

DEPARTMENT OF COMPUTER ENGINEERING
PRESENTS
TECHNICAL MAGAZINE

KBT TECH

ODYSSEY

EXPLORE / INNOVATE / BUILD TOMORROW



AI & MACHINE LEARNING

SHAPING A SMARTER TOMORROW

CYBERSECURITY

STRONGER SYSTEMS, SAFER TOMORROW

WEB & MOBILE TECH

CONNECTING IDEAS TO PEOPLE

DATA SCIENCE

TURNING DATA INTO IMPACT

INNOVATION & RESEARCH

IDEAS THAT DRIVE CHANGE

TABLE OF CONTENT

	Editor's Note Welcome to this year's technical issue.	03
01	The Rise of Multi-Agent Systems: Collaboration at Scale KARAN LABHADE	04
02	When Intelligence Steps Into the Real World RAHUL S. GAVIT	13
03	Digital Kumbh SAMIKSHA MUNDADA	20
04	Digital Twins: Creating a Virtual World of the Real World ADWAIT GAVIT	28
05	The Journey Towards Artificial General Intelligence VEDANT GAYAKE	34
06	Prompt engineering NEHUL MAHALE	40
07	Deepfakes and the New Trust Problem PARTH GAUD	47
08	Beyond Coding The Skills Every Engineering Student Needs in 2026 PRACHI JADHAV	53
09	Ransomware Attacks ANUJA MUTHAL	58
10	When the Machine Makes Things Up TEJAS MORE	66

COMPUTER DEPARTMENT

**“Technical
ideas deserve
beautiful
explanations.”**

| Mission

To empower and inspire undergraduate students in Computer Engineering with a foundation of excellence, equipping them to meet the evolving professional and societal needs of business and industry. Through innovative, scientifically designed academic processes, we prepare future engineers to lead and thrive in a technology-driven world.



Ms. Dipika L. Tidke
Editor-in-Chief



Samiksha Mundada
(TY Computer)
Co-Editor

| Vision

To become a center of excellence, shaping world-class engineers who thrive across multi-disciplinary domains. Our mission is to blend cutting-edge technology with real-world industrial and business practices, creating a dynamic academic environment that empowers innovation, fosters collaboration, and equips future leaders to solve tomorrow's challenges.

ARTICLE 01

THE RISE OF MULTI-AGENT SYSTEMS: COLLABORATION AT SCALE

AI agents collaborate to solve complex problems faster, smarter, and at scale.

BY KARAN LABHADE
TY BTECH | COMPUTER | B

INTRODUCTION

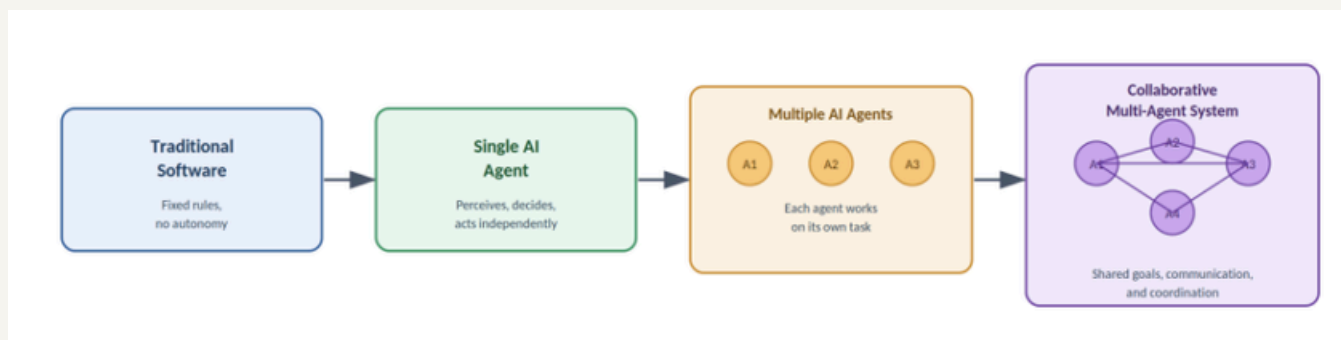


Figure 1. Evolution from rule-based software to a single AI agent, to multiple independent agents, to a coordinated multi-agent system.

Artificial Intelligence (AI), in the context of autonomous software, refers to systems that interpret information, make decisions, and adapt to new situations without being told each step. An AI agent is more specific: software that senses its environment, reasons about what it observes, and acts on its own toward a goal. Early software followed fixed rules; as problems grew more complex, developers built autonomous agents, and eventually systems of many cooperating, specialized agents, known as multi-agent systems (MAS) [1]. Coordinating many such agents toward a shared goal is often called collaboration at scale. MAS matter because modern problems, from large codebases to city traffic, involve many interdependent sub-tasks that specialized, cooperating agents can handle better than one monolithic program.

WHAT ARE MULTI-AGENT SYSTEMS?

A multi-agent system is a set of autonomous agents that interact with one another and with a shared environment to pursue individual or collective goals [1]. Agents sense, act, operate independently, and are driven by internal goals rather than a fixed script. Unlike a single-agent system, where one entity holds all decision-making, a MAS distributes decisions across agents, none of which usually has the full picture, so they rely on cooperation, coordination, or competition to reach an outcome.

Coordination is necessary because agents acting purely on local information can duplicate effort or work against one another; the shared environment supplies what agents sense and absorbs the effects of what they do.

ARCHITECTURE

Practical multi-agent systems are typically built from a recognizable set of parts: individual agents specialized for a sub-task; the environment they sense and affect; a communication mechanism for exchanging messages; a coordination mechanism that decides who does what and when; shared memory or a knowledge base any agent can read or write; tools and external systems such as APIs or databases; and, in many designs, a coordinator or manager agent that breaks the objective into sub-tasks. The coordinator behaves like a project lead, delegating rather than doing every task itself; the communication layer is what turns independent programs into one system; and shared memory lets later agents build on what earlier agents already produced.

