudes and beliefs of users and thus, ultimately, public discourse. We

» key insights

- **As AI systems become gatekeepers of information, LLMs may reinforce existing biases and promote particular perspectives, creating and perpetuating echo chambers and intensifying societal polarization.**
- **Emerging regulatory frameworks such as the EU AI Act and the DSA seek to mandate transparency, accountability, and fairness in AI systems, yet they often address communication bias only as a byproduct of broader safety and content moderation measures.**
- **Detecting and mitigating communication bias in LLMs is inherently challenging due to the models' complexity and the subtle, multifaceted nature of such bias.**
- **A comprehensive approach combining value chain regulation, content moderation, competition, and ongoing technology design governance is crucial in fostering diverse and transparent AI systems that mitigate bias while promoting a balanced digital information ecosystem.**

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK.COM



use the term *communication bias* as opposed to *data bias*, which is a possible cause of communication bias arising from skewed or unrepresentative training data; and *automation bias*, which may exacerbate these effects by making users more likely to accept and act on biased AI outputs without sufficient scrutiny. Specifically, communication bias need not only manifest in the selective presentation of facts or the rendering of false or made-up information, but can also reside in claims that sit somewhere in between, or outside, the categories of true and false.²³ Here, a prejudiced effect results from generative expression rather than imbalances in training data (data bias) or uncritical user uptake (automation bias).

Recent research shows that commercial LLMs consistently display communication bias and can adapt their responses to user or persona prompts.³ The integration of LLMs into consumer-facing applications such as chatbots, virtual assistants, and search engines has rendered them the primary interfaces between users and AI, enabling immediate, interactive communication and content generation. Hence, LLMs have a broad range of applications across various sectors.⁹ They can be employed to assist businesses by analyzing financial data and generating content, and can be used for tasks such as text generation, translation, and summarization. LLMs can also enhance personal user experiences. For instance, LLMs can facilitate patient–doctor communications, aid in personalized learning, provide life advice, and inform political decisions.³⁷ Owing to these features, LLMs have the potential to supplant existing gatekeepers and become agents that mediate various interactions, select underlying applications, and choose providers to respond to user queries.¹ By assuming control and entrusting consumers' ability to steer demand, input, and data to AI agents, LLMs may gradually come to serve as the main access points by which users engage with digital technologies and the Internet, eluding the grasp of present legal frameworks and thus becoming a distinct and vital new object of regulatory consideration.²⁴

LLMs may come to serve as the main access points by which users engage with digital technologies and the Internet, eluding the grasp of present legal frameworks.

This article provides an overview of the rapidly developing legal and regulatory background that governs AI and machine learning technologies, with a particular emphasis on LLMs' ability to affect citizens' political opinions and voting decisions. Recent studies demonstrate that LLMs can influence people's views, but there is currently no framework that can effectively address the issue. As AI and machine learning technologies become more widespread, there are increasingly urgent calls for proactive regulatory frameworks to establish guardrails that address the models' limitations and ensure their responsible use within society.¹³ For instance, the FBI's 2023 Internet Crime Report highlights that manipulation and social engineering have led to substantial financial losses, surpassing \$12.5 billion in 2023.¹¹ The European Union Agency for Cybersecurity's (ENISA's) 2023 Threat Landscape report identifies highly realistic synthetic media as progressively more hazardous to digital identity and trust.¹⁰ Due to the inherent complexity of LLMs and the difficulty of targeting communication bias in the models' outputs, the search for new regulatory approaches remains critical in ensuring an open digital information ecosystem.³

An emerging source of tension resides in the possibility that LLMs carry communication biases. For example, studies have revealed a variety of political leanings across different models and have demonstrated how such biases can impact the effectiveness of hate speech and misinformation detection systems.³ Other studies have identified biases in open source LLMs around divisive topics such as immigration, reproductive rights, and climate change.³ While most existing models appear to have liberal dispositions and are primarily U.S.-centered in their outputs, LLMs can also exhibit a range of conservative biases, depending on the issue.³ Such biases can be attributed to the models' algorithms and/or training data, but they may also arise as a result of purposeful decisions by AI companies. Users may not notice any bias when it comes to the deployment of LLMs, and most models have built-in safeguards designed to prevent the

generation of toxic content. However, communication bias can emerge in subtle ways, and the additional training aimed at limiting certain kinds of content generation can inadvertently introduce more bias. The problem may escalate as LLMs become more widespread, with future generations of the models increasingly trained on data that includes AI-generated output.¹⁴ In addition, some social groups may try to sway LLMs to promote their perspectives over those of others and may attempt to influence the models' training data. The issue is exacerbated if a few influential corporations come to dominate the AI market, providing citizens with the information that creates public knowledge.⁷ If LLMs selectively present or omit particular perspectives, this will inevitably shape public discourse by limiting the diversity of available opinions and possibly undermining democratic processes.

A number of regulations in the EU seek to address this tension. None, however, applies across the board, and the targeting of subtle communication bias in the output generated by LLMs frequently turns out to be a byproduct of more straightforward mechanisms intended to ban illegal content. In the following sections, we provide an overview of the existing regulations and offer potential routes for further discussion about the risks of communication bias in LLMs, as well as how such risks might best be addressed. While our focus is on the AI Act^a and the Digital Services Act (DSA),^b these frameworks do not operate in isolation: The size of the EU market and the EU's comprehensive and enforceable legal standards set de facto global norms that companies around the world must increasingly follow. AI technologies, especially LLMs, are developed and deployed transnationally, and different legal frameworks may lead to regulatory fragmentation, compliance burdens, and uneven protection for users.¹³ At this stage, focusing predominantly on the EU is justified because its rules are already influencing legislative agendas and business practices

in other jurisdictions, serving as a blueprint for global AI governance. Examining the EU's regulatory model and its underlying principles provides unique insight into the evolving global environment, the challenges of harmonization, and the practical realities faced by AI providers.

AI-Enabled Communication: Risks and Mitigation Strategies

Several studies have focused on identifying bias present in the output of LLMs, which can lead to subtle but harmful stereotypes and misinformation.³ In this section, we elaborate on some specific challenges relating to such bias and how these challenges complicate the backdrop of trustworthiness and reliability of AI-enabled communication.

Risks of AI-enabled communication. With the increasing spread of AI-generated outputs, digital information ecosystems are changing in a number of ways. A relatively well-known and distinct issue concerns AI-generated synthetic content as well as deepfakes and AI-generated text, which can mislead individuals and the public, eroding trust in AI-driven communication (Art. 50, AI Act). A far less overt form of harm that may nonetheless corrode the conditions of free and equitable public discourse involves the vast amount of AI-generated content that aligns with certain political narratives or worldviews. In the digital age, this threat is characterized by the shift from state-controlled speech to platform-dominated governance, where algorithms often prioritize engagement over democratic integrity.³² As LLMs become integrated into content-creation processes or online services such as search engines or chatbots, their output may subtly influence popular attitudes and beliefs and may impact societal perceptions. AI-powered search engines offer more conversational and context-aware interactions compared to traditional ones and are better able to personalize and customize their outputs based on user interactions.^{24,27} When responses are produced that align with user preferences rather than objective information, this can create and perpetuate echo chambers,

resulting in increased polarization. In turn, AI-generated outputs can facilitate feedback loops that reinforce existing beliefs and attitudes, limiting exposure to a diversity of perspectives. Even if users are aware they are relying on AI-generated content, they may still contribute to societal polarization and less-informed decision making, as LLMs can imperceptibly direct public opinion toward particular party positions and produce an unbalanced representation of views on public affairs.²⁷ In this way, LLMs increasingly mediate the communicative acts deemed necessary for the formation of public opinion by curating information, yet their biases risk distorting public discourse, reinforcing echo chambers, marginalizing dissent, and undermining democratic authorship.³²

Current approaches to address the risks of AI-enabled communication.

Due to the subtle nature of AI-enabled communication, efforts to address its risks will likely rely on a multifaceted approach based on value chain regulation, content moderation, competition, and technology design governance. Legislators, particularly in the EU, have established ex ante requirements across the AI value chain under the AI Act, imposing legal obligations on several actors, depending on the role they play. The AI Act targets providers (those who develop LLMs or make them available), deployers (those who use LLMs), importers, and distributors. It introduces a risk-based approach, categorizing AI systems according to different risk levels—unacceptable, high-risk, limited risk, and minimal risk—with stricter obligations for high-risk and general-purpose AI (GPAI) models, including LLMs (Art. 51(1)(a), Annex XIII). Some key requirements include risk management systems (Art. 9), data governance and quality criteria (Art. 10), the provision of technical documentation (Art. 11), transparency obligations (Art. 13), and human oversight (Art. 14). For GPAI and high-impact GPAI models, additional transparency, documentation, testing, evaluation, and risk mitigation duties apply (Arts. 53 and 55, Recitals 110–111). Enforcement is carried out by national market surveillance

a <https://artificialintelligenceact.eu/the-act/>

b <https://bit.ly/4r4Ar06>

authorities and the EU AI Office, with fines for noncompliance reaching up to 7% of global annual turnover (Art. 99). These measures seek to ensure that AI systems are developed and deployed with attention to fairness, transparency, and bias mitigation, especially where outputs may affect fundamental rights or democratic processes.

By contrast, content moderation, particularly under the DSA, operates *ex post*, focusing on outputs. The DSA does not specifically address LLMs; it regulates online intermediaries, including platforms and search engines, targeting harmful or illegal content as well as advertising transparency (Art. 1). Even so, the DSA’s provisions apply to LLMs when LLMs are integrated into platforms covered by the DSA. For instance, if LLMs are used for content generation or moderation on very large online platforms (VLOPs) or very large online search engines (VLOSEs), these platforms must ensure compliance with the DSA (Art. 33). In particular, such platforms must implement mechanisms for identifying and mitigating systemic risks, including those related to illegal content, civic discourse, and fundamental rights (Art. 34). Furthermore, the DSA mandates transparency reporting (Art. 15), user redress mechanisms (Art. 20), recommender system disclosures (Art. 27), and cri-

sis response protocols (Art. 36). Enforcement is managed by Digital Services Coordinators and the European Commission (EC), with fines for violations up to 6% of global annual turnover (Art. 74). While the DSA’s scope is narrower—concentrating on platforms rather than the entire AI value chain—it may nonetheless be critical for managing, through human and/or automated filtering systems, the societal impact of LLM-generated content.

From a conceptual perspective, value chain regulation (AI Act) and content moderation (DSA) differ in their objectives, scope of application, and methods of enforcement. Content moderation is typically more limited in scope than value chain regulation and is conducted through a combination of automated systems and human oversight. Both value chain regulation and content moderation, however, are capable of addressing bias mitigation, at least to a certain degree. Value chain regulation seeks to mitigate bias primarily through pre-market mechanisms, by ensuring high-quality datasets for training, bias testing, and accuracy requirements. The emphasis of content moderation is on post-market instruments, such as the removal of harmful or illegal content, algorithmic transparency, and user complaints. While value chain regulation

proactively governs system architecture and deployment, content moderation focuses reactively on managing outputs. Neither, however, addresses how the development and use of LLMs are influenced by the market mechanism itself. We therefore emphasize the need for complementary strategies, including competition policy and ongoing technology design governance, to keep a broader focus on the incentives that shape AI-mediated communication and how these incentives work to foster a pluralistic and trustworthy digital information ecosystem.

Regulating Bias in LLMs

The current landscape of AI-generated content is subject to several regulatory frameworks, although none of them have been devised with the possibility in mind that LLMs may affect citizens’ fundamental worldviews and social perspectives, especially their political opinions and voting decisions. In this section, we explain how these frameworks can nonetheless specifically address communication bias in LLMs, and what their constraints and limitations are.


The AI Act. The AI Act contains a set of obligations that apply to AI systems and GPAI models, among other things, and that explicitly encompass LLMs (Art. 51(1) point (a); Annex XIII). In essence, AI models are the technological foundation, while AI systems are the applications built on top of them (e.g., ChatGPT, which adds a user interface, moderation tools, and specific workflows). The AI Act defines an AI system as a machine-based system designed to operate with varying levels of autonomy and that is capable of generating outputs such as predictions, content, or recommendations based on input data (Art. 3(1)). By contrast, GPAI models are those trained with large amounts of data using self-supervision and capable of performing a wide range of tasks (Art. 3(63)). Providers of AI systems must meet requirements tailored to the system’s use context, such as risk management, ensuring high-quality and representative training data, and transparency for high-risk applications. Providers of GPAI models must comply with specific obligations.

Table 1. Regulatory approaches to LLM governance.


	Value chain regulation (AI Act)	Content moderation (DSA)
Primary objective	Establish <i>ex ante</i> requirements for AI development/deployment	Manage <i>ex post</i> outputs to prevent dissemination of harmful/illegal content
Regulatory scope	All actors in the AI value chain (providers, deployers, importers, distributors)	Intermediary services (VLOPs/VLOSEs hosting LLM-generated content)
Risk framework	Risk-based tiers: unacceptable > GPAI > high-risk > limited risk > minimal risk	Systemic risk mitigation for VLOPs/VLOSEs (illegal content, civic discourse, basic rights)
Key obligations	Conformity assessments Certain fundamental rights impact assessments Technical documentation Human oversight	Notice-and-action mechanisms Transparency reporting Recommender system disclosures Crisis response protocols
Enforcement	Fines up to 7% global annual turnover National market surveillance authorities EU-wide database for high-risk systems	Fines up to 6% global annual turnover Digital Services Coordinators Independent audits
Bias mitigation	Pre-market: training data governance, bias testing, accuracy requirements	Post-market: content removal, algorithmic transparency, user redress mechanisms
Differences	<i>Ex ante</i> compliance focused on system architecture	<i>Ex post</i> accountability focused on output management

They must prepare technical documentation, share information with downstream system providers, and publish a summary of training data. There are additional requirements if the model poses a systemic risk (due to high capability or widespread use), including model evaluation, adversarial testing, systemic risk assessment, and incident reporting. When a provider integrates its model into a system and offers it directly, both sets of obligations—those for models and for systems—apply (Recitals 85, 97).^{2,12}

Several of these obligations are relevant in addressing bias in LLMs. For high-risk systems in particular, high-quality training data must be used, and developers must implement data governance measures to ensure training datasets are representative and do not perpetuate biases. This includes regular audits of datasets used for training LLMs. The AI Act considers biases to be relevant if they are likely to affect the health and safety of persons, negatively impact fundamental rights, or lead to discrimination prohibited under EU law, especially where data outputs influence inputs for future operations (Art. 10(2)). Moreover, the AI Act requires providers to establish a continuous risk management system that identifies, analyzes, and mitigates foreseeable risks to health, safety, and fundamental rights, including risks from bias in system outputs (Art. 9). This process necessitates regular monitoring, testing (including under real-world conditions), and targeted measures to reduce residual risks that directly pertain to communication bias in LLMs. The AI Act complements this obligation by requiring providers to ensure their systems achieve and maintain appropriate levels of accuracy and robustness, explicitly requiring measures to manage feedback loops that could introduce or amplify bias over time, and to benchmark, document, and transparently declare accuracy metrics (Art. 15). Technical solutions must also address vulnerabilities that could be exploited to manipulate outputs, which includes bias-related exposure. In addition, the AI Act requires that human oversight mechanisms are in place for



If LLMs selectively present or omit particular perspectives, this will inevitably shape public discourse by limiting the diversity of available opinions and possibly undermining democratic processes.



high-risk AI systems, ensuring that outputs can be reviewed and corrected if they exhibit biased or harmful features (Art. 14). Such oversight is crucial for maintaining accountability in decision-making processes influenced by LLMs.³⁵ Similarly, some developers must actively prevent LLMs from producing biased or discriminatory outputs. This includes conducting impact assessments to evaluate how the models' outputs affect different social groups and to ensure that harmful or biased content generated by LLMs can be reported and addressed effectively, and corrective actions can be taken if biases are identified (Art. 27).

The DSA. Several of the DSA's provisions may have implications for LLMs, although their application is more nuanced. The DSA regulates a broad range of digital services, focusing on intermediary and hosting service providers as well as online platforms. This includes search engines and recommender or advertising systems, which may indirectly encompass services that use LLMs. The core components of the DSA mandate that platforms implement mechanisms to address illegal content, ensure transparency in their content moderation practices, enable users to challenge content moderation decisions, and hold platforms accountable for illegal or harmful content disseminated through their services. If LLMs are used for content generation, the platforms using LLMs must comply with content moderation and transparency obligations.


The DSA imposes stricter obligations on VLOPs or VLOSEs that reach more than 45 million active EU service recipients on average each month and are designated as such under the DSA (Art. 33(1)). While many LLMs do not meet this threshold on their own, if they are integrated into VLOPs or VLOSEs (e.g., as part of a chatbot feature on a social media website), they may be covered by the DSA. VLOPs and VLOSEs must, among other things, identify, analyze, and mitigate systemic risks associated with their services, particularly those related to illegal content, fundamental rights, freedom of expression and information—including the freedom

and pluralism of the media—and any actual or foreseeable negative effects on civic discourse and electoral processes through the design of their recommender and/or other relevant algorithmic systems, content moderation mechanisms, data-related practices, and so forth.


However, the DSA does not explicitly target LLMs as standalone entities, and its compliance requirements are far from clear-cut. It remains contentious whether GPAI models such as LLMs fall within the DSA's definitions, and such classifications hinge upon, among other things, interpretations of how LLMs store and provide user inputs and outputs.¹⁵ While LLMs that are part of online platforms might be the best fit for the DSA's provisions, most generative AI products will not qualify as VLOPs or VLOSEs, as they typically do not disseminate information publicly in a manner consistent with traditional online platforms.

Limitations of existing legal frameworks. We have identified several legal obligations that potentially ensure that LLMs do not perpetuate bias in digital information ecosystems; however, hardly any of them can conclusively deal with communication bias.

The AI Act contains several legal requirements—such as regular auditing, transparency, and responsible development practices—to help ensure LLMs do not perpetuate bias in digital information ecosystems, but these measures have notable limitations in directly addressing communication bias. While the AI Act requires pre-market auditing (Arts. 9, 10, 43) to identify and mitigate bias, including the use of diverse and representative training data and technical documentation, its obligations focus mainly on the provider's side and are most robust before LLMs are deployed. While transparency measures (Arts. 13, 53) mandate clear documentation of training data, model architecture, and risks to enhance user awareness, these measures do not directly target or reduce the propensity of LLMs to generate biased outputs. In addition, for GPAI models presenting systemic risk—typically limited to a handful of advanced LLM providers—the AI



Value chain regulation can impose overly strict requirements that may stifle innovation and discourage developers from experimenting with new approaches or technologies, slowing down advances in LLM capabilities.



Act imposes further obligations, including mandatory model evaluation, adversarial testing, systemic risk assessment, serious incident reporting, and adequate cybersecurity protections (Arts. 51, 55). However, these requirements largely retain the AI Act's pre-deployment focus, leaving post-deployment bias mitigation chiefly in the hands of providers rather than enabling ongoing, independent, or user-driven measures. Post-market, the AI Act requires the implementation of monitoring measures (Art. 72), albeit such measures are largely provider-driven and reactive, relying on incident reporting and periodic audits rather than real-time, user-guided bias correction. Deployers have certain operational duties, such as monitoring system usage and reporting incidents (Arts. 26, 61, 73), yet their role in ongoing bias mitigation remains relatively limited and is largely passive rather than proactive. Additional constraints, such as strict data protection rules on processing sensitive data for bias detection (Art. 9 GDPR, Art. 10(5) AI Act), further restrict the scope of alleviating bias. By and large, the AI Act's strongest mechanisms for bias correction converge in the pre-deployment phase, with more reflexive, less specific post-deployment safeguards.^{16,38}

While these strategies are essential for fostering trust in digital information ecosystems, we emphasize the need to further explore how a more balanced environment for content generation and dissemination can be promoted, especially once LLMs have been placed on the market. We address this issue in the next section.