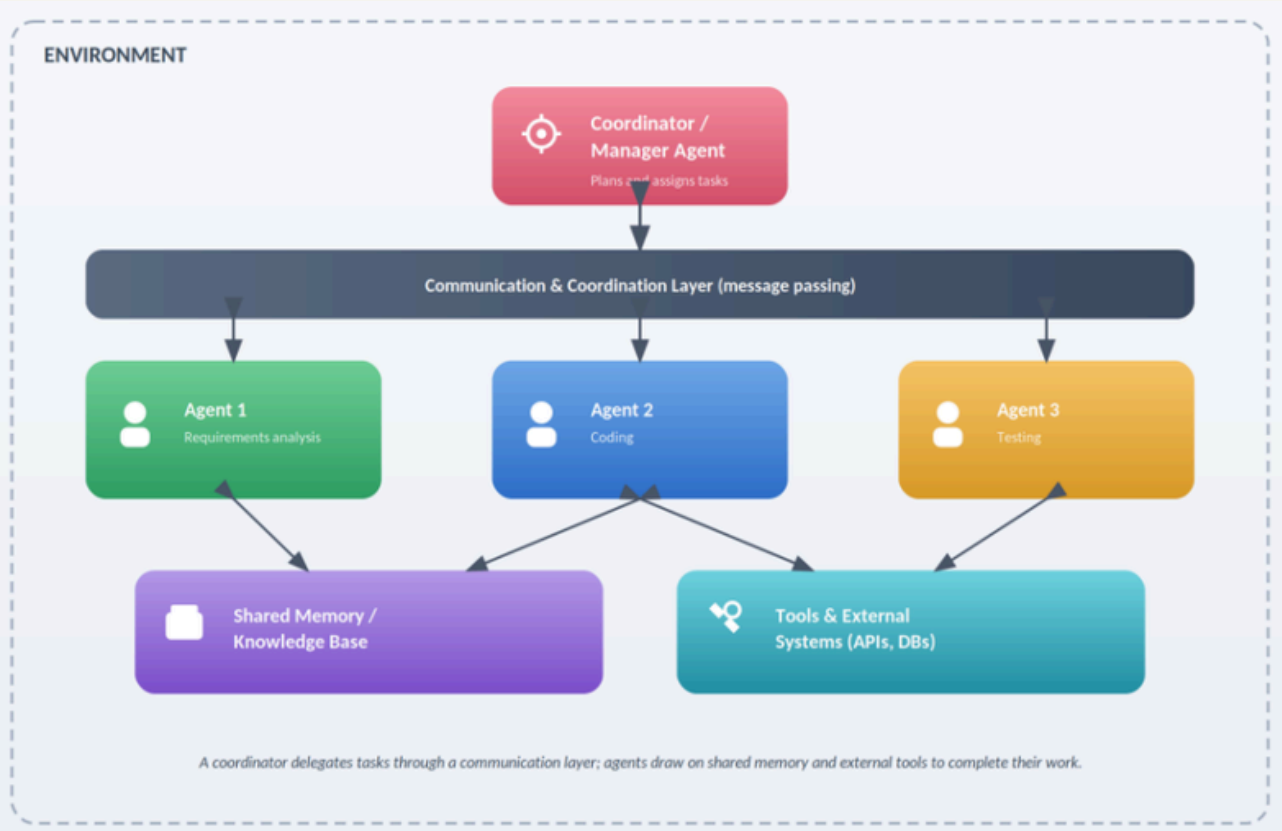


Figure 2. A coordinator assigns tasks through a communication layer; agents draw on shared memory and external tools inside a common environment.

HOW MULTI-AGENT SYSTEMS WORK ?

Most MAS follow a similar loop: receive a complex objective, break it into smaller tasks, assign tasks to matching agents, let each agent work largely independently, exchange intermediate results, combine and review outputs, and return a final result. In AI-assisted software development, for example, one agent might analyze requirements, another write code, another run tests, another fix bugs the tests reveal, and another produce documentation, mirroring how human teams divide work and explaining why software development is a popular MAS testbed [7]. Failed tests typically loop back to debugging rather than the workflow proceeding in one straight line, so a well designed system needs to support this kind of iteration.

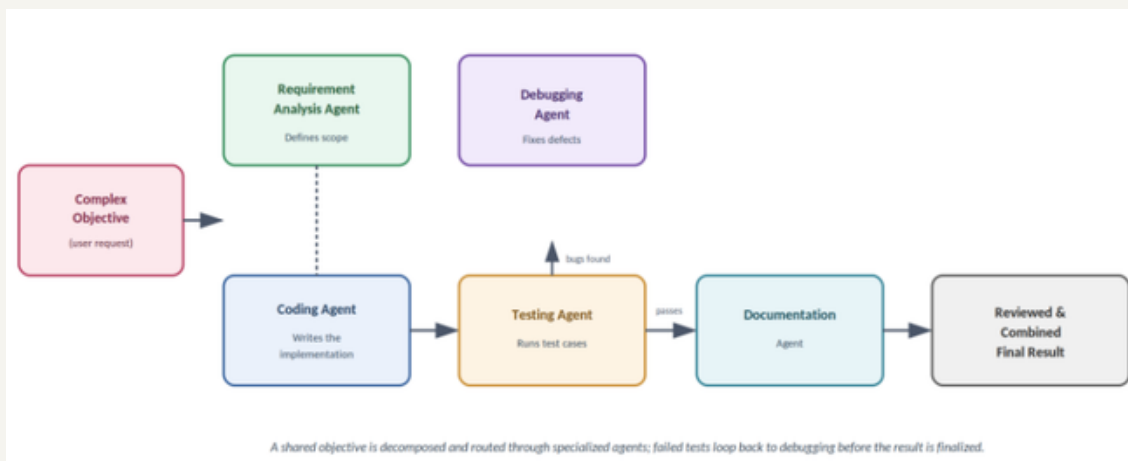


Figure 3. A shared objective is decomposed and routed through specialized agents; failed tests loop back to debugging before the result is finalized.

AGENT COMMUNICATION AND COORDINATION

Agents communicate either directly, exchanging messages that describe requests and results, or indirectly through shared memory that all agents can read and write. Task allocation, deciding who does what, can be planned by a coordinator in advance or negotiated dynamically; done well, it avoids duplicated work and unaddressed sub-tasks. Cooperation is agents sharing a goal; coordination is the protocols, timing rules, and conflict-resolution mechanisms, such as voting or a coordinator with final say, that keep cooperating agents from interfering with each other. Because the number of possible interactions grows much faster than the number of agents, coordination becomes disproportionately harder as a system scales, a challenge revisited below.

ROLE OF LARGE LANGUAGE MODELS IN MULTI-AGENT SYSTEMS

LLMs are now a common reasoning core for agents because they can interpret instructions, plan step by step, and produce structured output without a hand-written rule for every case. The ReAct technique interleaves reasoning traces with concrete actions and observations, letting an LLM-based agent plan, adapt, and call external tools mid task [4]. Because an LLM's training data has a fixed cutoff, many designs pair it with Retrieval-Augmented Generation (RAG), which retrieves relevant external documents at query time and feeds them into the model's context [5]. A separate memory component can further store conversation history or intermediate results beyond one context window, and frameworks such as AutoGen provide the conversational infrastructure, message routing, tool invocation, and human-in-the-loop support, that turns individual LLM agents into a coordinated system [6]. LLMs remain imperfect: they can hallucinate fluent but incorrect text and lack a reliable sense of their own uncertainty, so outputs generally still need verification by another agent, an automated check, or a human, especially where errors would be costly.

COLLABORATION AT SCALE

Realizing the promise of MAS at scale rests on a few ideas working together. Parallel execution lets independent sub-tasks run simultaneously, while specialization lets an agent tuned for one role perform it more reliably than a generalist attempting everything. As agent counts grow, flat all-to-all communication becomes unwieldy, so many systems adopt hierarchies, a top-level coordinator managing team-lead agents that in turn manage workers, with task assignment fixed in advance or decided dynamically. A shared knowledge base avoids re-discovering the same information, and human-agent collaboration remains valuable for judgment calls and approving high-stakes actions. Scaling up improves throughput and ambition, but every added agent is also another possible point of miscommunication, conflicting output, or cost, so collaboration at scale requires coordination and monitoring that grow alongside the agent count, not simply more agents.

APPLICATIONS

- Software engineering: coding, testing, debugging, documentation, and code review split across specialized agents (Section 4).
- Cybersecurity: agents monitor traffic, correlate alerts, and trigger incident response at a volume no single analyst could manage.
- Robotics: multiple robots coordinate exploration, object handling, or formation movement.
- Autonomous vehicles: vehicles coordinate with each other and with traffic infrastructure to reduce collisions and improve flow.
- Healthcare: separate systems assist diagnosis support, literature retrieval, and scheduling alongside clinicians.
- Smart cities: agents manage traffic signals, grid demand, and infrastructure across districts.
- Finance: agents specialize in market analysis, risk assessment, and trade execution, combining perspectives.

Across these domains the same pattern recurs: a large, continuous, or highly varied problem is divided among specialized agents whose combined output covers more of the problem than any one of them could alone.

ADVANTAGES

- Parallel processing reduces overall latency on separable sub-tasks.
- Task specialization often performs a narrow role more reliably than a generalist.
- Scalability: agents can, in principle, be added to handle growing workload.
- Flexibility: agents can be reconfigured or replaced without redesigning the system.
- Fault tolerance: other agents may keep operating if one fails.
- Reduced single-point dependence on one model or one line of reasoning.

CHALLENGES AND LIMITATIONS

Challenge	Why It Occurs	Possible Mitigation
Coordination complexity	Interactions grow much faster than agent count.	Hierarchies or a coordinator agent.
Conflicting objectives	Local goals may work against the system goal.	Shared objectives, negotiation protocols.
Error propagation	One agent's mistake is passed on and amplified.	Verification / review steps between agents.
LLM hallucination	Agents may generate plausible but wrong output.	Retrieval grounding, cross-checking, human review.
Monitoring difficulty	Hard to trace why the system reached a decision.	Log intermediate outputs; interpretable logic.

Beyond the table, MAS raise broader concerns about reliability and reproducibility, since the same objective may not always produce the same sequence of agent interactions, especially with LLM-based agents; security is discussed next; and evaluating emergent, multi-component behavior remains genuinely difficult with standard testing techniques.

SECURITY AND ETHICAL CONSIDERATIONS

Giving an agent access to external tools, databases, or APIs increases what can go wrong if it is compromised, since consequences can extend to real actions such as sending messages or executing code. The OWASP Top 10 for LLM Applications highlights prompt injection, where an attacker's embedded instructions cause the model to act against its intended purpose, and excessive agency, where an agent has more permission to act than its task requires [8]; in a MAS, one compromised agent can pass corrupted instructions to others, spreading the effect of an attack. Reliable agent authentication and authorization, careful handling of shared memory to avoid sensitive-information leakage, human oversight for costly-to-reverse actions, and clear accountability for autonomous decisions are all necessary safeguards as more decision-making is distributed across interacting agents rather than concentrated in one system.

FUTURE SCOPE

Some capabilities are already demonstrated: agents collaborating on coding tasks, structured multi-agent conversations through frameworks such as AutoGen, and retrieval-grounded reasoning, reproducible today, though not yet uniformly reliable in critical deployments. Plausible near-term progress includes more autonomous agents, deeper human-AI collaboration, expanded multi-agent robotics, better AI-assisted engineering tools, and agentic enterprise workflows. More speculative ideas, fully self-organizing agent networks, human-like agent memory, and standardized safety-critical coordination protocols, are discussed in the literature but remain research goals rather than established technology.

CONCLUSION

This article has traced the shift from isolated, single-purpose AI systems toward multi-agent systems in which specialized, communicating agents work together on a shared objective, because many real problems are too large or dynamic for one program to handle well. Specialization lets each agent perform its role competently, while coordination keeps those efforts pointed at the same overall goal; large language models have accelerated this field by giving agents a flexible reasoning core, but have not removed the underlying challenges of communication, task allocation, and conflict resolution. The potential of multi-agent systems spans software engineering, robotics, healthcare, finance, and beyond, but realizing it responsibly depends on continued attention to reliability, security, and human oversight as these systems take on more autonomous, consequential roles.

REFERENCES

- [1] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Harlow, U.K.: Pearson, 2020.

- [3] E. H. Durfee and V. R. Lesser, "Negotiating task decomposition and allocation using partial global planning," in *Distributed Artificial Intelligence*, Vol. II, L. Gasser and M. N. Huhns, Eds. San Francisco, CA: Morgan Kaufmann, 1989, pp. 229–244.
- [4] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. ICLR*, 2023. arXiv:2210.03629.
- [5] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.
- [6] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework," in *Proc. First Conf. on Language Modeling (COLM)*, 2024. arXiv:2308.08155.
- [7] S. Hong et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in *Proc. ICLR*, 2024. arXiv:2308.00352.
- [8] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," OWASP GenAI Security Project, 2025. Available: <https://owasp.org/www-project-top-10-for-large-language-model-applications/>
- [9] T. Guo et al., "Large language model based multi-agents: A survey of progress and challenges," in *Proc. IJCAI, Survey Track*, 2024, pp. 8048–8057.

ARTICLE 02

ARTICLE 02 WHEN INTELLIGENCE STEPS INTO THE REAL WORLD

AI moves beyond screens to perceive, act, and adapt in the physical world.

BY RAHUL S. GAVIT
TY BTECH | COMPUTER | B

INTRODUCTION

For the last decade, artificial intelligence has lived almost entirely on screens. It has written essays, generated images, answered questions, and predicted stock trends all from inside a digital sandbox with no consequences beyond the pixels it produces. That era is ending. A new wave of AI is stepping out of the browser tab and into warehouses, hospitals, roads, and homes. It is called Physical AI, and it may turn out to be the most consequential shift in the field since the arrival of large language models.

Where a chatbot predicts the next word, Physical AI predicts the next action and then actually performs it, with real motors, real sensors, and real-world consequences if it gets it wrong. Nvidia CEO Jensen Huang has called it "the next wave of AI," and companies from Tesla to Boston Dynamics to Nvidia itself are racing to build it. This article explores what Physical AI actually is, how it works, and why every computer engineer graduating in the next few years should understand it.

WHAT EXACTLY IS PHYSICAL AI?

Physical AI refers to artificial intelligence systems that are embodied in machines, robots, humanoids, autonomous vehicles, drones, and automated factory equipment and that sense, reason about, and act directly on the physical world, rather than simply processing text, images and handing a result back to a human [1][2]. Unlike a language model that answers a question and stops, a Physical AI system runs a continuous loop:

1. Perceive: Take in raw signals from cameras, lidar, microphones, and touch sensors.
2. Understand: Build a model of its surroundings: what objects are present, where they are, how they are likely to move.
3. Decide: Reason about what action best achieves its goal .
4. Act: execute the action through motors or actuators.
5. Observe and adjust: Sense the outcome and correct course in real time.

This loop repeats continuously, often many times per second, and every decision has physical consequences a dropped object, a missed step, a collision in a way a generated paragraph of text never does [1][3].

WHY NOW? THE MISSING PIECE FALLS INTO PLACE

Robots have existed on factory floors for decades, so why is "Physical AI" suddenly a buzzword? The answer lies in what changed underneath it. Older industrial robots were pre-programmed: they repeated the exact same welding motion thousands of times a day and broke down the moment something on the line shifted by a few millimeters. They had no real understanding of the world only a fixed script.

What is new is generalizable physical reasoning the ability of a system to learn how the physical world behaves (gravity, friction, momentum, how objects deform) and apply that understanding across many different tasks and environments, rather than being hand-engineered for one narrow job [4].

This shift has been made possible by three converging advances:

- Foundation models and world models that let a robot simulate "if I push this, what happens?" before acting, the same way humans mentally rehearse an action.
- Cheap, powerful sensors (depth cameras, lidar, tactile skins) that give machines rich enough perception to operate in messy, unstructured environments instead of tightly controlled ones.
- Massive simulation environments where robots can practice millions of attempts in virtual worlds before ever touching real hardware, dramatically cutting the cost of trial-and-error learning.

PHYSICAL AI VS. GENERATIVE AI

Generative AI	Physical AI
Operates entirely in the digital domain	Operates in the physical world through hardware
Predicts the next token or pixel	Predicts and executes the next physical action
A wrong output is easy to undo (regenerate)	A wrong action can be irreversible (a dropped patient, a collision)
Trained mostly on internet text and images	Trained on sensor data, simulation, and real-world interaction
Judged by relevance and coherence	Judged by safety, precision, and reliability under real physics

Physical AI is often described as the natural next step after generative AI the same reasoning capabilities that let a model write code or hold a conversation are now being extended to let a machine understand a kitchen counter well enough to safely pick up a glass [1][2].

REAL-WORLD APPLICATIONS ALREADY IN MOTION

Physical AI is not a distant concept confined to research papers, it is already being deployed:



Fig. Application of Physical AI [6]

Humanoid and legged robots: Companies like Boston Dynamics, Figure, and Tesla are building humanoids that can walk, climb stairs, and manipulate objects in spaces designed for humans, aimed at logistics and eventually domestic tasks [3].

Autonomous vehicles: Self-driving cars use a pipeline of object detection, localization, and movement prediction, fusing data from LiDAR, cameras, and radar to navigate unpredictable traffic in real time [3].

Warehouse and factory automation: Automated guided vehicles and robotic arms now adapt to variable inventory layouts instead of following a single fixed script, using computer vision to identify and pick items they were never explicitly trained on [3][5].

Healthcare: Surgical robots and physical assistance devices are beginning to combine precise motor control with real-time sensing to support doctors and patients directly [5].

Agriculture and drones: AI-piloted drones handle aerial imaging, crop monitoring, and precision spraying, adjusting their flight and actuation based on live field conditions rather than a pre-set path [5].