Where Do We Go from Here?

For the most part, the AI Act and the DSA lack provisions that specifically address communication bias in LLMs; where the AI Act and the DSA nonetheless apply, any mitigation of communication bias will typically be a byproduct rather than an intended objective of this legislation. This is partly because communication bias in LLMs can be subtle, multifaceted, and nuanced, rendering it difficult to target and evaluate objectively.^{3,14,27,28} In addition, striking the right bal-

ance between fostering innovation and implementing effective regulation depends on an ongoing evaluation of the impacts of LLMs on society. In the following, we describe some conceptual challenges specific to value chain regulation and content moderation. We then propose putting a greater emphasis on market mechanisms aimed at aligning technological design with societal values as a complement to existing regulations, and suggest several concrete measures to implement our proposal.

Conceptual challenges arising from value chain regulation. Value chain regulation of LLMs seeks to sensibly address the risks associated with AI-driven technologies, but such regulation also presents several challenges that can impact the models' development, deployment, and overall effectiveness. For instance, value chain regulation can impose overly strict requirements that may stifle innovation and discourage developers from experimenting with new approaches or technologies, slowing down advances in LLM capabilities and applications. Owing to the rapid advancement of AI technologies, regulators often struggle to keep pace with new developments, and existing frameworks may quickly become outdated and/or fail to address pertinent issues.² Indeed, compliance with new regulations can be overwhelming, especially for smaller companies or startups. Navigating regulatory obligations may require significant resources, diverting attention from core development activities and potentially leading to reduced competitiveness in the market. A one-size-fits-all attitude can lead to inappropriate obligations that fail to adequately address the harms associated with different use cases.³⁰ Value chain regulation also leads to the possibility that large companies can influence regulatory outcomes to their advantage, leading to provisions that favor established players over smaller actors.³⁴ This can stifle competition within the sector and pose challenges when monitoring compliance across different platforms or applications is required. As LLMs transition from development to widespread deployment, it becomes increasingly critical to consider not

only how the law can mitigate biases before LLMs are released to the market but also how a balanced and pluralistic information environment can be ensured once they are in active use.

Conceptual challenges arising from content moderation. Content moderation is often considered essential in addressing communication bias, but content moderation mechanisms also raise concerns around freedom of expression.¹⁷ This is partly due to the law's ambiguous language and the potential for such rules to be misused by authorities. For instance, the AI Act defines systemic risk as a threat unique to the high-impact capabilities of GPAI models, which can significantly affect the EU market due to their extensive reach or actual or anticipated effects on public health, safety, and fundamental rights, including upon society at large (Art. 3(65)). The DSA, for its part, identifies a broad spectrum of risks, such as the spread of illegal content and negative effects on fundamental rights, civic discourse, and public safety, as well as issues related to gender-based violence and the well-being of minors (Art. 34(1)). There are good reasons to protect these interests; however, it is relatively easy to envisage how the application of such rules might come to restrict rather than promote freedom of expression. For instance, authorities have employed safety, public security, and public health interests as reasons to justify speech restrictions.³⁶ In other instances, concerns around public safety and national security have been raised to defend Internet shutdowns.³⁹ Content moderation rules are often framed in general, sweeping terms and are subject to different interpretations. In addition, enforcement of the provisions is often overseen by a political body—in the EU, the EC via the AI Office—which has been viewed as an additional risk to freedom of expression. Similar issues pertain to algorithmic content moderation mechanisms. Despite their scalability relative to human moderation, the use of algorithms presents distinctive challenges due to the algorithms' ability to produce conflicting predictions, their arbitrariness in output generation, their disparate treatment of differ-

ent datasets, and their overall lack of consistency and predictability.¹⁴ We therefore argue that in addition to addressing overt harms, the law should support ongoing efforts that promote a diversity of viewpoints and equitable access to information, enabling a more balanced digital ecosystem for content generation and dissemination as LLMs become progressively entrenched in public discourse.

Competition and ongoing technology design governance as complements to existing regulations. Value chain regulation and content moderation focus on pre- and post-market decision making relating to harmful content. This may, however, act as a distraction, diverting attention from a closer examination of what the underlying incentives of LLMs are and how they introduce and generate communication bias in the first place. In particular, content moderation focuses on the outputs of LLMs, removing pieces of illegal content and banning individuals or groups that disseminate unwanted information. An assumption underlying content moderation is that social media produces toxic user-generated subject matter because malicious actors exploit technology products to inflict harm on others, while the technology is considered neutral, simply mirroring the undesirable characteristics of users. When it comes to addressing subtle communication bias, however, content moderation rules reveal several shortcomings. Historically, technology companies operated under an approach that emphasized consumer sovereignty, where the focus was on providing users with tools to express their viewpoints freely.²³ In the interim, there has been a notable shift toward an approach in which AI systems implement guardrails to prevent the dissemination of harmful or contentious output. This raises issues about what constitutes harmful content and who decides which output ought to be suppressed. To illustrate, some AI systems have been found to refuse to generate specific information that opposes certain social justice positions while readily producing other material that supports them.²⁷ This presents challenges for existing legal frameworks. Governments

might exert pressure on technology companies, via content-based regulations, (not) to promote particular political agendas, complicating the background of free speech and regulatory oversight. Furthermore, the concentration of media power makes it feasible for authorities to target platforms based on their influence, raising ethical and legal questions about censorship and the role of private companies in public discourse. Content moderation may become polarizing due to the inevitability that particular opinions will be quashed (some, perhaps, to a greater extent than others) because such opinions are liable to violate the policies of certain platforms. This has led, in part, to the recent move by Meta to end its fact-checking programs and to replace professional content moderators with user-generated community notes.¹⁸ Content moderation mechanisms thus face an inherent trade-off—the apprehension is that strict enforcement will affect more posts made by innocent citizens, which will accidentally be taken down, and vice versa. This exposes a fundamental flaw in the expectation that profit-driven platforms can effectively self-regulate, especially as changing societal and political pressures mount.

As AI systems become gatekeepers of information, LLMs may reinforce existing biases and promote particular perspectives. This may lead to a less pluralistic political environment where dissenting opinions are marginalized, making it more difficult for users to encounter diverse viewpoints. The potential for AI outputs to influence public opinion is particularly concerning in closely contested political landscapes. We therefore propose to focus on competition and ongoing technology design governance as a complement to value chain and content-based regulation. Contrary to conventional wisdom, it is well recognized in scientific literature that the properties of technology products help shape the behavior of human actors and that technology products are not blank slates that simply expose users' behavior.^{4,20,23,33} Instead, the design of technology products often drives the behavior of human actors, similar to the way in

which computer code can function as law. Technology design determines, at least to some extent, what users see, the information they divulge, the level of privacy they enjoy, and how they interact with others. Technology design can persuade and subtly nudge individual users toward specific actions and often seeks to maximize engagement rather than encourage behavior that benefits democratic integrity and public trust in particular sources of information.³⁹ Our proposal therefore involves putting a greater emphasis on regulation that helps to effectively readdress the incentives that drive the deployment of AI technologies and ensures, from the outset, that technology platforms proceed in the best interests of consumers.²³

Government regulation that aims to place guardrails on AI companies may suppress specific outputs that are considered harmful or immoral, forcing those companies to assume editorial roles, which may ultimately limit consumer sovereignty. This raises issues about who decides what constitutes harmful content, whether such content reflects a biased worldview that can stifle legitimate discourse, and whether the established guardrails may themselves introduce bias into AI outputs. As the recent move by Meta to end its fact-checking program demonstrates, such regulation may make AI companies even more susceptible to government pressure, because officials might target firms to promote particular agendas.¹⁸ What is more, because of the unique characteristics of generative AI technologies, the imposition of content-related mandates may impede rather than facilitate their ability to produce meaningful and relevant outputs; and the complexity involved in moderating content might contradict, rather than promote, the desired level of discretion and contextual understanding of diverse topics that the development of public opinion typically demands.³⁶

Moreover, if a few large AI companies progressively wield a disproportionate influence over political information, such concentration can lead to a homogenization of perspectives in which alternative points of

view risk becoming marginalized or excluded. Beyond value chain regulation and content moderation—which allow some intervention on the basis of truth or falsity and are required to ensure that communication is not misleading—we highlight the promotion of competition in the AI market. Low barriers to entry can dilute the influence of dominant players and facilitate a more diverse range of outputs, mitigating the risks associated with concentrated power and communication bias in a few LLMs.^{19,40} This is not to neglect the risk that increasing the number of outputs may introduce malicious and bad-faith actors, or to claim that the promotion of competition achieves virtually the same as debiasing. Accentuating these risks, however, conflates two distinct issues: defining the boundaries of public discourse in ways that give proper scope to some indispensable values (i.e., reducing bias at the source through value chain regulation and targeting illicit and harmful outcomes by way of content moderation); and fostering a more diverse public sphere in which a plurality of viewpoints can be articulated and debated. The promotion of competition does not seek to multiply sources of bias but rather aims to ensure that public discourse includes the full range of communicative processes deemed necessary for the formation of public opinion.³² In particular, where the removal of content is inherently contentious because such removal risks being perceived as censorship and presumes that LLMs should act as editorial gatekeepers, competition must play an essential part in addressing communication bias. And competition can further alleviate such bias when multiple models achieve similar overall accuracy but lean in different directions in specific instances, or have a disproportionate effect on some users or groups compared to others.^{5,6,31}

However, promoting competition among multiple AI systems should not simply result in AI companies' pursuit to capture market share; rather, competition should aim toward enabling the fulfillment of a broad spectrum of communicative needs and values in the public sphere. Achieving this objective involves addressing moral

hazard, where technology companies face minimal accountability for the risks they impose on users.²⁶ The moral hazard in communication bias stems from misaligned incentives between corporations seeking to maximize attention and engagement and users expecting to be able to rely on digital ecosystems that provide them with the most accurate and unbiased information.⁸ This means that the promotion of competition must simultaneously incorporate a significant amount of user control in terms of self-governance—that is, where users can have a say in the design of AI systems in terms of data collection/cleaning, model architecture/training, and output evaluation/refinement throughout their entire life-cycle—and help shape public opinion where LLMs can potentially influence communicative action.²³ Where users are allowed to determine, at least in part, the parameters that define the content they wish to generate or access through AI technologies, they are authorized to better express their expectations and viewpoints freely, without undue interference. Affording users a meaningful path of influence through continuous self-governance in synergy with competition will then offer a more robust response to communication bias in LLMs than can be achieved by relying solely on value chain regulation or content moderation. In an environment in which platforms' business models are driven, almost exclusively, by incentives to maximize engagement and attention, user preferences related to diversity, inclusivity, and integrity are routinely sidelined. Establishing a robust infrastructure that ensures meaningful user participation in the ongoing operation of LLMs may then help realign market incentives to permit AI companies to better serve the broader public interest.^{22,26}

Some of the EU Digital Markets Act's (DMA's)^c provisions serve that broader purpose, affording users a greater number of options in the market and compelling firms to internalize the harms resulting from data-driven technologies.²⁵ On the one hand, the DMA establishes a set



The moral hazard in communication bias stems from misaligned incentives between corporations seeking to maximize attention and engagement and users expecting to be able to rely on digital ecosystems that provide them with the most accurate and unbiased information.



of clearly defined criteria to identify gatekeepers—large undertakings providing core platform services (CPSs) such as online intermediation services, online search engines, online social networking services, video-sharing platform services, messaging services, operating systems, web browsers, virtual assistants, cloud computing services, and online advertising services (Arts. 2 and 3). While LLMs are not explicitly referenced as CPSs, they could potentially be included in the DMA's list of CPSs. In addition, LLMs may serve as foundational technologies that enhance or drive such services, such as when they are used in search engines for generating responses or summarizing information. Furthermore, companies that provide LLMs may rely on CPSs to deploy their models more effectively. For instance, an LLM might be integrated into a messaging service or a social media platform to facilitate user interactions. If a company providing LLMs also operates a CPS, it would be subject to the obligations contained in the DMA.^{7,29}


On the other hand, the DMA includes several obligations that can contribute to bias mitigation in LLMs. For instance, under the Act, gatekeepers are prohibited from favoring their services or products over those of competitors to unfairly disadvantage competing services. By prohibiting the practice of self-preferencing, the DMA encourages a level playing field for new technologies—an equal chance to succeed by requiring all actors to operate by the same set of rules—and a more diverse range of AI services to be made available to users, reducing the dominance of a single model that may exhibit particular biases.²¹ The DMA also includes obligations related to how data is combined across different services provided by gatekeepers. These obligations require transparency and user consent when combining data from various sources. By regulating how data is combined and ensuring that users have control over it, the DMA mitigates bias arising from specific datasets. If LLMs are trained on more diverse and representative data, inherent biases in outputs will decrease. Moreover, the

^c <https://bit.ly/40JLa5n>


DMA prohibits gatekeepers from using, in competition with business users, non-public data generated or provided by those business users in the context of their use of the relevant CPSs, including data generated or provided by the customers of those business users. When applied to LLMs, this means that gatekeepers are prohibited from using non-public data to train their AI models and to ensure that all providers have equal access to specific data sources. This further mitigates bias arising from training models on proprietary or exclusive datasets that fail to represent the broader population. Also, by requiring gatekeepers to rely on publicly available data, stakeholders can better scrutinize the sources of information that influence model behavior and can better identify and address potential biases. Preventing gatekeepers from leveraging non-public data will also help level the playing field for LLM developers, to the extent that smaller companies and new entrants can compete effectively against larger firms that might otherwise dominate due to exclusive access to valuable data. Lastly, the DMA mandates that, under certain circumstances, gatekeepers must provide third parties with access to essential data and services. This obligation may be crucial in fostering competition and allowing smaller actors and new entrants to access the data they require in order to develop new LLMs.

Recommendations. Although existing regulations such as the AI Act and DSA focus on mitigating risks and harmful outputs, they often overlook the equally important need to promote diversity (competition) and representativity (self-governance) in the design of digital technologies.¹⁵ Here, we highlight three sets of measures that can better ensure diversity and representativity in ongoing technology design to address communication bias in LLMs.

In the first instance, existing legal provisions should be interpreted and applied with the possibility of communication bias in mind. To illustrate, regular auditing should include assessing how LLMs align with particular perspectives. Developing



Companies should educate users about the potential biases inherent in LLMs and how these biases may affect generated content.




benchmark datasets, such as the GermanPartiesQA,³ can help evaluate the alignment of LLMs with political party positions and detect biases in their responses. Understanding how LLMs exhibit sycophancy—where the models tailor responses to align with user opinions—can help mitigate bias. By conducting prompt experiments that analyze how LLM outputs change based on user prompts, developers can better understand and address the models' propensities. In providing documentation about the training data, model architecture, and decision-making processes of LLMs, companies should educate users about the potential biases inherent in LLMs and how these biases may affect generated content. Ensuring that training data includes a wide range of viewpoints, demographics, and cultural contexts can further mitigate the risk of reinforcing stereotypes. Existing post-deployment obligations establish a regulatory basis for enhancing the responsiveness of AI systems through built-in human oversight tools, intervention protocols, and active monitoring of AI outputs. In addition, the rapid adoption and market impact of LLM-based applications justify the inclusion of LLMs within the regulatory scope of the DMA. Not only could existing provisions be applied to LLMs (especially when operated by designated gatekeepers), but LLMs could also be covered by amending the DMA to explicitly include LLMs as distinct CPSs. This would mandate compliance with DMA obligations and enable the EC to designate LLM providers as gatekeepers if they meet the DMA's quantitative or qualitative thresholds. Finally, current post-deployment mechanisms for mitigating communication bias in LLMs under the AI Act are limited by narrow complaint procedures that do not empower users to meaningfully influence product design or the generation of AI outputs. To institutionalize more effective complaint and investigation procedures, oversight mechanisms should move beyond provider self-assessment and internal monitoring, and instead grant authorities broader investigative and remedial powers. Investigations should

involve establishing external audits, enabling authorities to proactively examine systemic risks, and requiring providers to implement product design changes on behalf of users, where necessary. Drawing on the DSA's framework, vetted external researchers and auditors could be given access to LLMs to assess bias and the effectiveness of proposed mitigation strategies, with findings informing specific interventions. Additionally, authorities should be empowered to act based on user complaints, with the ability to put binding enforcement measures in place so users can articulate and put into effect their apprehensions. This would shift oversight from a pure pre-market approach to an inclusive, participatory, market-centered model that better protects public discourse by addressing all aspects of communication bias in LLMs.

Conclusion

Value chain regulation and content moderation exhibit a tendency to favor snapshot assessments at specific points in time and to focus on the front or back end of the algorithmic design process. Competition, together with an increased level of self-governance, highlights an additional requirement for content generation and dissemination once LLMs have been placed on the market. This is particularly pertinent in terms of bias mitigation, since competition and self-governance establish conditions aimed at facilitating an active search for those LLMs that are the most accurate and least partial, shifting responsiveness upstream to designers and overseers and compelling them to engage in the continuous evaluation and improvement of AI systems. Since the potential of LLMs comes with multiple issues and ambiguities, a number of different regulatory approaches are required to enable citizens to have evenhanded access to the public sphere and effectively partake in the creation of public opinion. We hope this overview offers a distinct perspective and highlights important means by which the development of AI-based technologies can be propelled in useful directions.

Acknowledgments

We would like to thank Volker Stocker, Jan Batzner, the German Research Foundation (DFG), SPP 2378 (project ReNO), 2023–2027, the Weizenbaum Institute for a generous fellowship grant, the HKU Faculty Development Fund, the HKU Seed Fund for Basic Research, project code: 2401102107, and the HKU URC Funding Scheme. 

References


1. Telekom. AI phone: Deutsche Telekom wants to free smartphones from apps (Feb. 2024); <https://bit.ly/3MTDYR6>
2. Almada, M. and Petit, N. The EU AI Act: Between the rock of product safety and the hard place of fundamental rights. *Common Market Law Rev.* 62, 1 (2025), 85–120.
3. Batzner, J., Stocker, V., Schmid, S., and Kasneci, G. GermanPartiesQA: Benchmarking commercial large language models and AI companions for political alignment and sycophancy. In *8th AAAI/ACM Conf. on AI, Ethics, and Society*. AAAI (2025).
4. Bennett Moses, L. and Zalnierute, M. Law and technology in the dimension of time. In *Time, Law, and Change: An Interdisciplinary Study*, S. Ranchordás and Y. Roznai, (Eds.). Hart (2020).
5. Black, E. et al. Less discriminatory algorithms. *Georgetown Law J.* 113, 1 (2024).
6. Black, E., Raghavan, M., and Barocas, S. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM Conf. on Fairness, Accountability, and Transparency*. ACM (2022), 850–863.
7. Bostoen, F. and Krämer, J. AI agents and ecosystems contestability. Issue Paper. CERRE (Nov. 2024).
8. Ennis, S. Consumer expectations and fair contracting for digital products. *CPI Antitrust Chronicle* (Jul. 2022).
9. Epstein, Z. and Hertzmann, A. Art and science of generative AI: Understanding shifts in creative work will help guide AI's impact on the media ecosystem. *Science* 380 (Jun. 16, 2023).
10. European Union Agency for Cybersecurity. ENISA Threat Landscape (Oct. 2023).
11. Federal Bureau of Investigation. Internet Crime Report 2023. Internet Crime Complaint Center (2023).
12. Fernández-Llorca, D., Gómez, E., Sánchez, I., and Mazzini, G., An interdisciplinary account of the terminological choices by EU policymakers ahead of the final agreement on the AI Act: AI system, general purpose AI system, foundation model, and generative AI. *Artificial Intelligence and Law* (2024).
13. Gasser, U. and Mayer-Schoenberger, V. *Guardrails Guiding Human Decisions in the Age of AI*. Princeton University Press (2024).
14. Gomez, J.F., Machado, C., Monteiro Paes, L., and Calmon, F. Algorithmic arbitrariness in content moderation. In *Proceedings of the 2024 ACM Conf. on Fairness, Accountability, and Transparency*. ACM (2024).
15. Gorwa, R. *The Politics of Platform Regulation: How Governments Shape Online Content Moderation*. Oxford University Press (2024).
16. Göksal, Ş.I., Solarte Vasquez, M.C., and Chochia, A. The EU AI Act's alignment within the European Union's regulatory framework on artificial intelligence. *Intern. and Comparative Law Rev.* 24, 2 (2024), 25–53.
17. Hacker, P., Engel, A., and Maurer, M. Regulating ChatGPT and other large generative AI models. [arXiv:2302.02337](https://arxiv.org/abs/2302.02337) (2023).
18. Isaac, M. and Schleifer, T. Meta to end fact-checking program in shift ahead of Trump term. *The New York Times* (Jan. 7, 2025).
19. Jones, A. and Kovacic, W.E. Antitrust's implementation blind side: Challenges to major expansion of U.S. competition policy. *Antitrust Bulletin* 65, 2 (2020), 227–255.
20. Koops, B.J. and Leenes, R. Privacy regulation cannot be hard-coded: A critical comment on the 'privacy by design' provision in data-protection law. *Intern. Rev. of Law, Computers & Technology* 28, 2 (2013), 159–171.
21. Kuenzler, A. Advancing quality competition in big data

markets. *J. of Competition Law and Economics* 15, 4 (2019), 500–537.

22. Kuenzler, A. Competition law as a catalyst for collective market governance: Gauging the discursive benefits of intensified administrative action. *Yearbook of European Law* 41 (2022), 252–278.
23. Kuenzler, A. *Restoring Consumer Sovereignty: How Markets Manipulate Us and What the Law Can Do About It*. Oxford University Press (2017).
24. Kuenzler, A. The shadow of digital antitrust: The law's purported deference to leading actors' product design choices. *Concurrences Rev.* 4, 1 (2022).
25. Kuenzler, A. Third-generation competition law. *J. of Antitrust Enforcement* 11, 1 (2023), 133–141.
26. Kuenzler, A. What competition law can do for data privacy (and vice versa). *Computer Law & Security Rev.* 47, 1 (2022).
27. Ma, T. LLM echo chamber: Personalized and automated disinformation. [arXiv:2409.16241](https://arxiv.org/abs/2409.16241) (Sep. 2024).
28. Margoni, T., Quintais, J., and Schwemer, S. Algorithmic propagation: How the data-platform regulatory framework may increase bias in content moderation. In *Flexibilities in Copyright Law*, C. Sganga and T. Synodinou, (Eds.). Routledge (2025).
29. Martinez, A. Generative AI in check: Gatekeeper power and policy under the DMA. SSRN (2024); https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5025742
30. Martínez Alles, M.G. *Torts and Retribution: The Case for Punitive Damages*. Cambridge University Press (2025).
31. Passi, S. and Barocas, S. Problem formulation and fairness. In *Proceedings of the 2019 Conf. on Fairness, Accountability, and Transparency*. ACM (2019), 39–48.
32. Post, R. Participatory democracy and free speech. *Virginia Law Rev.* 97, 3 (2011), 477–489.
33. Prifti, K. The theory of 'regulation by design': Towards a pragmatist reconstruction. *Technology and Regulation* (2024), 152–166.
34. Schäfer, Q.B. AI, IP, and competition policy: Adjusting policy levers to a new GPT. In *Artificial Intelligence and Competition Policy*, R. Abbott and T. Schrepel, (Eds.). Concurrences (2024).
35. Smuha, N.A. and Yeung, K. The European Union's AI Act: Beyond motherhood and apple pie? In *The Cambridge Handbook of the Law, Ethics and Policy of Artificial Intelligence*, N.A. Smuha, (Ed.). Cambridge (2025).
36. Tushnet, R. Content moderation in an age of extremes. *J. of Law, Technology and the Internet* 10, 1 (2019).
37. U.K. Department for Science, Innovation and Technology. A pro-innovation approach to AI regulation: Government response (Feb. 2024); <https://bit.ly/4cGD0v>
38. Wachter, S. Limitations and loopholes in the EU AI Act and AI liability directives: What this means for the European Union, the United States, and beyond. *Yale J. of Law & Technology* 26, 3 (2024).
39. Wilson, R. and Land, M. Hate speech on social media: content moderation in context. *Connecticut Law Rev.* 52, 1029 (2021).
40. Yasar, A.G. et al. AI and the EU digital markets act: Addressing the risks of bigness in generative AI. [10.48550/arXiv.2308.02033](https://arxiv.org/abs/2308.02033) (Jul. 2023).

Adrian Kuenzler is an associate professor at the Faculty of Law, University of Hong Kong, Hong Kong.

Stefan Schmid (stefan.schmid@tu-berlin.de) is a professor at the Faculty of Electrical Engineering and Computer Science, TU Berlin, Germany, and a principal investigator at the Weizenbaum Institute, Berlin, Germany.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

While general game playing is intellectually engaging and fun, it also serves as a laboratory for practical applications, and provides a theoretical framework for defining rationality in a way that takes into account problem representation, incompleteness of information, and resource bounds.

BY MICHAEL GENESERETH

General Game Playing

GAMES OF STRATEGY, such as chess and checkers, couple intellectual activity with competition. We can exercise and improve our intellectual skills by playing such games. The competition adds excitement and allows us to compare our skills to those of others. The same motivation accounts for interest in computer game playing as a testbed for artificial intelligence (AI). The idea is that programs that think better should be able to win more games, so we can use game playing as

an evaluation technique for intelligent systems.

Unfortunately, building programs to play specific games has limited value in this regard. To begin with, specialized game players have a very narrow focus. They can be good at one game but not another. Deep Blue² may have beaten the world chess champion, but it has no clue how to play checkers. A second, more subtle problem with specialized game-playing systems is they do only part of the work. Most of the interesting analysis and design is done in advance by their programmers. The systems themselves might as well be tele-operated.

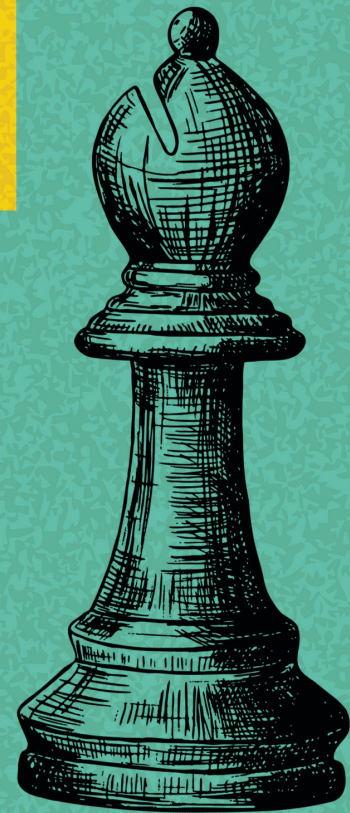
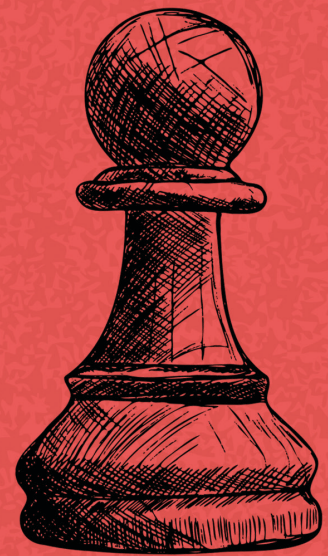
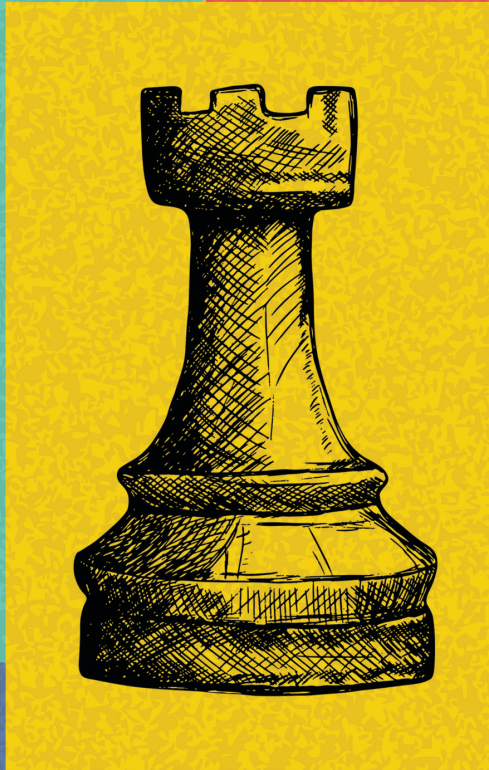
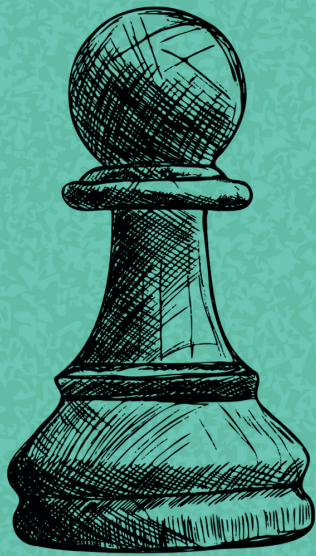
All is not lost. The idea of game playing can be used to good effect to inspire and evaluate good work in AI, but it requires moving more of the design work to the computer itself. This can be done by focusing attention on general game playing.

General game players are systems that accept descriptions of arbitrary games at runtime and use such descriptions to play those games effectively without human intervention. In other words, they do not know the rules until the games start.

Unlike specialized game players, such as Deep Blue² and Stockfish, general game players must be able to play different kinds of games. They must be able to play simple games (like tic-tac-toe) and complex games (like chess), games with differing numbers

» key insights

- **General game players are systems that accept descriptions of arbitrary games at runtime and use such descriptions to play those games effectively without human intervention.**
- **General-game-playing expertise depends on intelligence on the part of the game player and not just intelligence of the programmer of the game player.**
- **General game playing underscores the importance of knowledge representation, reasoning, and rational decision making in a world increasingly focused on machine learning and large language models.**



of players, cooperative games and competitive games, games with or without communication among the players, and so forth. Importantly, they must be able to play games they have never seen before, including games no one has seen before.

Importantly, general game players cannot rely on algorithms designed

in advance for specific games, as in the case of specialized game players. General-game-playing (GGP) expertise depends on intelligence on the part of the game player and not just intelligence of the programmer of the game player.

General game playing underscores the importance of knowledge repre-

sentation, reasoning, and rational decision making in a world increasingly focused on machine learning (ML) and large language models (LLMs). It is worth noting that, despite important and impressive game-related developments in these latter areas (e.g. Bannerjee et al.,¹ Hausknecht et al.,¹¹ Levine,¹⁶ and Silver.²⁵), no ML

Table 1. International General Game Playing Competition winners, 2005 to 2016.

Year	Player	Developer(s)
2005	Cluneplyer	Clune (USA)
2006	Fluxplyer	Schiffel, Thielscher (Germany)
2007	Cadiaplayer	Bjornsson, Finnsson (Iceland)
2008	Cadiaplayer	Bjornsson, Finnsson (Iceland)
2009	Ary	Mehat (France)
2010	Ary	Mehat (France)
2011	TurboTurtle	Schreiber (USA)
2012	CadiaPlayer	Bjornsson, Finnsson (Iceland)
2013	TurboTurtle	Schreiber (USA)
2014	Sancho	Draper, Rose (UK)
2015	Galvanise	Emslie (UK)
2016	WoodStock	Piette (France)

program has ever won an official GGP competition due to time limits on game play, and it is unlikely that systems based on LLMs would perform any better, due to limits on time and space and due to “obfuscation” (wherein words in game descriptions are replaced by nonsense words). By contrast, undergraduates with minimal training in knowledge representation and reasoning methods are able to create GGP programs running on laptops that excel in such competitions and frequently beat humans.

Recent History

The idea of *general problem solving* dates from the earliest days of AI. In 1958, Newell and Simon proposed a *general problem solver*¹⁹ that worked by applying general, task-independent problem-solving processes to descriptions of domain-specific tasks. The first explicit mention of GGP appeared in a subsequent paper by Jacques Pitrat²¹ in 1968. A related notion was described by Barney Pell,²⁰ who further developed the concept in his 1993 doctoral thesis. In the early 2000s, Stanford University researchers enlarged the concept from chess-like games to arbitrary discrete dynamic systems, thus broadening the concept and making it applicable beyond the world of recreational games.

To promote work on GGP, in 2005 the Association for the Advancement of Artificial Intelligence (AAAI) established the International General Game Playing Competition (IGGPC),^{7,8} an annual contest to determine the best automated general game players

in the world. Table 1 shows the winners of the first dozen competitions.

In addition to pitting automated players against each other, the competition frequently included a demonstration match between the competition winner and a human player(s). While the human player won the first of these demonstrations, the computer won all subsequent competitions. For example, in 2012, CadiaPlayer, in addition to defeating the other automated competitors, defeated the human race, represented by Chris Welty, one of the developers of Watson, the IBM computer that beat the best players on the American game show Jeopardy! (As a consolation prize, the human was awarded two bottles of Scotch to ease his disappointment at letting down the human race.)

The International GGP Competition was suspended after 2016. However, by then, the competition had served its primary purpose—it led to workshops on GGP at multiple international conferences and the publication of numerous research papers on GGP and its applications. Today, regional competitions continue to be run, and there are rumors that the international competition will resume, albeit in a slightly different format.

Today, there are GGP sites in multiple countries around the world. The Gamemaster site^a contains a specification of the game description language, games encoded in this language, and software for building players and running matches.

^a <http://gamemaster.stanford.edu>

Figure 1. 2012 Carbon vs. Silicon match: The human player taking counsel from another human.

Game Playing

Despite the variety of games treated in GGP, all games share a common abstract structure. Each game takes place in an environment with finitely many states, with one distinguished initial state and one or more terminal states. In addition, each game has a fixed, finite number of players; each player has finitely many possible actions in any game state, and each state has an associated goal value for each player.

Given this common structure, we can think of a game as a state graph, like the one shown in Figure 2. In this case, we have a game with one player, with eight states (named s_1, \dots, s_8). The arcs in this graph capture the transition function for the game. For example, if the game is in state s_1 and a player performs action a , the game will move to state s_2 . If the game is in state s_1 and a player performs action b , the game will move to state s_5 . Each game has just one initial state, in this case s_1 . However, there can be any number of terminal states, in this case s_4 and s_8 . The numbers associated with each state indicate the utilities of those states for the player. Players earn those scores only in terminal states. However, they are provided for all states and in some games indicate incremental progress.

We can extend this model to accommodate games with multiple players with two modifications.

1. We annotate each state with rewards for all players in the game. (The rewards can be the same or different for different players.)

2. For each state, we specify which player is in control, that is, whose turn it is to play. Turns need not strictly al-

ternate; in some cases, a single player may get several turns in a row. (This model can be extended to games with simultaneous moves, but for the sake of simplicity, we avoid that complexity in what follows.)

Since all the games we are considering are finite, it is possible, in principle, to describe such games in the form of state graphs. Unfortunately, such explicit representations are not practical in all cases. Even though the numbers of states and actions are finite, these sets can be extremely large; and the corresponding graphs can be larger still. For example, in chess, there are more than 10^{40} possible states.

In the vast majority of games, states and actions have a composite structure that allows us to define a large number of states and actions

in terms of a smaller number of more fundamental entities. In chess, for example, states are not monolithic; they can be conceptualized in terms of pieces, squares, rows, columns, diagonals, and so forth. By exploiting this structure, it is possible to encode games in a form that is more compact than direct representation.

The first step in solving this problem is to conceptualize states as *datasets* (as sets of facts that are true in those states) and to conceptualize actions as operations applied to arguments, as suggested by the graph in Figure 3.

Given a “structured” state graph of this sort, the second step is to encode game rules in terms of these datasets and actions. In GGP, the most popular game description language is GDL (for Game Description Language).¹⁷ Rules

in GDL are written in Epilog,⁹ a dynamic logic programming language based on Prolog. The following are the rules of tic-tac-toe as written in GDL:

```

role(x)
role(o)

init(cell(1,1,b))
init(cell(1,2,b))
init(cell(1,3,b))
init(cell(2,1,b))
init(cell(2,2,b))
init(cell(2,3,b))
init(cell(3,1,b))
init(cell(3,2,b))
init(cell(3,3,b))
init(control(x))

legal(mark(M,N)) :- cell(M,N,b)

mark(M,N) :: control(R) ==>
    cell(M,N,R) & ~cell(M,N,b)
mark(M,N) :: control(x) ==>
    ~control(x) & control(o)
mark(M,N) :: control(o) ==>
    ~control(o) & control(x)

line(Z) :- row(M,Z)
line(Z) :- column(M,Z)
line(Z) :- diagonal(Z)
row(M,X) :- cell(M,1,X) &
    cell(M,2,X) & cell(M,3,X)
column(N,X) :- cell(1,N,X) &
    cell(2,N,X) & cell(3,N,X)
diagonal(X) :- cell(1,1,X) &
    cell(2,2,X) & cell(3,3,X)
diagonal(X) :- cell(1,3,X) &
    cell(2,2,X) & cell(3,1,X)

goal(x,100) :- line(x) & ~line(o)
goal(x,50) :- ~line(x) & ~line(o)
goal(x,0) :- ~line(x) & line(o)
goal(o,100) :- ~line(x) & line(o)
goal(o,50) :- ~line(x) & ~line(o)
goal(o,0) :- line(x) & ~line(o)

terminal :- line(x)
terminal :- line(o)
terminal :- ~open

open :- true(cell(M,N,b))
    
```

Figure 2. Game as a state graph.