THE TECHNOLOGY STACK BEHIND PHYSICAL AI

Building a Physical AI system draws on several layers of technology working together:

- Perception stack: computer vision (often YOLO- style object detection), sensor fusion of camera, lidar, and radar data, and simultaneous localization and mapping (SLAM) to build a live map of the surroundings [1][2].
- World models: Neural networks trained to predict how the environment will change in response to an action, letting the system "imagine" outcomes before committing to them [2][4].
- Reinforcement learning: It is used heavily in simulation, where an agent is rewarded for successful physical outcomes (grasping an object, maintaining balance) and penalized for failures, refining its policy over millions of simulated trials [2].
- Control systems: The classical engineering layer motor controllers, actuators, and feedback loops that translates a high-level decision ("pick up the cup") into the precise torque and timing needed to execute it smoothly [3].

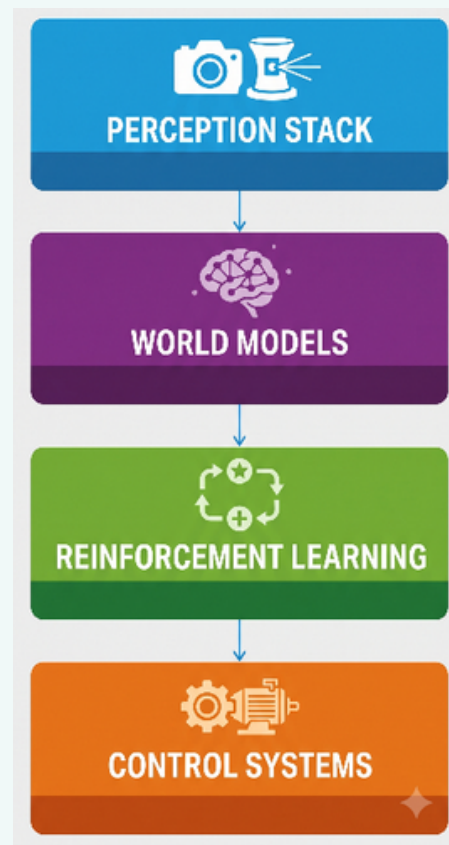


Fig. Technology stack of Physical AI (Image is generated by Gemini)

CHALLENGES THAT KEEP ENGINEERS UP AT NIGHT

Physical AI is exciting, but it faces hurdles that generative AI never had to deal with:

Safety and irreversibility: A chatbot's mistake can be regenerated; a robot's mistake near a human cannot always be undone. This raises the bar for testing and validation enormously [2][5].

The sim-to-real gap: A model trained flawlessly in simulation often struggles when deployed on real hardware, because simulations never perfectly capture friction, lighting, sensor noise, and the sheer unpredictability of the real world [2].

Cost and hardware constraints: Sensors, actuators, and compute capable of running large models on a moving robot are still expensive and power-hungry, limiting how widely these systems can be deployed [2][4].

Data scarcity: Unlike text, which is abundant on the internet, real-world physical interaction data is expensive and slow to collect, since every data point requires an actual robot performing an actual action [1][3].

Ethical and labor questions: As physical automation expands from factories into homes and public spaces, questions about job displacement, liability for accidents, and appropriate human oversight become far more pressing than they are for a text-only AI [4][5].

WHAT THIS MEANS FOR NEXT GENERATION OF ENGINEERS

For students in computer engineering, Physical AI represents a genuine convergence of disciplines that used to be taught in separate departments: machine learning, computer vision, control theory, robotics, and mechanical design now have to be understood together. The engineers who will build the next generation of warehouses, hospitals, and homes will need to be comfortable moving between a neural network's loss function and a robot arm's torque limits in the same afternoon.

This also opens a genuinely wide-open career field. While software roles in generative AI are becoming crowded, physical AI, robotics, and embodied intelligence remain comparatively under-resourced relative to the scale of investment pouring in from companies like Nvidia, Tesla, and Figure making it one of the more promising specializations for students willing to build hands-on skills in both software and hardware.

CONCLUSION

Physical AI marks the point where artificial intelligence stops being confined to a screen and starts sharing our physical space walking our floors, driving our roads, and working alongside us on factory lines. It carries far higher stakes than generative AI ever did, because a mistake is no longer just a bad paragraph it is a real-world event. But it also carries far higher promise: machines that do not just answer our questions, but genuinely help carry the load, literally.

For the computer engineers graduating into this decade, understanding how perception, reasoning, and physical action come together may prove just as essential as understanding how a neural network learns to write a sentence.

REFERENCES

- [1] Voxel51, "Physical AI." <https://voxel51.com/glossary/physical-ai-def>
- [2] IBM, "What Is Physical AI?" <https://www.ibm.com/think/topics/physical-ai>
- [3] Vicon, "What Is Physical AI?" <https://www.vicon.com/resources/blog/what-is-physical-ai/>
- [4] IQT, "What Is Physical AI? A Definition and Framework." <https://www.iqt.org/library/what-is-physical-ai-a-definition-and-framework>
- [5] Xenonstack, "What Is Physical AI? The Complete Guide to AI That Operates the Real World." <https://xenonstack.com/blog/physical-ai>
- [6] Vedyia "What Is Physical AI? A Complete Guide to Intelligent Systems and Real-World Applications" <https://www.vedya.ai/physical-ai>

ARTICLE 02

ARTICLE 03 DIGITAL KUMBH: HOW TECHNOLOGY IS PREPARING NASHIK FOR ONE OF THE WORLD'S LARGEST GATHERINGS

How AI, IoT, and digital infrastructure are transforming Nashik's Kumbh into a smarter, safer, and more connected gathering.

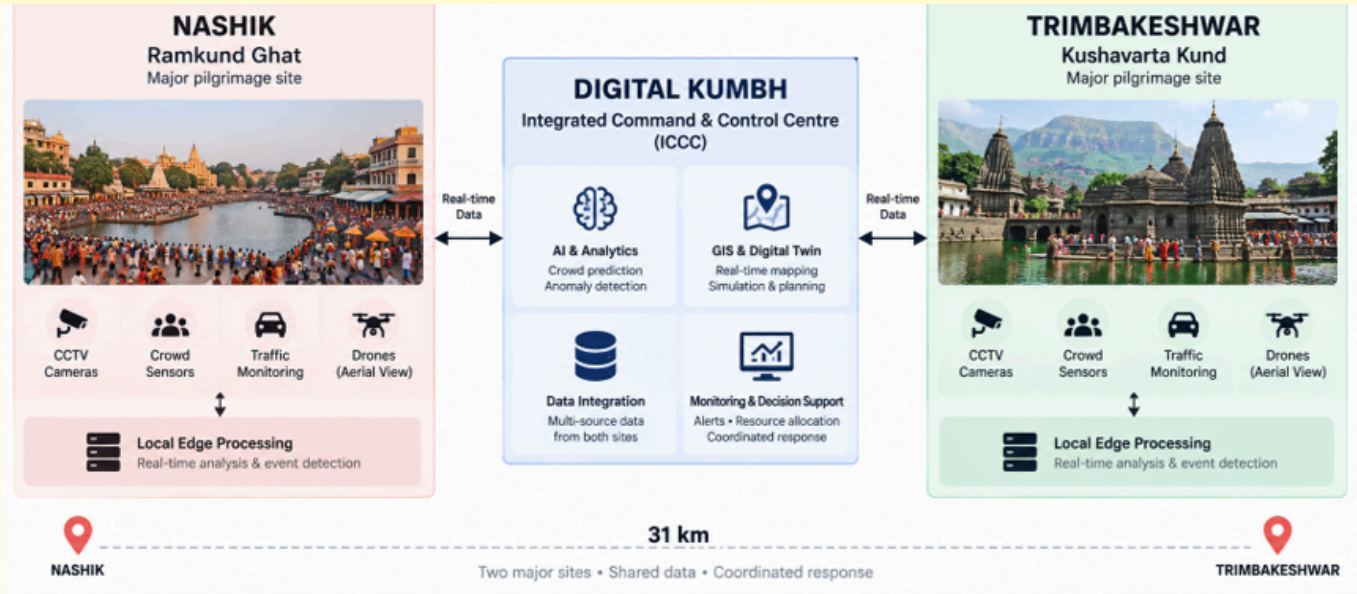
BY SAMIKSHA MUNDADA
TY BTECH | COMPUTER | B

INTRODUCTION

A Kumbh Mela is fundamentally different from a conventional event. It does not take place inside one permanent venue with a fixed capacity. Instead, large temporary settlements, roads, transport routes, public facilities and gathering zones are created to support a rapidly changing population. [1]

For computer engineers, this creates an unusual systems problem. The physical environment continuously produces data: cameras observe movement, sensors report conditions, GPS devices provide location, transport systems generate mobility data, and digital services receive requests from pilgrims. The challenge is to turn this collection of independent data streams into a coherent operational picture.