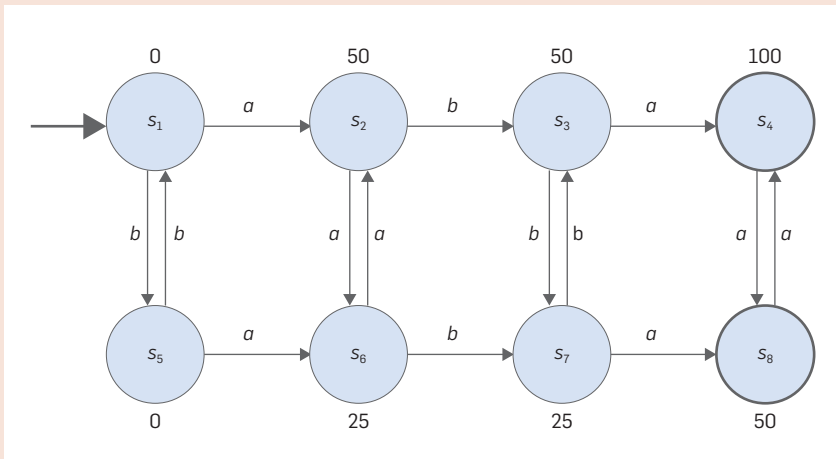
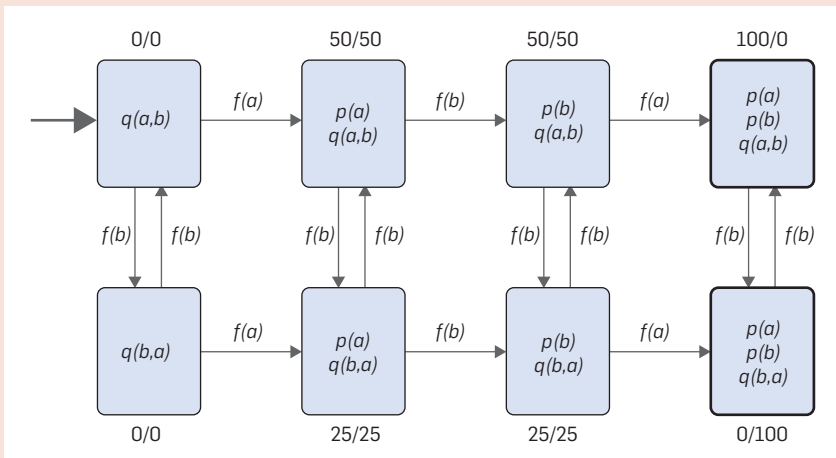


Figure 3. Game as a “structured” state graph.



```

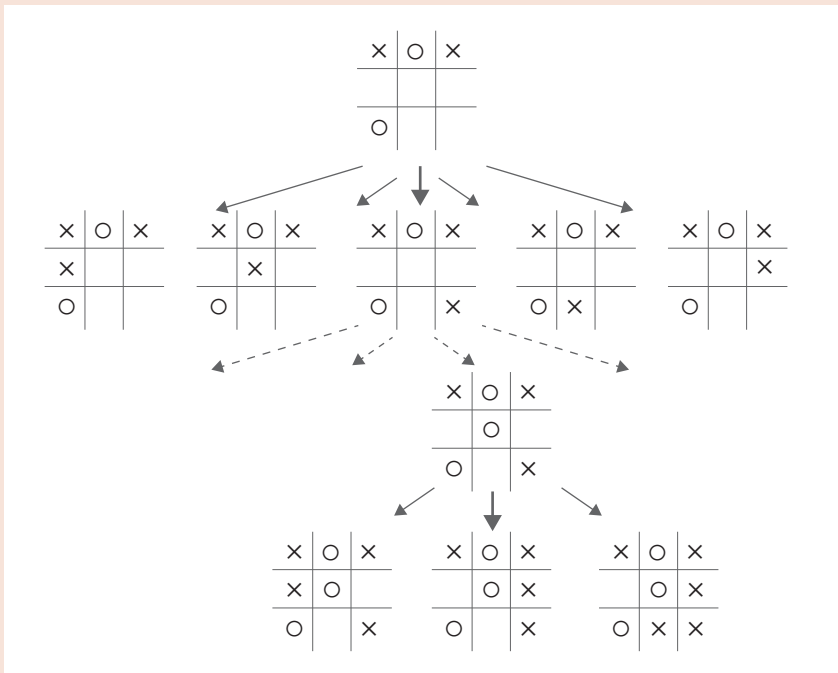
goal(x,100) :- line(x) & ~line(o)
goal(x,50) :- ~line(x) & ~line(o)
goal(x,0) :- ~line(x) & line(o)
goal(o,100) :- ~line(x) & line(o)
goal(o,50) :- ~line(x) & ~line(o)
goal(o,0) :- line(x) & ~line(o)

terminal :- line(x)
terminal :- line(o)
terminal :- ~open

open :- true(cell(M,N,b))
    
```

There are two players: *x* and *o*. In the initial state, all cells are blank (represented here by the symbol *b*), and the *x* player has control. It is legal for the player in control to mark a cell if that cell is blank. (In GDL, symbols that begin with capital letters are

Figure 4. Partial game tree for tic-tac-toe.



variables, while symbols that begin with lowercase letters are constants. The :- operator is read as “if”—the expression to its left is true if the expressions that follow it are true.) If the player in control marks a cell, that cell contains the player’s mark in the next state and the b is removed. Also, control alternates on each play. (The :: operator relates the action mentioned to its left as a transition rule that applies whenever the conditions before the ==> are true and produces a state in which the conditions to its right are true.) A line is a row of marks, a column of marks, or a diagonal. The x player / o player receives 100 points in any state with a line of x marks / o marks, it receives 50 points in any state with no lines, and it gets 0 points in any state with a line of o marks / x marks. Finally, a game terminates whenever there is a line of player marks or if the game is no longer open, that is, there are no blank cells.

The main thing to note about this example is that one page of rules fully describes a game of thousands of states. That is a significant savings over the state graph for tic-tac-toe (which contains more than 5,000 states). The compression in more complex games can be even more dra-

matic. For example, it is possible to describe the rules of chess in just four pages of rules like the ones above.

Interestingly, to prevent programmers from building in specialized capabilities for specific words in game descriptions, it is common in GGP to *obfuscate* descriptions. All words are consistently replaced by nonsense words. The only exceptions are variables and game-independent constants, such as role, legal, terminal.

Game Playing

Having a formal description of a game is one thing; being able to use that description to play the game effectively is something else entirely. The player must be able to compute the initial state of the game. It must be able to compute which moves are legal in every state. It must be able to determine the state resulting from particular moves. It must be able to compute the value of each state for each player. And it must be able to determine whether any given state is terminal.

From a description such as this, a general game player can reconstruct a game tree. See the partial game tree shown in Figure 4. The player starts with the initial state, computes the legal moves in that state, and for each move deduces the next state. This is

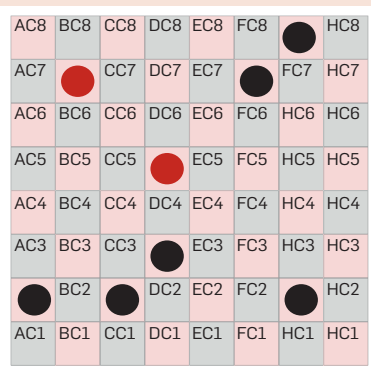
done repeatedly to expand the tree until a terminal state is reached on each branch. Given a game tree, a player can use minimax²² or an equivalent technique to determine its best possible move, and it can use sophisticated search techniques, such as alpha-beta pruning, to save work.¹⁴

Unfortunately, it is not always possible to search to the end of the game tree. In tic-tac-toe, there are thousands of states. This is large but manageable. In chess, there are more than 10⁴⁰ states.²⁴ A state space of this size, being finite, is fully searchable in principle but not in practice. Moreover, the time limit on moves in most games means players must select actions without knowing with certainty whether they are the best or even good moves to make.

The alternative is to do incomplete search, on each move expanding the game tree as much as possible within the available time and then making a choice based on the estimated values of non-terminal states. In traditional game playing, where the rules are known in advance, the programmer can invent game-specific evaluation functions to help in this regard. For example, in chess, we know that states with higher piece count and greater board control are better than ones with less material or lower control. Unfortunately, it is not possible for a GGP programmer to invent such game-specific rules in advance, as the game’s rules are not known until the game begins. The program must evaluate states for itself. Doing this effectively is the key to effective general game playing.

The approach used in early GGP programs was to develop game-independent heuristics,^{3,13,15,23} for example, proximity to a goal state, player mobility, and opponent restriction. Consider mobility. Proponents argue that, all other things being equal, it is better to move to a state that affords the player greater mobility and offers more possible actions than to be boxed into a corner. Symmetrically, proponents of mobility argue that it is good to minimize the mobility of one’s opponents. All of these heuristics have been shown to be effective in some games. Unfortunately, they are only heuristics. They frequently

Figure 5. Position in final match of IGGPC 2006.



fail, sometimes with comical consequences.

The final match of IGGPC 2006 is an example. The game was cylinder checkers, that is, checkers played on a cylinder. In this game, as in ordinary checkers, a player is permitted to move one of their ordinary pieces (pieces that are not kings) one square forward on each turn. In the position shown in Figure 5, red is moving from top to bottom, and black is moving from bottom to top. If a piece is blocked by an opponent's player, he can "jump" that player if there is an empty square on the other side. Moreover, the player must make such a jump if one is available. The objective of the game is to take all or as many of the opponent's pieces as possible while preserving one's own pieces. Here is a snapshot of the game. It is red's turn to play. What should it do? And what do you think it did?

Here's a hint. The player in this case was Cluneplayer,³ and it had decided, for some reason or other, that limiting the opponent's mobility was a good heuristic. If it were to move the rearmost piece, black would have multiple moves. However, if it were to move the piece in front, black would be forced to capture its piece. In other words, it would have at most one move. Clearly, moving the forward piece minimizes the opponent's mobility, so that is what Cluneplayer did. Actually, the whole match played out this way, with red giving black captures at every opportunity. It was sad to watch but also a little comical. The moral is that, while non-guaranteed heuristics are sometimes useful, they are not always useful.

Monte Carlo Search

The second generation of GGP programs used a different approach to evaluating non-terminal states, notably Monte Carlo Search and Monte Carlo Tree Search.

In Monte Carlo Search (MCS), the player expands the tree a few levels. Then, rather than using a local heuristic to evaluate a state, it makes some probes from that state to the end of the game by selecting random moves for all players. It sums up the total rewards for all such probes and divides by the number of probes to obtain an estimated utility for that state. It can then use these expected utilities in comparing states and selecting actions.

Monte Carlo Tree Search (MCTS)⁶ (Figure 6) improves on this by replacing the random choice with more intelligent selections. The result of introducing such techniques was dra-

matic. Suddenly, automated general game players began to perform at a high level. Using MCTS, Cadioplayer won the competition three times.

The use of statistical techniques changed the character of general game players from curiosities to programs capable of serious play. Virtually every general game playing program today includes some variant of MCS or MCTS. Unfortunately, statistical search alone has weaknesses, notably on games with complex descriptions requiring a lot of computation time and on games where there are few goal states; and, consequently, it must be combined with other methods to be successful.

Metagaming

Whereas the GGP techniques described above concentrated on game tree search performed at *runtime*, sub-

Figure 6. Monte Carlo Tree Search.

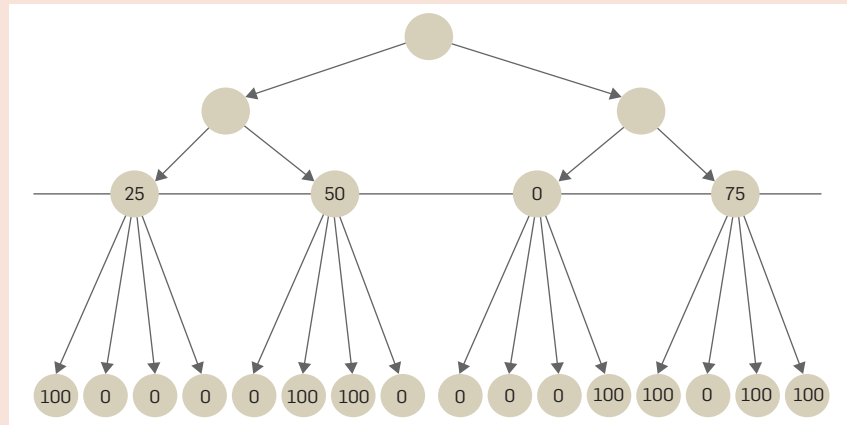
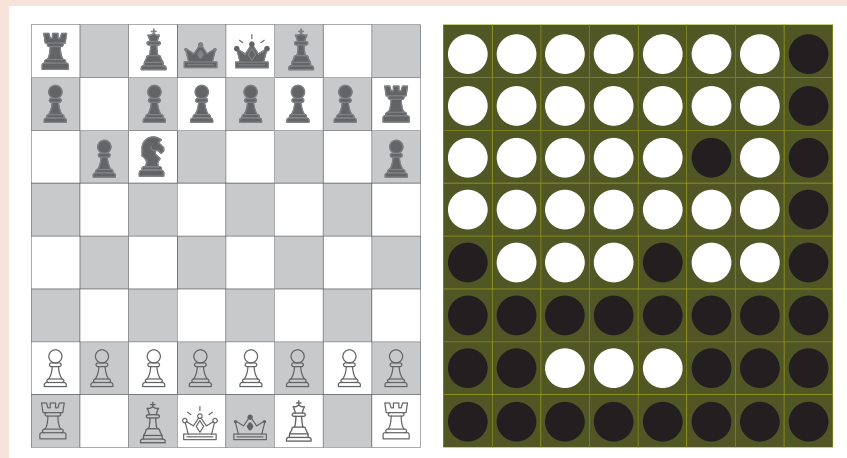


Figure 7. Hodgepodge.




sequent work was focused on *offline* processing of game descriptions. Examples include things like factoring games into independent subgames, reformulating game descriptions, and automatic programming. Often, it is the case that this sort of processing more than pays for itself. In such cases, players can expend a little time up front and gain a lot in processing time later.

This is really what programmers do when building specialized game players, and now we are building game players that can do these things for themselves. And this is what GGP was intended to stress from its very beginnings.


One example of offline analysis is game decomposition, also called *factoring*.^{4,10} Consider the game of Hodgepodge pictured in Figure 7. Hodgepodge is actually two games glued together. Here we show chess and othello, but it could be any two games. One move in a joint game of Hodgepodge corresponds to one move in each of the two constituent games. Winning requires winning at least one of the two games while not losing the other.

What makes Hodgepodge interesting is that it is factorable, that is it can be divided into two independent games. Realizing this can have dramatic benefit. To see this, consider the size of the game tree for hodgepodge. Suppose one game tree has branching factor a and the other has branching factor b . Then the branching factor of the joint game is a times b , and the size of the fringe of the game tree at level n is $(a*b)^n$. However, the two games are independent. Moving in one subgame does not affect the state of the other subgame. So, the player really should be searching two smaller game trees, one with branching factor a and the other with branching factor b . In this way, at depth n , there would be only $a^n + b^n$ states. This is a huge decrease in the size of the search space. Luckily, in many cases, it is possible to discover such factors in time proportional to the size of the game description.

Table 2 illustrates the computational benefits of factoring on three highly factorable games: *multiple buttons and lights*, *multiple switches*, and



The interesting thing about general game playing is that sometimes the cost of analysis is proportional to the size of the description rather than the size of the game tree.



multiple tic-tac-toe. The first column in Table 2 identifies the game, the second column indicates the depth to which the tree is searched, the third column lists the number of milliseconds to search the game tree to that depth without factoring, and the last column shows the number of milliseconds after factoring.

Factoring is just one example of game reformulation. There are many others. For example, it is sometimes possible to find symmetries in games that cut down on search space. In some games, there are bottlenecks that allow for a different type of factoring. Consider, for example, a game made up of one or more subgames in which it is necessary to win one game before moving on to a second game. In such a case, there is no need to search to a terminal state in the overall game; it is sufficient to limit search to termination in the current subgame. These examples are extreme cases, but there are many simpler everyday examples of finding structure of this sort that can help in curtailing search.

The trick in metagaming is to analyze and/or reformulate a game description without expanding the entire game tree. The interesting thing about general game playing is that sometimes the cost of analysis is proportional to the size of the description rather than the size of the game tree, as in the example mentioned above. In such cases, players can expend a little time and gain a lot in search savings.

Of course, factoring games and finding transformations such as these requires a lot of algorithmic expertise. What we really need is Donald Knuth in a box. Or maybe Corman and Leiser. And, of course, since we are talking about games, we should have expertise at game tree search. So, we had better cram Jonathan Schaeffer into our box as well.

An important point here is that GGP is not just about game tree search; to an even greater extent, it is about game descriptions and their use, and ultimately about automatic programming.

Conclusion

General game playing is an interesting application in its own right. It

Table 2. Computational benefits of factoring on three highly factorable games.

Game	Depth	Unfactored Cost	Factored Cost
Multiple Buttons and Lights	4	16,700	6
Multiple Switches	5	22M	210
Multiple Tic-Tac-Toe	3	88,000	150

is intellectually engaging and more than a little fun. But it is much more than that. It serves as a laboratory for practical applications, in business and law, science and engineering. In fact, some of the games used in competitions are drawn from such areas. More fundamentally, it provides a theoretical framework for defining rationality in a way that takes into account problem representation, incompleteness of information, and resource bounds. The upshot is that it raises questions about the nature of intelligence and serves as a laboratory in which to evaluate competing approaches to intelligence.

General game playing is a setting within which AI is the essential technology. It concentrates attention on the notion of runnable specifications. Building systems of this sort dates from the early years of AI.

It was in 1958 that John McCarthy invented the concept of the “advice taker.”¹⁸ The idea was simple. He wanted a machine that he could program by description. He would describe the intended environment and the desired goal, and the machine would use that information to determine its behavior. There would be no programming in the traditional sense. McCarthy presented his concept in a paper that has become a classic in the field of AI:

The main advantage we expect the advice taker to have is that its behavior will be improvable merely by making statements to it, telling it about its environment and what is wanted from it. To make these statements will require little, if any, knowledge of the program or the previous knowledge of the advice taker.

An ambitious goal! But that was a time of high hopes and grand ambition. The idea caught the imagina-

tions of numerous subsequent researchers—notably Bob Kowalski, the high priest of logic programming, and Ed Feigenbaum, the inventor of knowledge engineering. In a paper written in 1974, Feigenbaum gave his most forceful statement of McCarthy’s ideal:⁵

The potential use of computers by people to accomplish tasks can be “one-dimensionalized” into a spectrum representing the nature of the instruction that must be given the computer to do its job. Call it the what-to-how spectrum. At one extreme of the spectrum, the user supplies his intelligence to instruct the machine with precision exactly how to do his job step-by-step ... At the other end of the spectrum is the user with his real problem ... He aspires to communicate what he wants done ... without having to lay out in detail all necessary subgoals for adequate performance.


Some have argued that the way to achieve intelligent behavior is through specialization. That may work as long as the assumptions one makes in building such systems are true. For general intelligence, however, general intellectual capabilities are needed, and such systems should be capable of performing well in a wide variety of tasks. In the words of Robert Heinlein:¹²

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.

References

- Bannerjee, B., Kuhlmann, G., and Stone, P. Value function transfer for general game playing. In *Proceedings of the ICML Workshop on Structural Knowledge Transfer for Machine Learning* (2006).
- Campbell, M., Hoare, A.J., and Hsu, F.-H. Deep Blue. *Artificial Intelligence* 134, 1-2 (2002), 57–83.
- Clune, J. Heuristic evaluation functions for general game playing. In *Proceedings of the 22nd AAAI Conf. on Artificial Intelligence*. AAAI (2007), 1134–1139.
- Cox, E., Schkufza, E., Madsen, R., and Genesereth, M. Factoring general games using propositional automata. In *Proceedings of the IJCAI-09 Workshop on General Game Playing (GIGA-09)*. IJCAI (2009).
- Feigenbaum, E. *ACM Turing Award Lectures*. Association for Computing Machinery (2007).
- Finnsso, H. and Björnsson, Y. Simulation-based approach to general game playing. In *Proceedings of the 23rd AAAI Conf. on Artificial Intelligence*. AAAI (2008), 259–264.
- Genesereth, M. and Björnsson, Y. The International General Game Playing Competition. *AI Magazine* 34, 2 (2012).
- Genesereth, M., Love, N., and Pell, B. General game playing: Overview of the AAAI competition. *AI Magazine* 26 AAAI (2005), 62–72.
- Genesereth, M.R. and Chaudhri, V.K. *Introduction to Logic Programming*. Springer (2020).
- Gunther, M., Schiffel, S., and Thielscher, M. Factoring general games. In *Proceedings of the IJCAI-09 Workshop on General Game Playing*. IJCAI (2009).
- Hausknecht, M., Khandelwal, P., Miikkulainen, R., and Stone, P. HyperNEAT-GGP: A HyperNEAT-based Atari general game player. In *Proceedings of the 14th Annual Conf. on Genetic and Evolutionary Computation* (2012).
- Heinlein, R.A. *Time Enough for Love*. Ace Books (1988).
- Kirci, M., Sturtevant, N.R., and Schaeffer, J. A GGP feature learning algorithm. *Kunstliche Intelligenz* 25, 1 (2011), 35–41.
- Knuth, D.E. and Moore, R.W. An analysis of alpha-beta pruning. *Artificial Intelligence* 6, 4 (1975), 293–326.
- Kuhlmann, G. and Stone, P. Automatic heuristic construction in a complete general game player. In *Proceedings of the 21st AAAI Conf. on Artificial Intelligence*. AAAI (2006), 1457–1462.
- Levine, J. et al. General video game playing. In *Proceedings of the Dagstuhl Seminar on Artificial and Computational Intelligence in Games* (2013), 1–7.
- Love, N., Hinrichs, T., and Genesereth, M. Game Description Language Specification. Stanford University (2006).
- McCarthy, J. Programs with common sense. In *Proceedings of the Symp. on Mechanization of Thought Processes*. National Physical Laboratory (1958).
- Newell, A., Shaw, J.C., and Simon, H.A. Report on a general problem-solving program. In *Proceedings of the Intern. Conf. on Information Processing*. (1959), 256–264.
- Pell, B. *Strategy Generation and Evaluation for Meta-Game Playing*. Trinity College (1993).
- Pitrat, J. Realization of a general game-playing program. *IFIP Congress* 2 (1968), 1570–1574.
- Russell, S.J. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson (2021).
- Schiffel, S. and Thielscher, M. Fluxplayer: A successful general game player. In *Proceedings of the 22nd AAAI Conf. on Artificial Intelligence*. AAAI (2007), 1191–1196.
- Shannon, C.E. Realization of a general game-playing program. *Philos. Mag.* 41, 314 (1950), 256–275.
- Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* 362, 6419 (2018), 1140–1144.

Michael Genesereth (genesereth@stanford.edu) is a professor in the Computer Science Department at Stanford University and a professor by courtesy in the Stanford Law School, Stanford, CA, USA.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

Leveraging ML models to directly learn network flow configurations from empirical data can deliver robustly high performance, surpassing that of demand-prediction methods.

BY YARIN PERRY, SRIKANTH KANDULA, ISHAI MENACHE,
MICHAEL SCHAPIRA, AND AVIV TAMAR

Learning to Flow (Between Datacenters)

DUE TO ITS fundamental importance and wide-ranging applications, flow optimization is one the most well-studied topics in computer science. A key example of its practical significance is *traffic engineering (TE)*. Companies such as Microsoft, Google, Amazon, Meta, and Netflix invest enormous resources into high-capacity private “backbone networks” that interconnect providers’ datacenters to one another, and to external networks.^{3,9–11,18} Figure 1 illustrates Microsoft’s backbone network, which spans multiple locations across the globe. To manage the ever-increasing traffic demand, providers constantly optimize network flow configurations.^{7,9,11,12,18,19,23,31}

The state of the art involves periodically solving a logically centralized flow-optimization problem to meet a target optimization objective—for example, maximizing the total flow of traffic across the network, or minimizing network congestion.

But there is a major gap between the theory and practice.^a A standard assumption in flow-optimization theory is that traffic demands between network nodes are known *in advance*. In practice, such information is rarely available, and flow optimization is performed under *uncertainty*. While exact knowledge of traffic demands is elusive, network traffic typically does exhibit spatial and temporal patterns. A common approach to addressing this gap is thus *predicting* the upcoming traffic demands and optimizing for the *predicted* demands.¹⁸ As we will illustrate, this can yield suboptimal performance when network traffic is volatile. A theoretically appealing alternative, (*traffic-*) *oblivious routing*, optimizes for the *worst-case* demands.^{4,6,28} In this case, however, disregarding *all* informa-

a This article is based on a conference publication, which appeared at NSDI 2023.²⁷

» key insights

- **Flow optimization is a foundational problem in computer networks, yet a major gap persists: Theory assumes traffic demands are known in advance, while practice must grapple with high uncertainty.**
- **This gap is acute in private backbone networks connecting global datacenters, where volatile customer-facing traffic makes reliable forecasting of upcoming demands difficult.**
- **Modern large-scale stochastic optimization and machine learning methods can be leveraged to sidestep fragile traffic prediction by directly producing robust, high-quality flow configurations.**
- **The approach is theoretically grounded and practically impactful, yielding near-optimal flow and order-of-magnitude runtime gains on empirical datasets from production networks.**



tion about network traffic patterns also results in far-from-optimal performance.¹⁹

We propose a novel approach: *direct optimization for traffic engineering (DOTE)*. Instead of predicting traffic demands and optimizing for these, in DOTE empirical data is used to learn a *direct mapping* from recent traffic conditions to a network-wide flow configuration. More specifically, DOTE enables providers to harness the wealth of empirical data at their disposal to train a deep neural network (DNN) to generate flow configurations that yield *robustly high* performance. Our approach leverages recent advances in large-scale DNN-based optimization. We have shown, through extensive

evaluation on empirical datasets from production networks, that DOTE significantly outperforms the state of the art in terms of solution quality. Furthermore, by relying on DNN inference for real-time decision making, DOTE achieves online runtimes one to two orders of magnitude faster than today's schemes, which involve solving a large linear program (LP) in real time.

To gain a deeper theoretical understanding of DOTE, here we examine the geometric properties of the class of flow-optimization problems it addresses. We prove that these problems possess desirable characteristics for (stochastic) optimization, such as quasiconcavity. Building on these insights, along with

algorithmic knowledge from convex optimization and machine learning (ML) theory, we prove that an idealized DOTE training process achieves optimal performance. The empirical success of our implementation of DOTE, which approximates this idealized process, further supports this claim.

Many computer systems involve decision making under uncertainty about the environment, with external inputs not fully known but displaying patterns (such as traffic demands, in our setting). Examples include different flavors of resource allocation, scheduling, and load balancing, domains in which we believe our approach and technical insights may find additional applications.

Optimizing Flow Under Demand Uncertainty

Here, we outline our framework for flow optimization under traffic-demand uncertainty. To simplify exposition, we henceforth restrict our attention to the classical maximum-multicommodity-flow objective, which captures the natural desideratum of maximizing the total flow across the network. Our approach and results extend to other well-studied optimization objectives, including minimizing network congestion and optimizing

various notions of fairness—for example, maximum concurrent flow and α -fairness.^{4–6,13,14,20–22,24,28}

Consider a capacitated network graph $G = (V, E, c)$, where V and E represent the nodes and communication links, respectively, and each edge $e \in E$ has capacity $c(e) > 0$. For every node pair $i, j \in V$, there exists a set of *permissible paths* P_{ij} that traffic between i and j can follow. In practice, private backbone networks typically allow a small number of permissible paths (e.g., 6–8) to prevent state over-

load in network devices. Figure 2a illustrates a simple example with two permissible paths for each node pair (1,2), (2,3), and (4,5), colored blue, red, and green, respectively. All link capacities are 1. In classical flow optimization, the input also includes a demand matrix \mathbf{D} , where the (i,j) entry $d_{ij} \geq 0$ specifies the communication demand between source i and destination j .

In practice, demand matrices are rarely exactly known when determining a flow configuration, and can only be measured *after the fact*. To address this, a flow configuration in our framework specifies, for each permissible path p , a *flow cap* ω_p , representing the maximum flow path p is allowed to carry. In our example, a possible flow configuration is assigning flow caps of $\frac{3}{4}$ and $\frac{1}{4}$ to the shortest and alternate paths for each source-destination pair, respectively, shown in Figure 2b. We require that in any flow configuration $\vec{\omega} = \{\omega_p\}$, the flow caps ω_p be such that no link capacity is exceeded even if all caps are reached (as in our example flow configuration), thus preventing capacity violations *regardless of realized traffic demands*.

A flow configuration $\vec{\omega}$ thus only *upper bounds* the volume of flow each network path can carry. The *actual* flow for each source-destination pair is induced by the specific realization of traffic demands. For example, consider the flow caps shown in Figure 2b, and the realized demands $\frac{1}{2}$, $\frac{1}{2}$, and $1\frac{1}{2}$, for the pairs (1,2), (2,3), and (4,5), respectively, with all other realized demands being 0. Network capacity is sufficient for carrying all demands.

Figure 1. Microsoft's backbone network.



Figure 2. Flow optimization example.

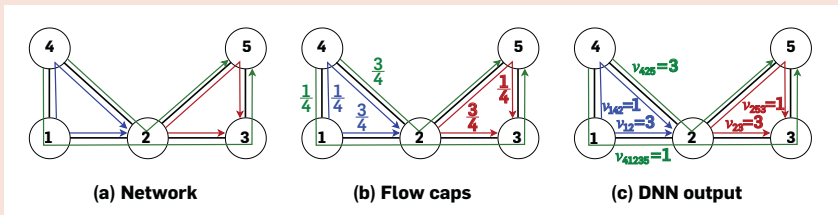
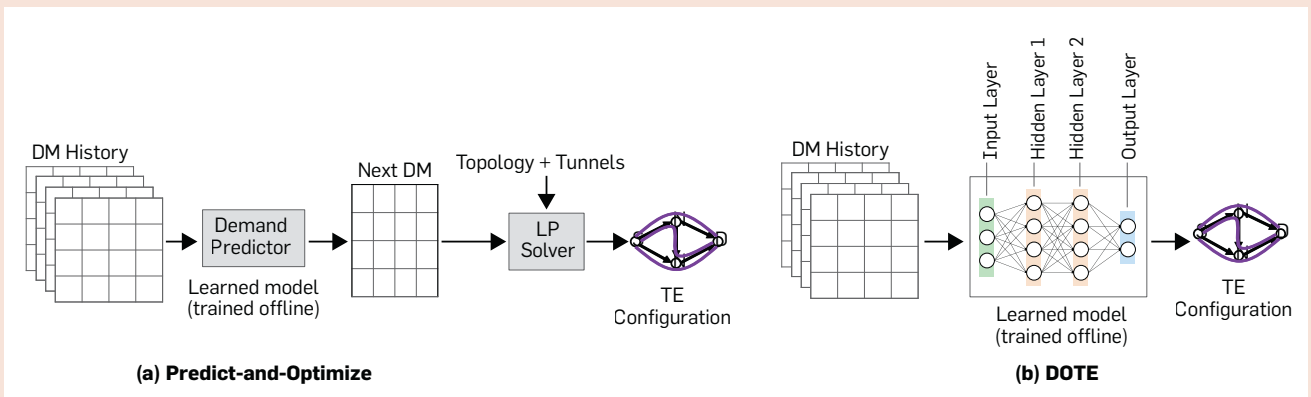


Figure 3. The key differences between predict-and-optimize and DOTE architectures.



However, only 1 unit out of (4, 5)'s demand of $1\frac{1}{2}$ will be delivered because its flow caps total 1, whereas the flow for each of the pairs (1, 2) and (2, 3) is $\frac{1}{2}$ (its demand), “wasting” half of its assigned capacity.^b In general, the flow f_{ij} for source-destination pair (i, j) under flow configuration \vec{w} and realized demand matrix \mathbf{D} is the minimum of the total flow cap on its permissible paths and its realized demand d_{ij} . When maximizing flow under demand uncertainty, the goal is to select a flow configuration \vec{w} that maximizes $F(\vec{w}, \mathbf{D}) = \sum_{i,j} f_{ij}(\vec{w}, \mathbf{D})$ with respect to the realized future demand matrix \mathbf{D} .

In our framework for flow optimization under demand uncertainty, a centralized *controller* monitors network traffic and periodically optimizes flow for a target objective (e.g., maximum flow). At fixed intervals (e.g., every five minutes at some large service providers), the controller takes as input the network topology, permissible paths, and recent traffic demand history (e.g., 12 demand matrix snapshots from the previous hour) and computes a flow configuration $\vec{w}^{(t)}$ for the next time interval. The realized demand matrix \mathbf{D} for this interval is only observed when the interval is concluded. When targeting the maximum-flow objective, the controller aims to maximize the average total flow across all intervals $\frac{1}{T} \sum_{t=1}^T F(\vec{w}^{(t)}, \mathbf{D}^{(t)})$, where T is the optimization period.

Predict-and-Optimize and Its Pitfalls

A common approach to flow optimization under demand uncertainty is *predict-and-optimize*, where a trained predictor infers the upcoming demand matrix from recent traffic conditions (and optionally other factors like time of day). This predicted demand, along with the network topology and permissible paths, is then used to compute an optimal flow configuration by solving a linear program (LP) (see Figure 3a). Next we discuss three drawbacks of this approach.

^b We assume that, in situations like this, where the demand for a source-destination pair (i, j) is strictly less than its aggregate assigned capacity, there is some specified way for splitting the flow between the permissible paths. Our results are independent of the specific choice of such a rule.

Figure 4. Variability in traffic demands for inter-datacenter traffic and customer-facing traffic across weeks.

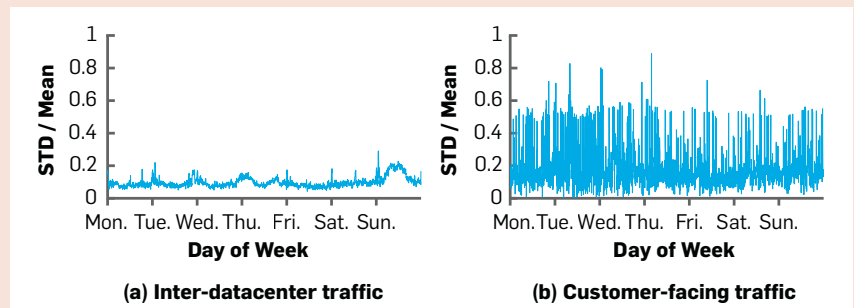


Table 1. DOTE vs. demand predictor.

	Training Data	Performance Metric	Output
Demand Prediction	Empirically observed sequences of past DMs	Prediction loss (e.g., RMSE)	Demand matrix
DOTE		The end-to-end TE objective	TE configuration

Customer-facing traffic is hard to predict. Traffic in private backbone networks can be divided into traffic between datacenters, for example geo-replication of datasets and search index updates, and customer-facing traffic, including Web traffic, videos, and so on. The latter constitutes the majority of traffic in some large service providers’ backbone networks.¹⁸ Figure 4a plots the standard deviation in *inter-datacenter traffic* demands, normalized by the mean, across 11 consecutive weeks, for the busiest datacenter pair. Figure 4b shows the same for *customer-facing traffic* over four weeks. Observe that customer-facing traffic is much more volatile. This can be attributed to flash-crowds causing surges in customer requests and congestion on communication links toward external destinations. The unpredictability of customer-facing traffic challenges flow optimization; inaccurate predictions can lead to wasted capacity by reserving resources for overestimated demands, and to insufficient capacity for underestimated demands.

Possible objective mismatch between predictor loss and flow-optimization goals. The demand predictor minimizes a loss metric, such as root mean square error (RMSE), over the difference between predicted and realized demand matrices. This, however, can misalign with the goal

of generating high-performance flow configurations. For instance, RMSE treats all source-destination demand errors equally, although errors on specific demands that are more problematic to carry on the specific network topology can exert a disproportionately large effect on flow quality.

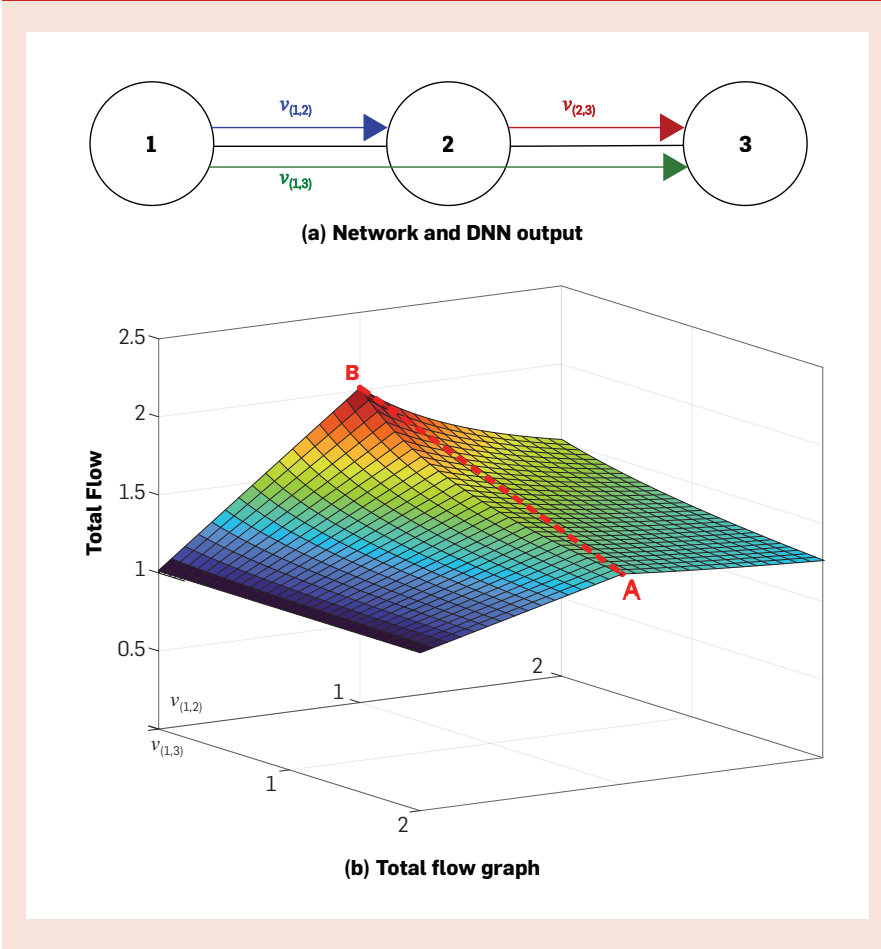
Solving large LPs is slow. LP solver runtimes on large networks might take minutes,^{2,25} slowing sequential flow optimization and hindering responses to traffic changes or network failures.

Learn the Output, Not the Input!

To overcome the limitations of predict-and-optimize, we introduce a novel approach: *direct optimization for traffic engineering (DOTE)*. Unlike predict-and-optimize, where an ML model predicts the next demand matrix (DM) for flow optimization, in DOTE an ML model *directly* learns to produce the next flow configuration, replacing both the demand prediction and LP optimization components of predict-and-optimize. As shown in Figure 3b, the DOTE controller has the same input (recent demand history) as the demand predictor, but maps it directly to a flow configuration for the next time interval. In our implementation, this mapping is performed by a DNN.

To understand the key differences between predict-and-optimize and DOTE, it helps to compare their

Figure 5. DNN training illustration.



ML components, summarized in Table 1. Both approaches are trained *offline* on sequences of empirically observed past demand matrices (e.g., five-minute demand snapshots from the past two months). However, their outputs and objectives differ. The demand predictor in predict-and-optimize outputs a demand matrix \mathbf{D} for the next interval and minimizes the distance (loss) with respect to the realized demand $\mathbf{D}^{(t)}$. In contrast, DOTE outputs a flow configuration and optimizes directly for the network operator’s flow objective, such as maximizing total flow.

DOTE avoids the pitfalls of predict-and-optimize. First, by not optimizing for a specific predicted demand matrix, DOTE generates flow configurations that provide robustly high performance across possible demand realizations. Second, by directly optimizing for the flow objective, DOTE avoids the objective mismatch in predict-and-optimize. Additionally, by relying on DNN inference instead of

solving LPs, DOTE achieves orders of magnitude reductions in online computation time.