[1][2]The Digital Kumbh approach can therefore be understood as a computing layer placed over the physical event. Data is acquired from the environment, processed locally or centrally, combined with geographic information, analysed using AI and converted into information that can support authorities, emergency teams and pilgrims. [1][2][2]



FROM A PHYSICAL GATHERING TO DIGITAL SYSTEM

The core idea of a Digital Kumbh is not simply to create an application. It is to create a connected system in which physical events can be represented digitally and analysed in near real time. [1][2][2]

A useful way to understand the system is as a pipeline:

Physical environment → sensing → data processing → AI and analytics → decision support → human response.

The physical environment contains people, vehicles, roads, ghats, facilities and temporary infrastructure. Sensing systems convert activity in that environment into digital signals. Computing infrastructure then filters and combines these signals, while AI models identify patterns that may not be obvious through manual observation. This architecture is especially important for crowd management. A camera feed by itself is only a stream of images. When combined with spatial information, historical data, other camera feeds and movement information, it can become a measurement of density, direction and change over time.

ARCHITECTURE

3.1 Data Acquisition Layer

The first layer collects information from the physical environment. CCTV cameras provide video streams, while IoT sensors can provide environmental and infrastructure measurements. GPS/GNSS data can support location and mobility analysis. Drones can provide an aerial view of selected zones, while transport systems and public-facing services can contribute information. [2][2] The important engineering principle is that these sources are heterogeneous. They generate different data formats, sampling rates and levels of accuracy.

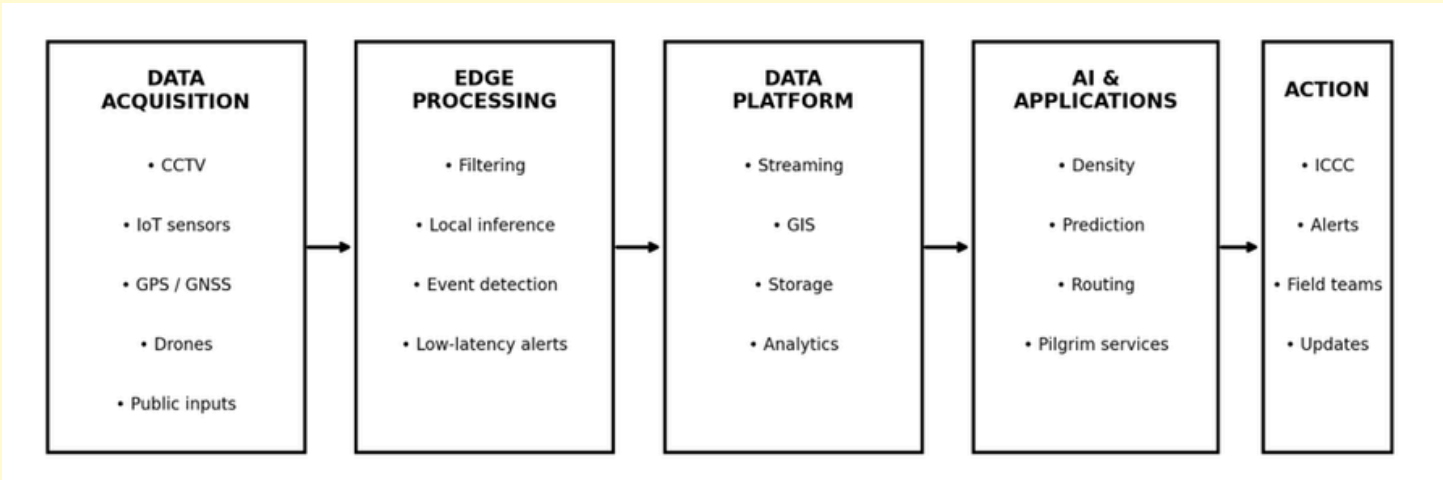
3.2 Edge Processing

Sending every raw video frame to a central data centre is expensive in bandwidth and can introduce latency. Edge computing addresses this by moving selected computation closer to the cameras and sensors. [2] For example, an edge device can analyse a video stream locally and transmit a compact event like “density increasing rapidly in Zone A” instead of continuously transmitting every raw frame. Edge processing is particularly useful during peak periods, when the number of devices and data streams can increase sharply. [2]

3.3 Data Integration and GIS

AI results become much more useful when they are associated with location. A GIS layer can represent roads, ghats, entry and exit points, medical facilities, transport routes and temporary infrastructure. Real-time crowd measurements can then be attached to these geographic regions. [2]

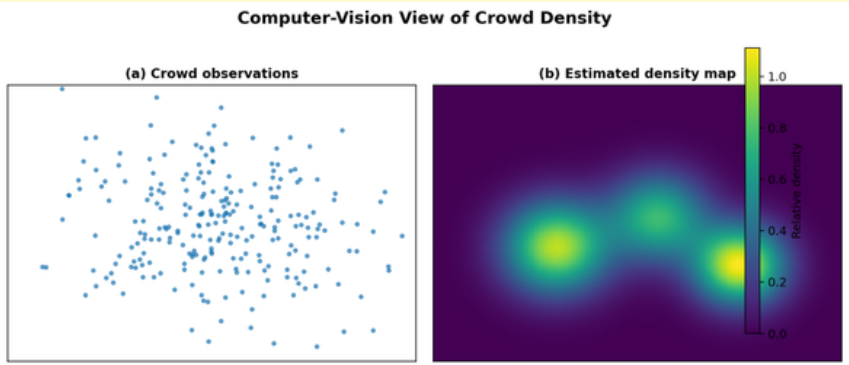
This creates a dynamic map rather than a static map. Instead of only showing where a road exists, the system can represent conditions on that road: density, direction of movement, congestion and changes over time. [2]



AI-BASED CROWD MONITORING

Crowd monitoring is one of the most technically demanding components of a Digital Kumbh. The objective is not merely to count people. The system must estimate density, understand movement and identify abnormal changes early enough for an operational response. [2][2]

In lower-density areas, object-detection models can identify individual people. As density increases, however, people become partially or completely occluded. In such environments, density-estimation methods are more appropriate because they stimate a continuous crowd distribution rather than requiring every individual to be detected. [1][2]



CROWD FLOW AND TEMPORAL PREDICTION

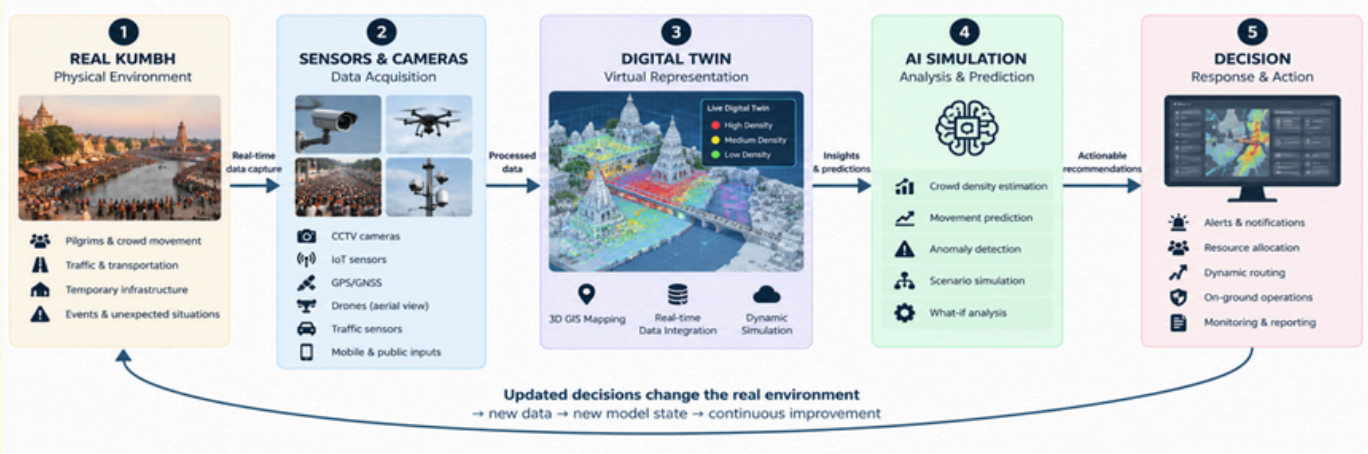
Technique	Primary purpose	Suitable use
Object detection	Detect individual people/vehicles	Lower-density and entry/exit zones
Density estimation	Estimate crowd distribution	High-density areas
Optical flow	Estimate movement	Crowd-flow monitoring
CNN-LSTM	Model temporal patterns	Short-term movement prediction
Anomaly detection	Identify unusual behaviour	Critical zones and alerts