A Closer Look at DOTE

Next we present key algorithmic insights underlying DNN training in DOTE. For further details on the specific DNN realization (e.g., architecture, size), see Perry et al.²⁷

Normalizing the DNN’s output to respect link capacities.

Our implementation of DOTE utilizes a DNN to map recent demand histories to flow configurations. The DNN outputs a nonnegative value v_p for each permissible path $p \in P$. Since these values may violate link capacities, they cannot directly serve as flow caps ω_p . We can illustrate how this is handled by revisiting the example in Figure 2. We treat v_p (see Figure 2c) as tentative flow sizes f_p , then identify the most violated link e via $\max_e \{ \frac{f_e}{\alpha(e)} \}$, where f_e is the total flow on edge e . Defining $\gamma = \max \{ \max_e \{ \frac{f_e}{\alpha(e)} \}, 1 \}$, we set $\omega_p = \frac{v_p}{\gamma}$. This normalization scales outputs to

ensure that no link exceeds capacity (as in Figure 2b).

Optimal DNN training via (stochastic, normalized) gradient ascent: Geometric intuition.

Consider the network in Figure 5a, with all link capacities 1, and each node pair (1, 2), (1, 3), and (2, 3) connected by the *single* permissible path in the figure. The realized demand matrix $\mathbf{D}^{(t)}$ for all time intervals t is identical, with demands $d_{12} = d_{13} = d_{23} = 1$ and all other demands 0. We examine how the total flow F is impacted by value assignments $v_p \geq 0$ to the permissible paths. Let $v_{(1,2)}$, $v_{(1,3)}$, and $v_{(2,3)}$ denote the values associated with paths (1, 2), (1, 3), and (2, 3), respectively. Fixing $v_{(2,3)} = 1$, we visualize the total flow F as a function of $v_{(1,2)}$ and $v_{(1,3)}$ in Figure 5b.

A geometric interpretation of concavity is that the graph of a function lies *on or above* the straight line (chord) connecting any two points on the graph. As shown in Figure 5b, the total flow F is *not* concave in the path values, as the graph of the function lies *below* the red line. We show, however, that the total flow is *quasiconcave*, meaning the function’s values along the line between any two points do not drop below the smaller of the two. This is visually evident for points A and B : All function values on the red line exceed the lowest value (point A). Quasiconcavity allows for efficient optimization via normalized gradient ascent.²⁶ In our case, starting with arbitrary values for $v_{(1,2)}$ and $v_{(1,3)}$, we can apply (normalized) gradient ascent to reach the maximum flow at point B .

We prove that the quasiconcavity of the total flow extends to *any* network topology and choice of permissible paths,²⁷ so normalized gradient ascent always converges to the maximum flow. This extends to virtually all other well-studied flow-optimization objectives, including maximum concurrent flow and congestion minimization, and to *probabilistic*, time-varying demands. Indeed, building on results in ML theory,^{8,17} we show that an idealized training process via *stochastic normalized gradient ascent*²⁹ yields optimal performance. To handle the nondifferentiability of certain objectives, our analysis relies on *sub*-gradient updates. See Perry et al.²⁷ for

a detailed exposition of the idealized training process, and for our theoretical results. These theoretical results provide a foundation for the empirical results discussed below, which were obtained for an implementation of DOTE that approximates this idealized training process.

Closed-form (sub)gradients. DNN training in DOTE leverages another key structural property: Unlike many optimization settings that rely on approximations to estimate gradients, our setting admits a closed-form expression, enabling more efficient optimization. The intuition is that once a demand matrix is realized, the induced performance for *all possible flow configurations*—not just the one actually chosen—can be computed. Revisiting the example in Figure 5a, for any realized demand matrix $\mathbf{D}^{(i)}$, the total flow F as a function of the path values is explicitly computable as: $\min\{\frac{v_{12}}{\gamma}, d_{12}\} + \min\{\frac{v_{13}}{\gamma}, d_{13}\} + \min\{\frac{v_{23}}{\gamma}, d_{23}\}$, where γ is the normalization factor applied to DNN-outputted path values. While this expression for F is not differentiable at some points (where two or more arguments of a minimum coincide), it is subdifferentiable everywhere. Thus, one can directly compute subgradients with respect to the path values without resorting to approximations. This property generalizes to arbitrary networks.²⁷

Leveraging recent advances in large-scale DNN training. DNN training in our implementation of DOTE builds on recent developments in large-scale optimization, namely, the Adam stochastic gradient descent optimizer,¹⁵ which incorporates momentum and an adaptive step size. This accommodates efficient training on extensive empirical data (many tens of thousands of traffic-demand matrix snapshots in our experiments).

Evaluation

Using network topologies and traffic demands from production networks, including Microsoft's, we sought to answer the following questions: How does DOTE compare against an *omniscient oracle* with *perfect* knowledge of future demands? And how does DOTE stack up against predict-and-optimize flow-optimization meth-

ods in backbone networks in terms of solution quality and runtime efficiency?

We evaluated DOTE against predict-and-optimize^c using empirically derived historical datasets from three production networks: GEANT,³⁰ Abilene,¹ and Microsoft's private backbone network (MBB). Traffic on MBB includes both datacenter-to-datacenter and external customer traffic. To analyze performance by traffic class, we also considered MBB_{DC}, which includes only datacenter-to-datacenter traffic. Together, these empirical datasets span network topologies of varying sizes, demand matrices with different levels of predictability, and differing amounts of available training data (from weeks to months). The above-mentioned networks were supplemented by the KDL network from Topology Zoo¹⁶ to assess runtimes. Table 2 summarizes topology sizes and traffic granularity. For each topology,

^c Results for predict-and-optimize are based on linear regression. Other predictors (ridge regression, random forest, autoregressive models, and various DNN architectures) showed similar or worse performance.²⁷

we evaluated two node-interconnection schemes: k -shortest paths ($k = 8$) and edge-disjoint paths. (See Perry et al.²⁷ for more details about our evaluation framework.)

Solution quality. Figure 6 shows results for the maximum-flow objective across all four network datasets and the two path-generation schemes, normalized by the total flow achieved by an omniscient oracle with perfect future traffic knowledge. For each network, 75% of demand matrices are used for training, and 25% for testing. Candlesticks display performance distributions: boxes represent the 25th to 75th percentiles, whiskers show the full range, and average and median values are also plotted. Strikingly, DOTE approaches oracle performance even in the lower percentiles, consistently outperforming predict-and-optimize. On Microsoft's datasets, DOTE delivers roughly 10% more traffic on average than predict-and-optimize, a significant improvement in network utilization. As expected, the gap between DOTE and predict-and-optimize is more pronounced for all Microsoft traffic (MBB) compared to the more predictable inter-daten-

Table 2. Datasets used to evaluate DOTE.

	#Nodes	#Edges	Length	Granularity
Abilene	11	14	4.5 months	5 minutes
GEANT	23	37	4 months	15 minutes
MBB	O(100)	O(100)	O(1) months	minutes
MBB_{DC}	O(10)	O(10)	O(1) months	minutes
KDL	754	895	N/A	N/A

Figure 6. Results for maximum flow.

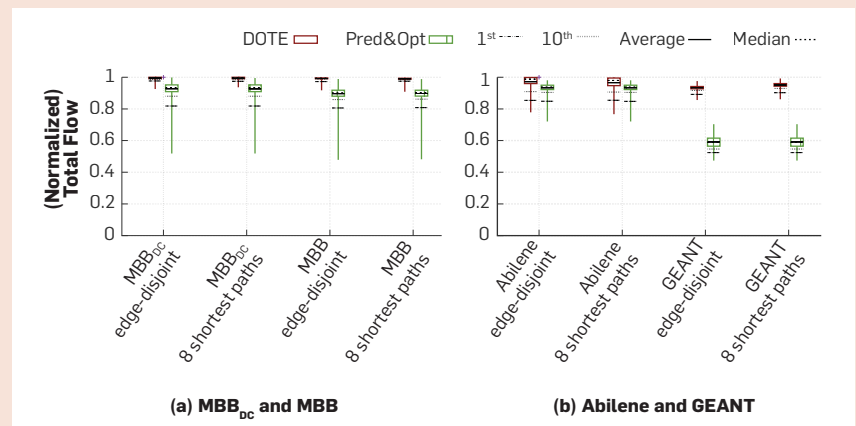


Table 3. Comparing DOTE and predict-and-optimize online runtimes for k -shortest paths.

WAN	Online Lat. (s)	
	DOTE	LP
Abilene	0.0005	0.02
MBB _{DC}	0.003	0.05
Geant	0.002	0.04
MBB	0.2	1
KDL	1.1	110

ter traffic (MBB_{DC}). DOTE maintains high network utilization even at the very lowest percentiles, while predict-and-optimize lags in tail performance across all evaluated scenarios.

Runtimes. Table 3 compares the online runtimes of the two flow-optimization approaches. Observe that DOTE's DNN inference time is significantly faster than predict-and-optimize's LP-based optimization. LP computation latency can be minutes on very large networks.^{2,25} Indeed, solving the LP for the large KDL network (see Table 2) lasts close to two minutes. In contrast, DOTE's DNN inference latency on large networks such as KDL is still within a few seconds.

We refer the reader to Perry et al.²⁷ for many more results, covering additional flow-optimization objectives (congestion minimization and maximum concurrent flow), network topologies and traffic scenarios, path-assignment schemes, and comparison baselines (oblivious routing, reinforcement learning, and hybrid approaches). Also see that paper for results on robustness to network failures, sensitivity to unexpected changes in traffic and to natural demand drift, and training-time analysis.

Conclusion

In this article, we introduced a novel algorithmic perspective on bridging a major gap between the theory and practice of flow optimization in computer networks. We laid down the theoretical foundations for our methodology, and realized our method by leveraging recent advances in large-scale optimization and deep learning. Using extensive real-world data, we established that our approach substantially surpasses the state of the art in terms of both flow solution quality and online runtimes. Our work leaves

many exciting open avenues for future research on extending and applying DOTE. Promising directions include applying machine learning to cope with network failures, learning the permissible network paths, and leveraging DOTE's fast DNN-inference-based computation for network capacity planning. **C**

References

- Abilene/Internet2. (2004); <http://www.internet2.edu/>
- Abuzaid, F. et al. Contracting wide-area network topologies to solve flow problems quickly. In *18th USENIX Symp. on Networked Systems Design and Implementation*. USENIX Association (2021), 175–200; <https://www.usenix.org/conference/nsdi21/presentation/abuzaid>
- Ahuja, S.S. et al. Capacity-efficient and uncertainty-resilient backbone network planning with hose. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conf.* ACM (2021), 547–559.
- Applegate, D. and Cohen, E. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM (2003), 313–324.
- Axioti, K., Mądry, A., and Vladu, A. Faster sparse minimum cost flow by electrical flow localization. In *2021 IEEE 62nd Annual Symp. on Foundations of Computer Science*. IEEE (2022), 528–539.
- Azar, Y. et al. Optimal oblivious routing in polynomial time. In *Proceedings of the Thirty-Fifth Annual ACM Symp. on Theory of Computing*. ACM (2003), 383–388.
- Bogle, J. et al. TEAVAR: Striking the right utilization-availability balance in WAN traffic engineering. In *Proceedings of the ACM Special Interest Group on Data Communication*. ACM (2019), 29–43.
- Hazan, E., Levy, K., and Shalev-Shwartz, S. Beyond convexity: Stochastic quasi-convex optimization. In *Advances in Neural Information Processing Systems*, Vol. 28. Curran Associates (2015); https://proceedings.neurips.cc/paper_files/paper/2015/file/934815ad542a4a7c5e8a2dfa04fea9f5-Paper.pdf
- Hong, C.-Y. et al. Achieving high utilization with software-driven WAN. In *Proceedings of the ACM SIGCOMM 2013 Conf. on SIGCOMM*. ACM (2013), 15–26.
- Hong, C.-Y. et al. B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN. In *Proceedings of the 2018 Conf. of the ACM Special Interest Group on Data Communication*. ACM (2018), 74–87.
- Jain, S. et al. B4: Experience with a globally deployed software defined WAN. In *Proceedings of the ACM SIGCOMM 2013 Conf. on SIGCOMM*. ACM (2013), 3–14.
- Kandula, S., Menache, I., Schwartz, R., and Babbula, S.R. Calendaring for wide area networks. In *Proceedings of the 2014 ACM Conf. on SIGCOMM*. ACM (2014), 515–526.
- Karakostas, G. Faster approximation schemes for fractional multicommodity flow problems. *ACM Trans. Algorithms* 4, 1 (Mar. 2008), 17.
- Kelner, J.A., Miller, G.L., and Peng, R. Faster approximate multicommodity flow using quadratically coupled flows. In *Proceedings of the Forty-Fourth*

- Annual ACM Symp. on Theory of Computing*. ACM (2012), 1–18.
- Kingma, D.P. and Ba, J. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] (2014); <https://arxiv.org/abs/1412.6980>
 - Knight, S. et al. The internet topology zoo. *IEEE J. on Selected Areas in Communications* 29, 9 (2011), 1765–1775.
 - Konnov, I.V. On convergence properties of a subgradient method. *Optimization Methods and Software* 18, 1 (2003), 53–62.
 - Krishnaswamy, U. et al. OneWAN is better than two: Unifying a split WAN architecture. In *20th USENIX Symp. on Networked Systems Design and Implementation*. USENIX Association (2023), 515–529; <https://www.usenix.org/conference/nsdi23/presentation/krishnaswamy>
 - Kumar, P. et al. Semi-oblivious traffic engineering: The road not taken. In *15th USENIX Symp. on Networked Systems Design and Implementation*. USENIX Association (2018), 157–170; <https://www.usenix.org/conference/nsdi18/presentation/kumar>
 - Lan, T., Kao, D., Chiang, M., and Sabharwal, A. An axiomatic theory of fairness in network resource allocation. In *2010 Proceedings IEEE INFOCOM (2010)*, 1–9.
 - Leighton, T. et al. Fast approximation algorithms for multicommodity flow problems. *J. Comput. System Sci.* 50, 2 (1995), 228–243.
 - Leighton, T. and Rao, S. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM* 46, 6 (Nov. 1999), 787–832.
 - Liu, H.H. et al. Traffic engineering with forward fault correction. In *Proceedings of the 2014 ACM Conf. on SIGCOMM*. ACM (2014), 527–538.
 - Mo, J. and Walrand, J. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking* 8, 5 (2000), 556–567.
 - Narayanan, D. et al. Solving large-scale granular resource allocation problems efficiently with POP. In *Proceedings of the ACM SIGOPS 28th Symp. on Operating Systems Principles*. ACM (2021), 521–537.
 - Nesterov, Y.E. Minimization methods for nonsmooth convex and quasiconvex functions. *Matekon* 29, 3 (1984), 519–531.
 - Perry, Y. et al. DOTE: Rethinking (predictive) WAN traffic engineering. In *20th USENIX Symp. on Networked Systems Design and Implementation*. USENIX Association (2023), 1557–1581; <https://www.usenix.org/conference/nsdi23/presentation/perry>
 - Racke, H. Minimizing congestion in general networks. In *The 43rd Annual IEEE Symp. on Foundations of Computer Science, 2002. Proceedings*. IEEE (2002), 43–52.
 - Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM (2021).
 - Uhlig, S., Quoitin, B., Lepropre, J., and Balon, S. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.* 36, 1 (Jan. 2006), 83–86.
 - Zhang, H. et al. Guaranteeing deadlines for inter-data center transfers. *IEEE/ACM Trans. on Networking* 25, 1 (2017), 579–595.

Yarin Perry (yarin.perry@mail.huji.ac.il) is a Ph.D. student at the Hebrew University of Jerusalem, Jerusalem, Israel.

Srikanth Kandula is a senior principal engineer at Amazon Web Services, Seattle, WA, USA. This work was done while he was at Microsoft.

Ishai Menache is a partner research manager at Microsoft Research, Redmond, WA, USA.

Michael Schapira is a professor of computer science at the Hebrew University of Jerusalem, Jerusalem, Israel.

Aviv Tamar is an associate professor of computer science at the Technion - Israel Institute of Technology, Haifa, Israel.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

research highlights

P. 88

**Technical
Perspective
Fusing LLMs with
Completion Engines
for Automated
Program Repair**

By Gabriele Bavota

P. 89

Copiloting the Copilots for Automated Program Repair

By Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang

Technical Perspective

Fusing LLMs with Completion Engines for Automated Program Repair

By Gabriele Bavota

THE LAST DECADE has seen the rise of artificial intelligence (AI) in software engineering (SE), with the (partial) automation of complex tasks such as code writing, summarization, and repair. The successful application of AI in SE is mostly due to three factors: first, the unprecedented amount of software-related data that can be found in open source projects hosted on platforms such as GitHub which, at the time of writing, counts more than 500 million public repositories; second, the advances in deep learning (DL), such as the introduction of novel architectures like transformers; and third, the availability of powerful hardware at a reasonable cost, which made DL more practical.

Let's discuss examples of how a DL model can automate an SE task, and in particular program repair, the task subject of the accompanying paper. Automated program repair aims to automatically fix a given buggy code. Training data for this task can be collected from open source projects in the form of bug-fixing commits, namely code changes in which developers fix bugs. This gives access to the *buggy* and to the *fixed* version of the code (before and after the commit, respectively), which can be used to teach the DL model: (i) coding patterns characterizing buggy code and (ii) code transformations (patches) typical of bug-fixing activities. Also, recent large language models (LLMs) can learn what a *natural* coding pattern is by ingesting massive amounts of source code, thus suggesting the rewriting of *unnatural* statements, possibly representing buggy implementations. With this approach, LLMs do not even require the availability of labeled buggy and clean code samples to be applied for automated program repair.


While LLM-based program repair has achieved remarkable results, LLMs applied to code face a fundamental challenge: They process code

as a sequence of tokens rather than as structured, semantically constrained entities. This means many LLM-generated patches fail to meet even basic compilation requirements. Addressing this limitation is crucial for making AI-assisted software repair useful in practice. This is the issue tackled by Wei et al. in the accompanying paper. The basic idea proposed by the authors is to approach automated program repair by combining a LLM with a code completion engine. The latter is a tool that can parse incomplete programs and suggest semantically valid completions for it, mostly relying on syntax rules and static analysis.

The proposed approach, named "Repilot," takes as input a program affected by a single bug. Such a bug must be fixable by modifying (rewriting) a contiguous subset of the statements featured in the code, identified via fault localization. Repilot asks an LLM to rewrite the buggy part of the code token by token: Every time a token must be generated, the LLM outputs a list of the most likely tokens, with associated probabilities. The completion engine is then queried to modify the probability list by dynamically zeroing the probabilities of invalid tokens. The token to generate is selected as the one having the highest probability in the updated list. The completion engine is also used for better predicting complex identifiers needed in the patch (e.g., long identifiers possibly composed by

uncommon words), which represent another challenging scenario for LLM-based program repair.

By showing that Repilot significantly outperforms the state-of-the-art baselines, fixing more bugs and generating more compilable patches, this work makes a strong case for neuro-symbolic AI as a key direction in software engineering research. Neuro-symbolic AI combines the learning capabilities of neural networks with the structured reasoning of symbolic systems, thus bridging the gap between statistical learning and rule-based constraints, creating models that are flexible and logically sound. All tasks dealing with source code are likely to benefit from a neuro-symbolic approach for their automation. For example, code generation and completion could be constrained by the symbolic component, limiting the production of invalid code. In the case of automated refactoring, the neural part could come up with revised and improved code designs, while the symbolic part could discard solutions not preserving the code behavior. Even for automated testing, a symbolic component could provide feedback to the LLM about uncovered code branches, thus improving the generated test suite.

For anyone interested in the intersection of AI and software engineering, the accompanying paper presents an important step forward in making AI-driven program repair more practical, reliable, and efficient. Also, it offers valuable insights into the power of neuro-symbolic AI and lays the groundwork for future innovations in AI-assisted code-related tasks. 

This work makes a strong case for neuro-symbolic AI as a key direction in software engineering research.

Gabriele Bavota (gabriele.bavota@usi.ch) is a full professor at the Faculty of Informatics of the Università della Svizzera Italiana (USI), Lugano, Switzerland, where he is part of the Software Institute and leads the SEART research group.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

Copiloting the Copilots for Automated Program Repair

By Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang

ABSTRACT

During automated program repair (APR), it can be challenging to synthesize correct patches for real-world systems in general-purpose programming languages. Recent large language models (LLMs) have been shown to be helpful “copilots” in assisting developers with various coding tasks, and have also been directly applied for patch synthesis. However, most LLMs treat programs as sequences of tokens, meaning they are ignorant of the underlying semantics constraints of the target programming language. This results in plenty of statically invalid generated patches, impeding the practicality of the technique. Therefore, we propose Repilot, a general code-generation framework to further copilot the copilots (i.e., LLMs) by synthesizing more *valid* patches during the repair process. Our key insight is that many LLMs produce outputs autoregressively (i.e., token by token), resembling human writing programs, which can be significantly boosted and guided through a completion engine. Repilot synergistically synthesizes a candidate patch through the interaction between an LLM and a completion engine, which prunes away infeasible tokens suggested by the LLM and proactively completes the token based on the suggestions provided by the completion engine. Our evaluation on a subset of the widely used *Defects4j 1.2* and *2.0* datasets shows that Repilot outperforms state-of-the-art techniques by fixing 27% and 47% more bugs, respectively. Moreover, Repilot produces more valid and correct patches than the base LLM with the same budget. While we focus on leveraging Repilot for APR in this work, the overall approach is also generalizable to other code-generation tasks.

1. INTRODUCTION

Automated program repair (APR) seeks to reduce the manual bug-fixing effort of developers by automatically synthesizing patches given the original buggy code.⁹ Recent advancements in large language models (LLMs) empower researchers to directly leverage their power for APR.^{11,19,21} LLMs not only achieve impressive performance on many natural language processing (NLP) tasks,³ but they are also shown to be reliable *copilots* in assisting developers with various coding tasks.^{2,14} The reason is that modern LLMs often include large amounts of available open source code repositories as part of their training dataset. Recognizing the power of LLMs, researchers have recently applied LLMs

for APR: Instead of translating buggy code into correct code, LLMs are directly used to synthesize the correct patch from the surrounding context. AlphaRepair¹⁹ reformulates the APR problem as a *cloze* (or infilling) task.^{4,8} It first replaces buggy code snippets with masked tokens and then uses CodeBERT⁶ to fill in correct code given the surrounding context. Other studies on LLMs for APR have applied even larger LLMs with different repair settings (including generating complete patch functions).^{13,21}

While prior LLMs for APR techniques achieve state-of-the-art bug-fixing performance, they use LLMs in a black-box manner, where the underlying LLMs generate programs according to the token distribution without any structural or semantic understanding of the code. To highlight the limitations with current LLMs for APR tools, in Figure 1 we show three scenarios where LLMs can generate incorrect patches:

1. Generating infeasible tokens. In Figure 1.1, the LLM has a high probability (>90%) of generating `String` to complete the `asString` method. However `asString` is not a valid field access for the object `t` and is also not part of the scope of the current buggy method. In this case, the patches generated using `asString` will never be correct, as it cannot compile. By directly using the model probabilities, LLMs are likely to generate many patches using invalid tokens and decrease the likelihood of generating the correct patch with `End` (0.2%).

2. Hard to generate rare tokens. LLMs usually cannot generate a complete identifier name in one step, since it uses subword tokenization¹⁷ to break uncommon words into smaller subwords. These uncommon words manifest as rare identifiers in code, where identifier names are CamelCase or underscore combinations of multiple words (e.g., `asEndTag` in Figure 1.2). As such, LLMs need to generate these identifiers step by step, needing not only multiple iterations but also accurate output in each step. Since prior approaches^{13,21} sample based on probability, the likelihood of completing a rare token to fix a bug can be extremely low.

3. No explicit consideration of types. In addition to potentially generating out-of-scope identifiers, LLMs do not have access to various type information that gives hints to the valid identifiers. The return type of `asEndTag()`, in Figure 1.3, is `EndTag`, whose definition is not explicitly given to the LLM in its immediate context. As such, LLMs do not know the correct member fields of `EndTag` and may generate invalid patches containing identifiers that do not fit the required type. On the contrary, a completion engine has full access to the project and can easily figure out the return type of `asEndTag()` through static analysis on the abstract syntax tree of the program. By treating code as a sequence

This paper was originally published in *Proceedings of the 31st ACM Joint Meeting European Software Engineering Conf. and Symp. on the Foundations of Software Engineering* (2023).

Figure 1. Limitations of existing LLM-based APR approaches.



of textual tokens, the important type information is not encoded.

To address the aforementioned limitations, we propose Repilot, a framework to further copilot the AI copilots (i.e., LLMs) via fusing LLMs with completion engines to synthesize more *valid* patches. Completion engines¹⁵ can parse incomplete programs and reason about the semantics in an error-tolerant manner. Our key insight is *to liken LLM autoregressive token generation as a human developer code writing, where the Completion Engine can provide real-time updates to check if the human/LLMs written partial code is valid*. Repilot first uses the LLM to provide the probabilities of generating the next token in the patch and then queries the completion engine to modify the probability list by dynamically zeroing the probabilities of invalid tokens. We can then sample from the new probability list to select the next token. Furthermore, recognizing the ability for completion engines to suggest completions, we use this feature whenever there is only one possible identifier suffix to complete the context. This not only allows Repilot to generate patches with valid rare and long identifiers but also reduces the work of LLMs needed to iteratively generate long identifier names.

To demonstrate the generalizability of Repilot, we instantiate Repilot with two LLMs having distinct architectures and sizes: CodeT5-large,¹⁸ an encoder-decoder LLM with 770 million parameters, and INCODER-6.7B,⁸ a decoder-only LLM with 6.7 billion parameters, both capable of code infilling from prefix and suffix context. We further implement a Java completion engine for Repilot based on the Eclipse JDT Language Server,^{7a} since it provides various semantics-based analyses through a consistent language server protocol.¹⁵ We evaluate Repilot on a subset of the widely studied Defects4J 1.2 and 2.0 datasets¹² and demonstrate state-of-the-art results in both the number of correct fixes and compilation rate—the percentage of the generated patches that can be successfully compiled. Furthermore, while we evaluated Repilot for APR in this work, we believe the overall framework can be easily applied to other code-generation tasks, including code completion,^{5,23} program synthesis,^{14,16} and test generation.^{4,20} In summary, we make the following contributions:

► **Direction.** We open a new direction for fusing LLMs with completion engines for more powerful APR and beyond. Compared to prior techniques, which either perform post-process-

a Eclipse JDT LS (2023); <https://projects.eclipse.org/projects/eclipse.jdt.ls>.

ing to fix invalid generations or use simple static methods to approximate these valid tokens, our approach leverages a powerful completion engine to directly provide accurate feedback on partial programs to avoid invalid token generations.

► **Technique.** We implement Repilot, an LLM for APR approach instantiated with the CodeT5 and INCODER models to perform cloze-style repair combined with our modified Eclipse JDT Language Server⁷ as the completion engine. In Repilot, we use the completion engine to systematically prune invalid tokens generated by LLMs and to directly complete code given the current prefix. Furthermore, we implement optimizations to significantly reduce the overhead of Repilot. We have open sourced our tool on GitHub.^b

► **Study.** We compare Repilot against state-of-the-art APR tools on Defects4J 1.2 and 2.0. Repilot is able to achieve new state-of-the-art results of 66 Defects4J 1.2 single-hunk bugs and 50 Defects4J 2.0 single-line bugs fixed respectively, with 30 more combined fixes across both datasets compared to the previous best baseline. Our further evaluation shows that Repilot consistently improves the validity and correctness of the generated patches with a limited overhead (7% for CodeT5 and negligible for INCODER).

2. Approach

Following most recent deep learning-based APR tools,^{19,22,24} Repilot focuses on fixing single-hunk bugs, where the patch is obtained by changing a continuous section of code under perfect fault localization. Repilot can be extended for multi-hunk bugs by replacing all hunk locations at the same time with separate infilling tokens and using LLMs to generate the replacement hunks. Benefiting from the era of LLMs, as shown in Figure 2, in this paper, we treat the repair problem as a *cloze* task,¹⁹ where a patch is formed by first replacing the buggy hunk with a masked span token () and then using the LLM to direct-synthesize the fixed hunk from the surrounding code context to replace the span token.

b <https://github.com/ise-uiuc/Repilot>

Figure 2. Cloze-style program repair.

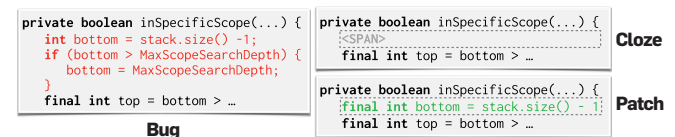
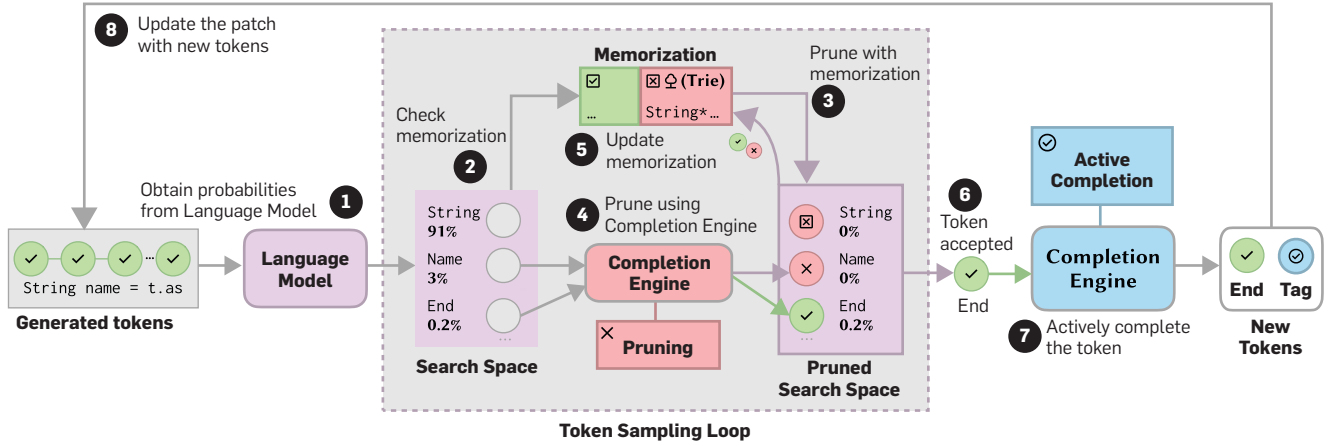


Figure 3. Overview of Replit.



2.1. Overview.

Figure 3 shows an overview of how Replit synthesizes a program that acts as the repaired hunk of the original buggy program. The generation loop consists of a loop that keeps updating the generation with tokens newly generated from the synergy between the language model and completion engine. The loop starts by applying the current generation as the input to the language model (1), which returns a search space of a mapping from a suggested next token to its probability. Replit then enters a token-selection phase that repeatedly samples a token from the search space, checking its feasibility and pruning the search space until a token is accepted. Every time a token is sampled, Replit

Algorithm. Main repair loop of Replit.

Inputs: Large language model LM , Completion engine CE , Buggy program $prog$, and range of buggy hunk $range$.

Output: Patch for the buggy program.

```

1: func Repair( $LM$ ,  $CE$ ,  $prog$ :  $\Sigma_{PL}^*$ ,  $range$ :  $\mathbb{N} \times \mathbb{N}$ )  $\rightarrow$   $\Sigma_{PL}^*$ :
2:    $encoder-inputs$ :  $\Sigma_{LM}^*$  := BUILDINPUTS( $prog$ ,  $range$ )
3:    $encoded-rep$ : EncRep :=  $LM.encode(encoder-inputs)$ 
4:    $decoder$ :  $\Sigma_{LM} \rightarrow DecRep := LM.decode(encoded-rep)$ 
5:    $hunk$ :  $\Sigma_{LM}^* := \epsilon$ 
6:   while true do
7:      $\triangleleft$  Form  $patch$  by replacing buggy hunk with  $hunk$   $\triangleright$ 
8:      $patch$  :=  $prog[range \leftarrow Str(hunk)]$ 
9:      $\triangleleft$  Move  $caret$  after the current generation  $\triangleright$ 
10:     $caret$  :=  $range.start + |Str(hunk)|$ 
11:     $tokens$ :  $\Sigma_{LM} \rightarrow [0, 1]$  :=  $decoder(hunk)$ 
12:     $next-token$ :  $\Sigma_{LM} := GUIDEDPRUNE(CE, patch, caret, tokens)$ 
13:    if  $next-token = end-token$  then
14:      return  $patch$ 
15:     $completion-toks$ :  $\Sigma_{LM}^* := ACTIVELYCOMPLETE(CE, patch, caret)$ 
16:     $hunk$  :=  $hunk \cdot next-token \cdot completion-toks$ 

```

first checks if it hits the memorization (2), which stores the tokens known to be feasible or infeasible. The memorization of infeasible tokens includes the use of a prefix tree data structure (Trie). When the token hits the memorization and is infeasible, the search space is pruned by setting this token's probability to zero (3), and the next sampling will run on the updated search space. In this way, the same token is not sampled again during the token-selection phase. If the token misses the memorization, the search space is pruned under the guidance of the completion engine (4). Provided the sampled token is rejected by the completion engine, Replit zeroes out its probability. Otherwise, it is accepted and this token-selection process terminates. The memorization gets updated in both cases (5). After a token is accepted (6), we further leverage the completion engine, trying to actively complete the token (7). The active completion may either produce more tokens or add nothing to the accepted token. Finally, Replit appends all the newly generated tokens to the current generation and begins a new loop until a complete patch is generated. The loop stops when the model generates the special token *end-token*.

The algorithm details this process and shows how a complete patch program is generated. It additionally describes how Replit performs the pre-processing (Lines 2–5) and formalizes completion-guided pruning procedures illustrated in Figure 3 using two functions **GuidedPrune** and **ActivelyComplete** (Lines 6–16). In all our algorithms, we use a "dot-notation" to specify an entity of a tuple (e.g., $LM.encode$) but use an abbreviation form when the context is clear (e.g., Σ_{LM} and Σ_{PL} for $LM.\Sigma_{LM}$ and $CE.\Sigma_{PL}$). We also optionally apply *type annotations* for clarification. Note that we simplify the definition of the completion engine by restricting it to be called with one program. In practice, a completion engine is always initialized with the entire project and can provide suggestions based on global information.

3. EVALUATION

3.1. Comparison with existing tools.

We follow the prior approach for cloze-style APR¹⁹ to make use of repair templates for a faithful evaluation. Instead of

Table 1. Number of correct fixes on Defects4J 1.2 single-hunk and Defects4J 2.0 single-line bugs.

Tool	Methodology	#Correct Fixes		
		Defects4J 1.2	Defects4J 2.0	Total
CoCoNuT	NMT	30	-	-
DLFix	NMT	32	-	-
PraPR	Template	35	-	-
TBar	Template	41	7	48
CURE	NMT	43	-	-
RewardRepair	NMT	45	24	69
Recoder	NMT	51	10	61
AlphaRepair	LLM	52	34	86
Repilot	LLM	66	50	116

replacing the entire buggy line with model-generated code, these templates systematically keep parts of the buggy line to reduce the amount of code the LLM needs to generate.

Defects4J 1.2. We first compare Repilot against the state-of-the-art APR tools on single-hunk bugs from Defects4J 1.2. Table 1 shows the number of correct patches produced by Repilot, evaluated in cloze-style, along with the baselines. *Repilot achieves the new state-of-the-art result of 66 correct bug fixes on Defects4J 1.2, outperforming all previous APR tools.* Figure 4a shows the Venn diagram of the unique bugs fixed for the top-performing LLM- and NMT-based APR tools, where Repilot is able to obtain the highest number of eight unique bugs. Furthermore, Figure 4b compares the unique bugs fixed for all top-performing baselines and with all other APR tools combined (Others). We observe that Repilot is able to fix seven bugs that no other baselines have been able to fix so far.

To demonstrate the ability of Repilot to fix difficult bugs, Figure 5 shows a unique bug (Closure-133) from Defects4J 1.2 only Repilot can fix. Adding the new assignment statement using the global variable `NO_UNREAD_TOKEN`—which is difficult to generate, as it does not appear within the surrounding context of the bug location—fixes this bug. Repilot first uses CodeT5 to generate the initial prefix of `unread`. Then, using the completion engine, Repilot recognizes that `Token` is the only semantically correct continuation and directly performs active completion to return `unreadToken`. Similarly for generating `NO_UNREAD_TOKEN`, Repilot first generates `NO_` and then uses active completion to directly generate this rare identifier without having to repeatedly sample the LLM. It is difficult for prior LLM- and NMT-based APR tools to generate this fix, as LLMs or NMT models may not be able to complete this rare identifier since it requires multiple continuous steps to generate. In contrast, Repilot, through the use of active completion, can directly generate this rare identifier given only the initial identifier prefix to quickly arrive at this correct patch.

Defects4J 2.0. We further evaluate Repilot against baselines evaluated on the single-line bugs in Defects4J 2.0. For these bugs, we follow prior approach for cloze-style APR¹⁹ to make use of repair templates. Instead of replacing the entire buggy line with model-generated code, these templates systematically keep parts of the buggy line (e.g., prefix or suffix, method parameters and calls) to reduce the amount of code the LLM needs to generate. We apply these repair templates for Defects4J 2.0 single-line bugs only since they are designed for single-line bugs. Table 1 also shows the number of correct fixes on Defects4J 2.0 compared with the baselines. We observe that *Repilot is able to fix the highest number of bugs 50 (16 more than the next best baseline) on Defects4J 2.0.* This improvement over existing baselines shows that

Figure 4. Correct fix Venn diagrams on Defects4J 1.2.