DIGITAL TWIN

A digital twin extends the idea of a live dashboard. Instead of merely displaying current measurements, it maintains a digital representation of the physical environment and updates that representation as conditions change. [2][2]

For a Kumbh-scale deployment, a digital twin can combine geographic information, infrastructure data, crowd measurements, transport conditions and other operational inputs. It can then be used to explore scenarios such as changing a route, restricting access to an area or redirecting movement. [2][2]

The value of a digital twin is therefore not only visualisation. Its greater potential is simulation: testing possible decisions in the digital environment before applying them to the physical environment. [2]



INTEGRATED COMMAND AND CONTROL

AI models cannot manage a physical event by themselves. Their outputs need to reach people and systems capable of acting on them. An Integrated Command and Control Centre (ICCC) can act as the operational layer connecting analytics with field response. [2][2] A command centre can combine live camera views, maps, alerts, transport information, incident reports and AI-generated insights. Operators can then evaluate the situation and coordinate with police, traffic teams, medical services, emergency personnel and other departments. [2] [2] The engineering challenge is to prevent information overload. A system that produces thousands of alerts without prioritisation can be as difficult to use as a system that produces no alerts. Event classification, severity levels and clear visualisation are therefore important parts of the software design. [2]

PILGRIM-FACING TECHNOLOGY

The Digital Kumbh concept also extends beyond government and command-centre systems. Pilgrims need information in a form that is accessible under real-world conditions. [1] A multilingual AI assistant can provide information about routes, facilities, crowd conditions and frequently requested services. A voice interface can be particularly useful when users are walking, when typing is inconvenient, or when users are more comfortable speaking than reading. [1] From a systems perspective, this creates another important problem: the information shown to users must be generated from the same trusted operational data used by the control system. Otherwise, different parts of the digital ecosystem may provide conflicting information. [1]

CYBERSECURITY

A digital infrastructure operating at this scale also creates a large cybersecurity surface. Cameras, sensors, applications, APIs, databases and command-centre systems all become potential attack points. [1][1] Security should be considered at every layer, Device authentication, encrypted communication, access control, secure APIs, logging, network segmentation. Privacy is equally important when systems process images, locations or information related to people. The technical objective should be to collect and retain only the information required for a purpose. [1][1]

KEY TECHNICAL CHALLENGES

Challenge	Technical issue	Possible engineering response
Extreme density	Occlusion reduces individual detection accuracy	Density estimation + multiple viewpoints
Network congestion	Large numbers of devices generate high traffic	Edge processing + prioritised streams
Latency	Delayed information reduces operational value	Local inference + low-latency pipelines
System failure	A single failure should not stop the platform	Redundancy + failover + replication
Alert overload	Too many alerts can overwhelm operators	Severity classification + intelligent filtering
Changing infrastructure	Routes and facilities may change during the event	GIS updates + dynamic data models

PROPOSED END-TO-END DATA FLOW

- The complete system can be viewed as a continuous loop:
1. Cameras, sensors, and GPS observe the physical environment.
 2. Edge devices perform initial filtering and AI inference.
 3. Relevant events and measurements are sent to the central data platform.
 4. GIS associates those measurements with physical locations.
 5. AI models estimate density, movement and patterns.
 6. The digital twin represents the current state and can simulate possible changes.
 7. The command centre receives alerts and operational information.
 8. Field teams or digital services communicate the response.
 9. The resulting change in the physical environment generates new data, closing the loop. [2][2]

CONCLUSION

Digital Kumbh represents a shift from treating a mega-event as a collection of physical arrangements to treating it as a large-scale cyber-physical system. [1][2][2] For Nashik–Trimbakeshwar Kumbh 2027, technologies such as AI-based crowd analysis, GIS, edge computing, real-time data platforms, digital twins, distributed systems and multilingual digital services can form interconnected layers of a digital infrastructure. [1][2][2][1][2]

The most interesting engineering problem is not simply how to deploy more technology. It is how to make thousands of heterogeneous devices, data streams, AI models and human decision-makers operate as one reliable system under rapidly changing conditions.

In that sense, Digital Kumbh is a real-world demonstration of a central idea in modern computer engineering: software is no longer confined to the screen. It increasingly observes, models and interacts with the physical world

REFERENCES

- [1] Divisional Commissioner Office, Nashik, Government of Maharashtra, “About Nashik–Trimbakeshwar Authority,” Nashik–Trimbakeshwar Kumbh Mela Authority, 2026.
- [2] Divisional Commissioner Office, Nashik, Government of Maharashtra, “Meeting Proceedings,” including the 22 May 2025 meeting regarding AI/ML-based analysis on CCTV video feeds 2025–2026.
- [3] MIT Media Lab, “MIT Kumbhathon: A City-Stack Powered by Decentralized AI.
- [4] National Institute of Standards and Technology (NIST), “Digital twins,” NIST, 2025–2026.
- [5] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes,” arXiv:1802.10062, 2018.
- [6] M. A. Hossain, K. Cannons, D. Jang, F. Cuzzolin, and Z. Xu, “Video-Based Crowd Counting Using a Multi-Scale Optical Flow Pyramid Network,” Proc. Asian Conference on Computer Vision (ACCV), 2020.
- [7] C. Mahmoudi, A. Battou, and F. Mourlin, “Formal Definition of Edge Computing: An Emphasis on Mobile Cloud and IoT Composition,” Proc. IEEE Third International Conference on Fog and Mobile Edge Computing, 2018. doi: 10.1109/FMEC.2018.8364042.
- [8] M. Fagan, K. Megas, J. Marron, et al., “IoT Device Cybersecurity Guidance for the Federal Government: Establishing IoT Device Cybersecurity Requirements,” NIST SP 800-213, National Institute of Standards and Technology, 2021.

ARTICLE 04

ARTICLE 04 DIGITAL TWINS: CREATING A VIRTUAL WORLD OF THE REAL WORLD

Digital twins mirror real-world systems virtually, enabling smarter monitoring, simulation, and decision-making.

BY ADWAIT GAVIT
TY BTECH | COMPUTER | B

INTRODUCTION

Picture a wind turbine spinning somewhere off the coast of Denmark. Now picture a second, invisible version of that same turbine — one that exists only as data, updating itself in real time as sensors on the physical machine report vibration, temperature, and wear. Engineers can spin that invisible turbine forward in simulated time, stress-test it against a storm that hasn't happened yet, and know it'll fail three years before it actually does. That invisible twin isn't a 3D model sitting idle in some CAD file. It's alive, in the sense that it keeps listening to its physical counterpart and keeps changing because of it.

That's the core idea behind a digital twin, and it's quietly become one of the more consequential ideas in engineering over the last decade — not because it's new (the concept traces back to NASA's Apollo era practice of building physical mirror systems on the ground [1]), but because sensors, connectivity, and computing finally got cheap enough to make it real-time and everywhere, not just in a space program's budget.

WHAT IS A DIGITAL TWIN?