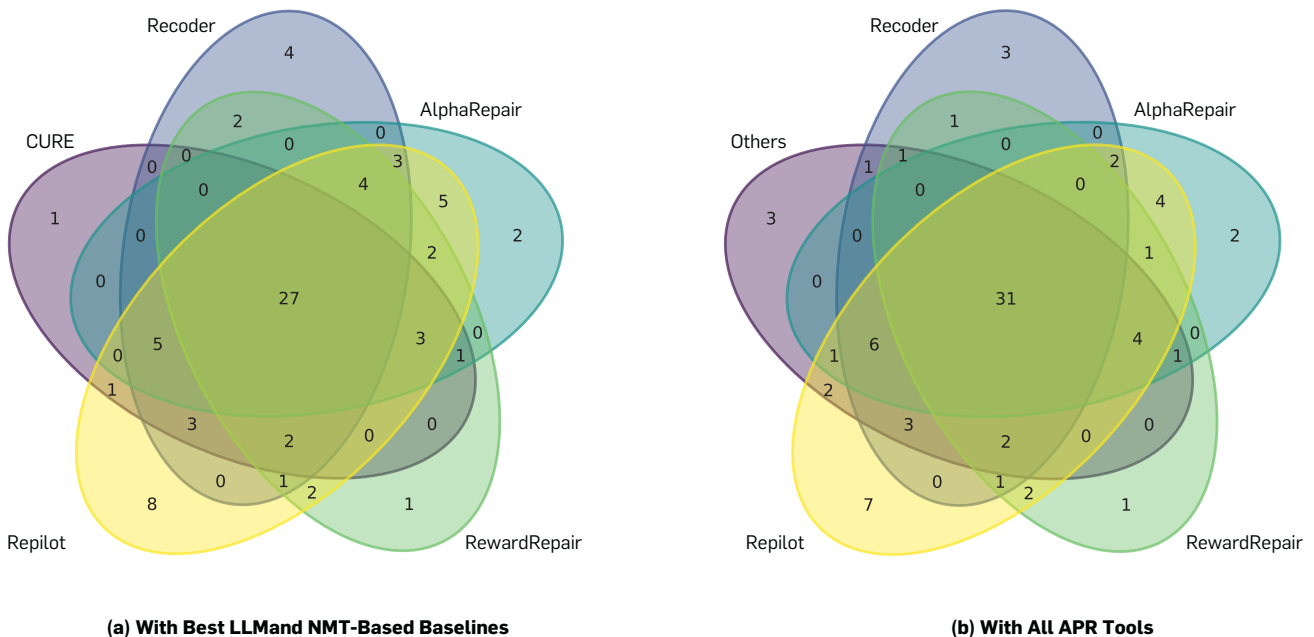
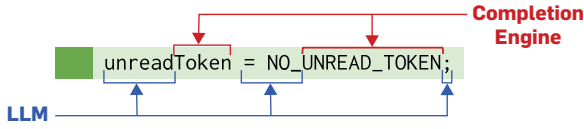


Figure 5. Unique bug fix by Repilot on Defects4J 1.2.

```
private String getRemainingJSDocLine() {
    String result = stream.getRemainingJSDocLine();
+   unreadToken = NO_UNREAD_TOKEN;
    return result;
}
```

Bug-ID: Closure-133



Patch Generation Process

Repilot can generalize to two versions of Defects4J datasets and demonstrates the power of repair templates to boost the performance of LLM-based APR tools.

Figure 6 shows a unique bug from Defects4J 2.0 only Repilot can fix. First, Repilot generates the patch up to the caret position. The completion engine then captures the exact type of the object from `Token.EndTag` to `String`. Using this information, Repilot correctly prunes tokens that are not a part of the `String` class (e.g., `name` and `text`). Hence, the generated patch contains a valid `String` class method of `toLowerCase()`, which correctly fixes this bug. Similar to the previous unique bug fix in Defects4J 1.2, prior LLM-based APR tools may waste a lot of time generating semantically incorrect continuations as they do not have access to the type information. Furthermore, NMT-based APR tools, such as CURE¹⁰ over-approximate the list of valid identifiers by statically grabbing all the accessible fields and may not generate this fix since a pruned identifier (e.g., `name`) can also be valid for a different object type. Repilot uses the completion engine to analyze partial programs and realize complex type propagation for effective pruning.

3.2. Compilation rate analysis.

We evaluate the compilation rate of the patches generated by Repilot compared with prior learning-based APR techniques. Table 2 shows the percentage of compilable patch-

Table 2. Comparison with existing APR tools on compilation rate on Defects4J 1.2. “-” denotes data not available.

Tool	% Compilable Patches			
	Top-30	Top-100	Top-1000	Top-5000
SequenceR	33%	-	-	-
CoCoNuT	24%	15%	6%	3%
CURE	39%	28%	14%	9%
AlphaRepair	25%	22%	16%	13%
RewardRepair	45%	38%	33% ^c	-
Repilot	66%	62%	58%	59%

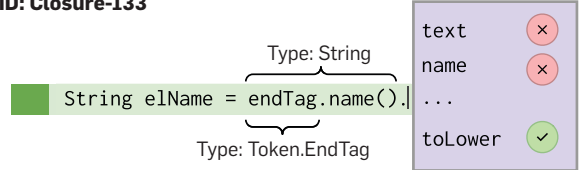
c This is the top 200 rate for RewardRepair as it does not include top 1,000.

Figure 6. Unique bug fix by Repilot on Defects4J 2.0.

```
private void popStackToClose(Token.EndTag endTag) {
-   String elName = endTag.name();
+   String elName = endTag.name().toLowerCase();

    Element firstFound = null;
}
```

Bug-ID: Closure-133



Patch Generation Process


es on the Defects4J 1.2 dataset. We observe that across all numbers of patches generated, Repilot significantly improves the percentage of compilable patches compared with prior tools. We first notice that LLM-based APR tools (Repilot and AlphaRepair), are able to sustain their compilation rate compared with NMT-based tools (CoCoNuT and CURE), where the compilation rate drastically decreases as we increase the number of patches. This shows the ability for LLMs to generate large amounts of reasonable patches. Repilot is able to sustain a near 60% compilation percentage at 1,000 patches generated while the prior approach is barely above 30%.

Compared with CURE¹⁰, where an overestimation of valid identifiers is obtained via static analysis and used to prune invalid tokens generated by NMT model, Repilot leverages the powerful completion engine to track the current context to obtain a more accurate pruning step. Furthermore, compared with RewardRepair²² where the compilation rate is boosted through penalizing uncompileable patches during training, Repilot directly uses a LLM combined with a completion engine to avoid this high cost of training a new model. Additionally, Repilot uses the active completion ability of the completion engine to directly generate these rare identifiers to further boost the compilation rate. As such, Repilot achieves the highest percentage of compilable patches across all four different settings.

4. CONCLUSION

We propose Repilot—the first APR approach to combining the direct usage of LLMs (e.g., CodeT5 and INCODER) with on-the-fly guidance provided by completion engines. During autoregressive token generation, Repilot queries the completion engine not only to *prune* invalid tokens but also to *proactively complete* the currently generated partial program, thereby reducing the search space of the LLM. Our evaluation on a subset of the widely studied Defects4J 1.2 and 2.0 datasets shows Repilot achieves state-of-the-art results. Furthermore, Repilot, through the usage of completion engines, can generate more valid and compilable patches than prior tools with minimal overhead compared with directly using LLMs for APR.

ACKNOWLEDGMENT

We thank Yifeng Ding for his helpful discussion on this work. This work was partially supported by NSF grants CCF-2131943 and CCF-2141474, as well as Kwai Inc. 

References


1. Aghajanyan, A. et al. CM3: A causal masked multimodal model of the internet. *CoRR, abs/2201.07520* (2022); <https://tinyurl.com/2da3ww27>.
2. Austin, J. et al. Program synthesis with large language models. *CoRR, abs/2108.07732* (2021); <https://tinyurl.com/yj2zpk86>.
3. Brown, T.B. et al. Language models are few-shot learners. *CoRR, abs/2005.14165* (2020); <https://tinyurl.com/yaehcygq>.
4. Deng, Y. et al. Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models. In *Proc. of the 32nd ACM SIGSOFT Intern. Symp. on Software Testing and Analysis*, ACM (2023), 423–435; <https://doi.org/10.1145/3597926.3598067>.
5. Ding, Y. et al. Crosscodeeval: A diverse and multilingual benchmark for cross-file code completion. *arXiv* (2023).
6. Feng, Z. et al. CodeBERT: A pre-trained model for programming and natural languages. *arXiv, volume abs/2002.08155* (2020); <https://tinyurl.com/2ylaayox>.
7. Eclipse Foundation and Wei, Y. UniverseFly/eclipse.jdt.ls: Modified Eclipse JDT LS 1.0.3 (Aug. 2023); <https://doi.org/10.5281/zenodo.8278193>.
8. Fried, D. et al. InCoder: A generative model for code infilling and synthesis. In *Proc. of the 11th Intern. Conf. on Learning Representations* (2023); <https://tinyurl.com/24d2un5m>.
9. Gazzola, L., Micucci, D., and Mariani, L. Automatic software repair: A survey. *IEEE Transactions on Software Engineering* 45, 1 (2019), 34–67; DOI: 10.1109/TSE.2017.2755013.
10. Jiang, N., Lutellier, T., and Tan, L. Cure: Code-aware neural machine translation for automatic program repair. In *Proc. of the 43rd Intern. Conf. on Software Engineering*, IEEE Press (2021), 1161–1173; DOI: 10.1109/ICSE43902.2021.00107.
11. Joshi, H. et al. Repair is nearly generation: Multilingual program repair with LLMs. AAAI (Feb. 2023); <https://tinyurl.com/26kd26vj>.
12. Just, R., Jalali, D., and Ernst, M.D. Defects4j: A database of existing faults to enable controlled testing studies for java programs. In *Proc. of the 2014 Intern. Symp. on Software Testing and Analysis*. Association for Computing Machinery (2014), 437–440; DOI: 10.1145/2610384.2628055.
13. Kolak, S.D., Martins, R., Le Goues, C., and Helleendoorn, V.J. Patch generation with language models: Feasibility and scaling behavior. *Deep Learning for Code Workshop* (2022); <https://tinyurl.com/2bxb2grx>.
14. Liu, J., Xia, C.S., Wang, Y., and Zhang, L. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. *arXiv* (2023).

15. Microsoft. Language server protocol (2023); <https://tinyurl.com/ygln8hc>.
16. Poesia, G. et al. SynchroMesh: Reliable code generation from pre-trained language models. In *Proc. of the Intern. Conf. on Learning Representations* (2022); <https://tinyurl.com/28kxt3ta>.
17. Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Assoc. for Computational Linguistics (Vol. 1: Long Papers)*. Association for Computational Linguistics, (Aug. 2016), 1715–1725; <https://tinyurl.com/24sbfm5e>.
18. Wang, Y., Wang, W., Joty, S., and Hoi, S.C. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conf. on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics (Nov. 2021), 8696–8708; <https://tinyurl.com/2cyoc9bu>.
19. Xia, C.S. and Zhang, L. Less training, more repairing please: Revisiting automated program repair via zero-shot learning. In *Proceedings of the 30th ACM Joint European Software Engineering Conf. and Symp. on the Foundations of Software Engineering*. Association for Computing Machinery (2022), 959–971; 10.1145/3540250.3549101.
20. Xia, C.S. et al. Fuzz4All: Universal fuzzing via large language models. *arXiv* (2023).
21. Xia, C.S., Wei, Y., and Zhang, L. Automated program repair in the era of large pre-trained language models. In *Proceedings of the 45th Intern. Conf. on Software Engineering*. IEEE Press (2023), 1482–1494; DOI: 10.1109/ICSE48619.2023.00129.
22. Ye, H., Martinez, M., and Monperrus, M. Neural program repair with execution-based backpropagation. In *Proceedings of the 44th Intern. Conf. on Software Engineering*, Association for Computing Machinery (2022), 1506–1518; DOI: 10.1145/3510003.3510222.
23. Zhang, F. et al. Repocoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conf. on Empirical Methods in Natural Language Processing*. Assoc. for Computational Linguistics (2023), 2471–2484.
24. Zhu, Q. et al. A syntax-guided edit decoder for neural program repair. In *Proceedings of ESEC/FSE 2021*. Association for Computing Machinery (2021), 341–353; DOI: 10.1145/3468264.3468544.

Yuxiang Wei, University of Illinois Urbana-Champaign, Champaign, IL, USA.

Lingming Zhang, University of Illinois Urbana-Champaign, Champaign, IL, USA.

Chunqiu Steven Xia, University of Illinois Urbana-Champaign, Champaign, IL, USA.

 This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

Functional Data Structures and Algorithms

A Proof Assistant Approach

Tobias Nipkow, Editor

ISBN: 979-8-4007-3159-4

DOI: 10.1145/3731369

<http://books.acm.org>



ACM BOOKS
Collection III



Calculated Imagery

A History of Computer Graphics in Hollywood Cinema

Mark J. P. Wolf

ASSOCIATION FOR COMPUTING MACHINERY

Pick, Click, Flick!

The Story of Interaction Techniques

Brad A. Myers

ASSOCIATION FOR COMPUTING MACHINERY

From Algorithms to Thinking Machines

The New Digital Power

Domenico Talia

ASSOCIATION FOR COMPUTING MACHINERY

Turing's Children

How His Ideas Have Shaped The Modern World

Devdatt Dubhashi
Alessandro Panconesi
Gerardo Schneider

ASSOCIATION FOR COMPUTING MACHINERY

Advanced Topics and Techniques

Omar Alonso
Ricardo Baeza-Yates
(Editors)

ASSOCIATION FOR COMPUTING MACHINERY

Multi-LLM Agent Collaborative Intelligence

The Path to Artificial General Intelligence

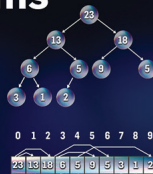


Edward Y. Chang

ASSOCIATION FOR COMPUTING MACHINERY

Thinking About Programs

Gavin Lowe



ASSOCIATION FOR COMPUTING MACHINERY

Digital Dreams Have Become Nightmares

What We Must Do

Second Edition

Ronald M. Baecker
with Jonathan Grudin

ASSOCIATION FOR COMPUTING MACHINERY

The Seymour Cray Era of Supercomputers

From Fast Machines to Fast Codes

Boelie Elzen

In-Depth. Innovative. Insightful.

Inspired by the need for high-quality computer science publishing at the graduate, faculty, and professional levels, ACM Books are affordable, current, and comprehensive in scope.

Collections I & II complete.

Collection III now publishing!



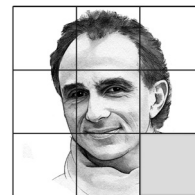
ACM BOOKS

For more information, please go to:
<http://books.acm.org>

1601 Broadway, 10th Floor
New York, NY 10019, USA
212-626-0658
acmbooks-info@acm.org



Biomedical



Dennis Shasha

DOI:10.1145/3797279

Upstart Puzzles

TimesSquare

Squaring up the grid.

A **TIMESQUARE** IS a square grid of single digits between 1 and 9 inclusive such that the sum of the products of the row values less the sum of the products of the column values is equal to a value called RowCol. Your job is to fill in a TimesSquare given part of the grid including its diagonal values, the sum of all its digit values, and the RowCol. Let's start off simple.

Warm-Up: Suppose the RowCol of a 2 by 2 grid is 8, the diagonal digits are [4, 2], and the sum of the digit values is 18. What is the 2 by 2 TimesSquare?

Solution to Warm-Up: Let's start by saying what we know. The diagonal has 4, 2, so we have

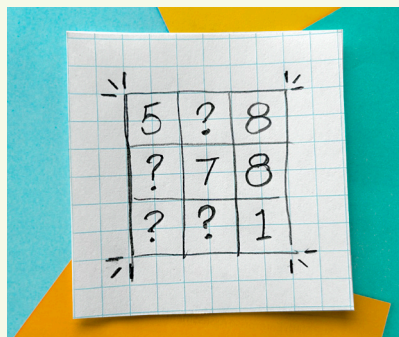
$$\begin{matrix} (4x \\ y2) \end{matrix}$$

Because the sum of the digit values is 18, $x + y = 12$. Now, because the grid squares contain single digits between 1 and 9, the possibilities for x and y are:

- xy
- 39
- 48
- 57
- 66
- 75
- 84
- 93

One approach is to try all these possibilities, but when the problem size grows, we will have to think more cleverly. Let's try for this one. For example, let's try to compute the RowCol symbolically: $(4x + 2y) - (4y + 2x) = 2x - 2y$. This result will be positive only when $x > y$, so we can throw out many possibilities just by reasoning. I invite you to try. In this case the unique TimesSquare is:

Figure. Can you replace the question marks in this matrix with digits from 1 through 9 such that the RowCol (sum of product of rows less the sum of products of columns) is 9, and the sum of the digit values is 47?



$$\begin{matrix} (48 \\ 42) \end{matrix}$$

Challenge: Suppose the RowCol of a 3 by 3 grid is 9, and the sum of the digit values is 47. What is the TimesSquare?

$$\begin{matrix} (5?8 \\ ?78 \\ ??1) \end{matrix}$$

Solution:

$$\begin{matrix} (528 \\ 278 \\ 771) \end{matrix}$$

Why does this work? The sum of 5, 2, 8, 2, 7, 8, 7, 7, 1 is indeed 47. The product of the first row digits is $5 \times 2 \times 8 = 80$, second row is 112, the third 49. The sum of all those is $80 + 112 + 49 = 241$. The products of the columns are 70, 98, and 64, respectively, with a total of 232. Finally, $241 - 232 = 9$.

Challenge: Here is a much more difficult instance. Try to fill in the 4 by 4 TimesSquare example here with a sum

of digits of 64 and a RowCol of 132:

$$\begin{matrix} (4?66 \\ ?3?1 \\ 1?? \\ ?447) \end{matrix}$$

Solution: Here is the TimesSquare:

$$\begin{matrix} (4366 \\ 4341 \\ 1273 \\ 5447) \end{matrix}$$

Challenge: Try to fill in the 5 by 5 TimesSquare below with a sum of digits of 110 and a RowCol of 1744.

$$\begin{matrix} (2685? \\ 2?4?2 \\ ?448? \\ ???12 \\ 2?2?8) \end{matrix}$$

Solution: Here is the TimesSquare:

$$\begin{matrix} (26854 \\ 25462 \\ 64486 \\ 24812 \\ 24258) \end{matrix}$$

Upstart: Given a TimesSquare of size $N \times N$, the RowCol and the sum of digits and the diagonals, can you design a sub-exponential algorithm to fill in at least one solution. ■

Dennis Shasha (shasha@cs.nyu.edu) is a professor of computer science in the Department of Computer Science at the Courant Institute at New York University, New York, NY, USA.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

NEW BOOK RELEASE



ACM BOOKS

Collection III



Science of Software Product Lines

Don Batory

Science of Software Product Lines

Don Batory

ISBN: 979-8-4007-3167-9

DOI: 10.1145/3731302

The book is aimed at both practitioners and advanced students, assuming some familiarity with programming in Java. It draws on the author's unique personal insights from over four decades of experience as an academic at a leading research university as well as a hands-on developer of software tools.

This text is about compositional programming, where programs are constructed by composing prewritten software building blocks. Three modern compositional paradigms are covered, listed in order of decreasing emphasis:

- A **Software Product Line** is a design for a family of programs. Each program is composed from predefined increments of program functionality. The Linux Kernel is the largest known product line whose family size exceeds 10^{2000} distinct programs.
- **Model Driven Engineering** is a general-purpose engineering methodology to support system design, analysis, construction, and evolution. Software designs are expressed as models. Transformations are composed to convert models into other models for analysis, to produce documentation and/or source.
- **Streaming Applications** are dataflow graphs whose nodes (called boxes) are computations with input and output data streams; boxes are wired/composed together to produce a custom program (aka software circuit). Examples are Unix pipe-and-filters and distributed stream processing.

<http://books.acm.org>

「Programming」2017

April 3-6, 2017 · Vrije Universiteit Brussels, Belgium

The Art, Science, and Engineering of Programming
Conference and Journal



The Art, Science, and Engineering of Programming

「Programming」2018

April 9-12, 2018 · Nice, France



「Programming」2024

March 11-14 Lund Sweden

The Art, Science, and Engineering of Programming



Ten Years of 「Programming」

「Programming」2020

4th International Conference on the Art, Science, and Engineering of Programming

March 23-26, 2020
Porto, Portugal



「Programming」2019



「Programming」2026

March 16-20 Munich Germany

The Art, Science, and Engineering of Programming

「Programming」2023

令和五年
プログラミング

The Art, Science, and Engineering of Programming

March 13-17 · Tokyo · Japan



「Programming」

令和五年
プログラミング

March 13-17
The University of Tokyo
Tokyo · Japan

The Art, Science, and Engineering of Programming

<https://programming-conference.org>



The Art, Science, and Engineering of Programming



The Art, Science, and Engineering of Programming



「Programming」2025

June 2-5 Prague Czechia