Put simply: a digital twin is a virtual representation of a real object or system that receives data from the real world. That's it — no more, no less. The formal engineering literature adds a bit more texture: a digital twin has been described as a set of virtual constructs detailed enough that anything you could learn by inspecting the physical object, you could also learn from its twin. What separates a digital twin from an ordinary simulation or 3D model is that live data link back to reality. A CAD model of a bridge is a static picture.

A digital twin of that same bridge is constantly being fed sensor data — strain, temperature, traffic load — and updates itself accordingly, so the virtual bridge always reflects the current condition of the real one, not the condition it was in when someone last opened the file.

Digital twins now span a wide range of domains: factories and production lines, individual vehicles and fleets, entire cities, and — increasingly — the human body itself, where "human digital twins" model a patient's physiology for personalized diagnosis and treatment.

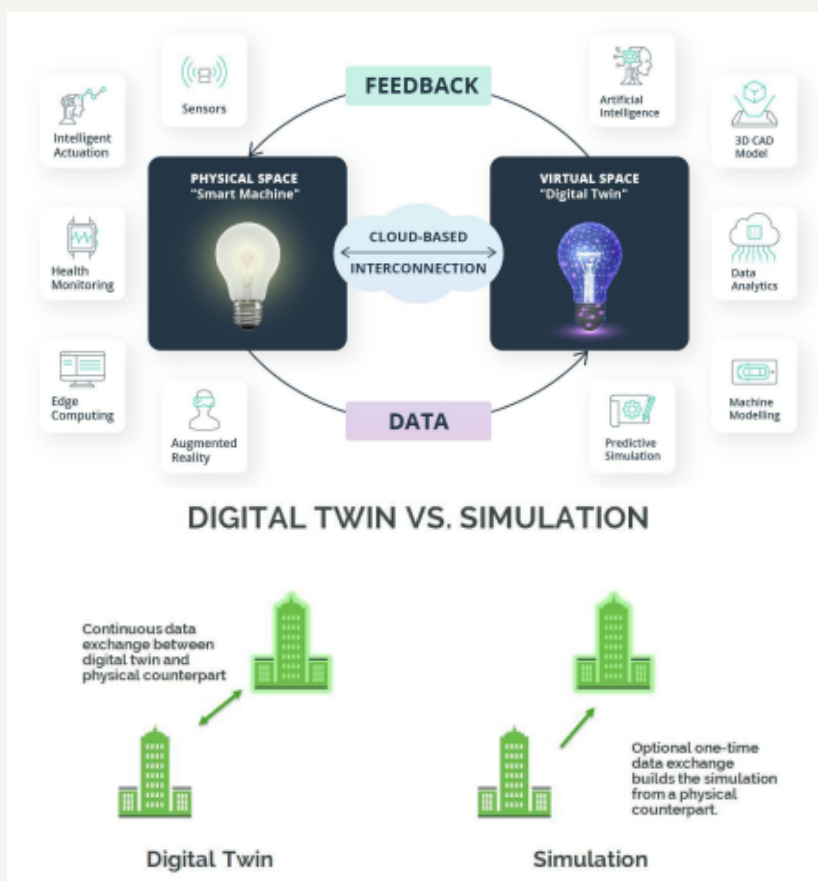
HOW A DIGITAL TWIN ACTUALLY WORKS

Strip away the industry-specific jargon and most digital twins run on a fairly consistent loop:

- A physical entity (a machine, a building, a person, a city block) is fitted with sensors
- Those sensors stream real-time data — vibration, temperature, location, usage — up to a virtual model
- The virtual model updates itself to mirror the current state of the physical entity

- Engineers or algorithms run simulations and predictive analytics on the virtual model — testing scenarios, forecasting failure, optimizing performance
- Insights flow back down to the physical entity — a maintenance alert, a design tweak, an operational change
- The physical entity is adjusted, and the loop begins again

This closed loop — physical entity feeding data up, insight flowing back down — is what separates a twin from a one-off simulation. A simulation answers a question once. A digital twin keeps answering it, forever, because it never stops listening.



WHY IT'S EXPLODING RIGHT NOW

A few forces converged at once to make digital twins practical at scale rather than a NASA-budget luxury:

- Cheap, ubiquitous sensing — IoT sensors that once cost hundreds of dollars now cost cents, making it economical to instrument an entire factory floor or vehicle fleet, not just a single prototype
- Cloud and edge computing — running a continuously updating simulation used to require dedicated infrastructure; now it can run on rented cloud compute or even at the edge, near the device itself
- AI and machine learning — predictive models trained on the twin's historical data can forecast failures and optimize performance far beyond what rule-based simulation could do alone
- 5G and low-latency networks — for twins that need to react in near real time (autonomous vehicles, robotics), the network finally caught up to the ambition

THE RISKS NOBODY SHOULD SKIP

The enthusiasm around digital twins comes with real, well-documented friction:

- Interoperability chaos — a systematic review of urban digital twins identified over a dozen distinct technical challenges, and found that disparate semantic standards — different systems literally not agreeing on what a piece of data means — were among the most severe [4]. Two cities' "digital twins" often can't talk to each other, or even to different departments within the same city.
- Security and privacy exposure — a twin that mirrors a real system in real time is also, by definition, a live map of that system's vulnerabilities; a comprehensive survey on digital twin architecture flags security and privacy as unresolved, cross-cutting concerns rather than solved problems.
- Definitional fuzziness — researchers who mined over 15,000 publications on digital twins in the built environment found there still isn't consensus on what actually counts as a "true" digital twin versus a glorified 3D model with a data feed bolted on.

- Fidelity drift — if sensor data is noisy, delayed, or missing, the twin quietly stops mirroring reality while still looking authoritative — a dangerous gap when decisions are being made off its output.
- Cost and organizational buy-in — building the sensing and integration layer is expensive, and non-technical barriers like unclear business value and mismatched incentives across departments show up just as often as technical ones.

TOWARD RESPONSIBLE DIGITAL TWINS

Researchers aren't arguing against digital twins — they're arguing for structure around them. Two threads worth knowing: Semantic and systems integration, rather than ad hoc data pipelines, is being pushed as the fix for the interoperability problem: getting different systems to agree on shared meaning for data, not just shared file formats, so twins across departments or organizations can actually exchange information without manual translation [5]. Standardized definitions and lifecycle frameworks — efforts to pin down what a digital twin is and map its risks to specific stages of its lifecycle (design, deployment, operation, retirement) — aim to stop teams from building twins that are digital twins in name only, and to make the technical and non technical risks visible early rather than discovered after deployment [4][7].

WHY SHOULD YOU CARE AS A STUDENT?

If you're in engineering, computer science, or anything adjacent, you're going to run into digital twins whether you plan a career around them or not — they now sit at the intersection of IoT, cloud computing, simulation, and AI, which means the skillset (sensor integration, real-time data pipelines, model validation) transfers across manufacturing, healthcare, civil engineering, and urban planning alike. Employers increasingly need people who can reason about when a twin is trustworthy and when it's drifted from reality — not just people who can wire up sensors and call it done.

CONCLUSION

A digital twin, at its heart, is a deceptively simple promise: a version of the real world that never stops listening and never goes stale. That promise is powerful enough to reshape how factories are run, how cities are planned, and how patients might one day be treated — but the research is equally clear that the promise only holds if the underlying data pipeline is trustworthy, the systems can actually talk to each other, and the definition of "digital twin" isn't stretched thin enough to mean anything with a sensor attached to it. The virtual world only stays useful as long as it keeps telling the truth about the real one.

REFERENCES

- 1] M. Grieves, "Origins of the Digital Twin Concept," White paper, Florida Institute of Technology, 2016.
- [2] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in *Transdisciplinary Perspectives on Complex Systems*, Springer, 2017. [3] M. W. Lauer-Schmaltz et al., "Towards the Human Digital Twin: Definition and Design — A Survey," *arXiv:2402.07922*, 2024.
- [4] B. Lei and F. Biljecki et al., "Challenges of Urban Digital Twins: A Systematic Review and a Delphi Expert Survey," *Automation in Construction*, vol. 147, 2023.
- [5] H. Y. Quek et al., "The Conundrum in Smart City Governance: Interoperability and Compatibility in an Ever-Growing Ecosystem of Digital Twins," *Data & Policy*, Cambridge University Press, 2023. [6] Y. Wang et al., "A Survey on Digital Twins: Architecture, Enabling Technologies, Security and Privacy, and Future Prospects," *IEEE Internet of Things Journal*, vol. 10, no. 17, 2023. [7] M. Abdelrahman et al., "What Is a Digital Twin Anyway? Deriving the Definition for the Built Environment from over 15,000 Scientific Publications," *arXiv:2409.19005*, 2024

ARTICLE 05

ARTICLE 05 THE JOURNEY TOWARDS ARTIFICIAL GENERAL INTELLIGENCE

AI is evolving from specialized tools toward systems capable of learning, reasoning, and adapting across tasks.

BY VEDANT GAYAKE
TY BTECH | COMPUTER | B

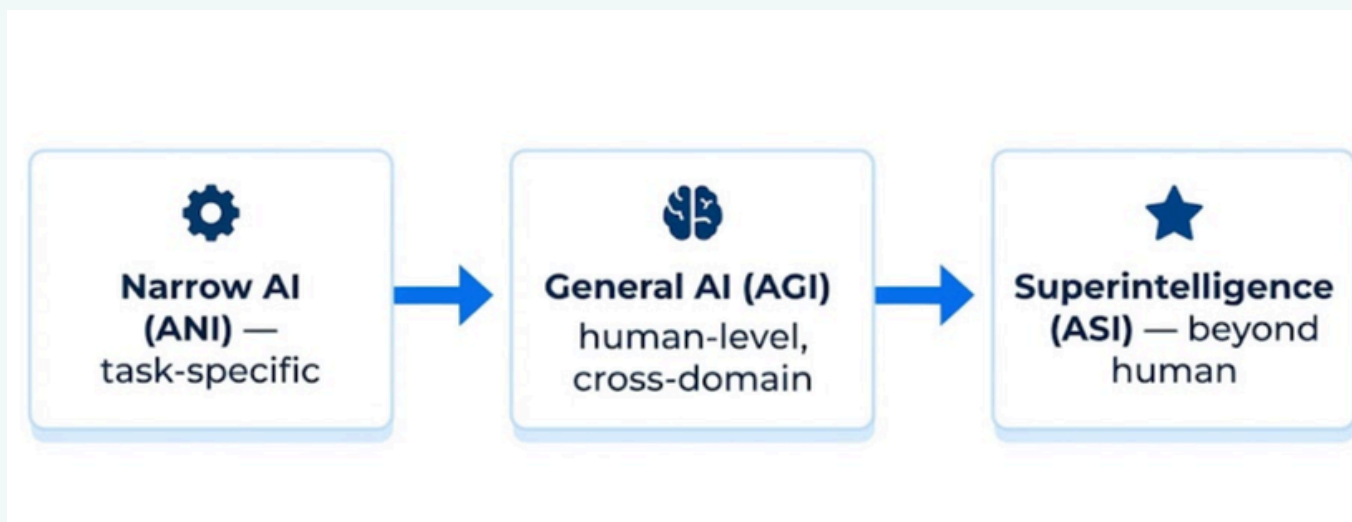
INTRODUCTION

For decades, one question has fascinated computer science: can a machine think like a human, not merely play chess or sort photos, but reason, learn, and adapt across domains it has never encountered before? Today's AI can write code, assist in medical diagnosis, and hold conversations that feel remarkably human. Yet when the same systems are asked to perform unfamiliar physical tasks or recognize contradictions in their own reasoning, their limitations become apparent. This gap between machines that are highly capable at specific tasks and machines that can adapt across a broad range of domains is exactly what Artificial General Intelligence (AGI) seeks to address. AGI is not a product or a single technological breakthrough; it is a direction of research. In recent years, the topic has moved from philosophy and science fiction into peer reviewed research, major technology laboratories, and government policy discussions [1][5].

NARROW AI, GENERAL AI, AND THE SPACE BETWEEN

Many of the AI systems we use today, including recommendation engines, spam filters, and large language models such as ChatGPT, are specialized systems whose capabilities remain limited outside the tasks and environments in which they have been developed and evaluated. Artificial General Intelligence, by contrast, is commonly described as a system capable of understanding, learning, and applying knowledge across a wide range of tasks at a level comparable to humans, not because it has been individually trained for every task,

but because it possesses a more general capacity for learning and adaptation [1]. Beyond AGI lies a more speculative concept: Artificial Superintelligence (ASI), referring to a system that exceeds human intellectual capabilities across virtually all domains. Most current research, however, is focused on the transition from specialized AI systems toward increasingly general and adaptable forms of intelligence.



HOW DO YOU EVEN MEASURE GENERAL?

One of the hardest problems in AGI research is not simply building more capable models, but determining what intelligence actually means. In 2019, researcher François Chollet published *On the Measure of Intelligence*, arguing that intelligence should be understood not primarily as accumulated knowledge or skill, but in terms of skill acquisition efficiency: how efficiently a system can learn to solve a problem it has never encountered before, using limited prior experience [3]. To operationalize this idea, the ARC benchmark was introduced as a collection of visual, puzzle like tasks designed to test a system's ability to identify abstract patterns and generalize from very few examples. Such tasks are often straightforward for humans but difficult for systems that rely heavily on memorization or pattern matching. Subsequent versions, including ARC AGI 2, have been designed to make superficial shortcuts increasingly difficult and to place greater emphasis on genuine reasoning and generalization [3][4]. Alongside benchmarks, Google DeepMind researchers proposed another framework for discussing progress toward AGI. Their 2024 paper, *Levels of AGI: Operationalizing Progress on the Path to AGI*, classifies AI systems using two independent dimensions: performance, or how well a system performs, and generality, or how broadly its capabilities transfer across tasks. The framework describes a progression from "No AI" to "Superhuman," while also considering a separate dimension of autonomy, ranging from AI systems that primarily function as tools to systems capable of operating as agents [1].

"SPARKS" OF AGI: THE DEBATE OVER GPT 4'S CAPABILITIES

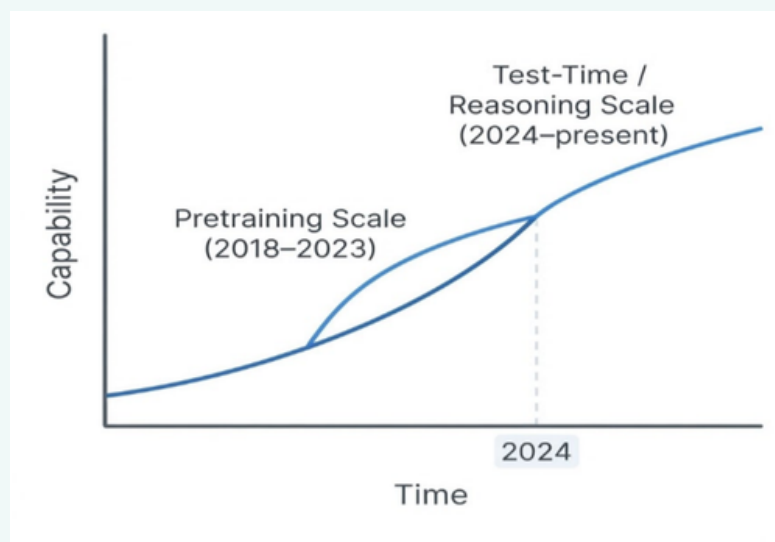
In March 2023, a team at Microsoft Research published a 155 page paper titled Sparks of Artificial General Intelligence: Early Experiments with GPT 4. The authors reported that GPT 4 could solve novel and difficult problems across areas including mathematics, coding, medicine, law, and psychology without task specific training. They described its performance as "strikingly close to human level" in several areas and highlighted substantial improvements over previous models [2].

The paper generated significant discussion. Critics, including cognitive scientist Gary Marcus, argued that broad performance across many tasks does not by itself establish general intelligence, particularly when a system remains unreliable or inconsistent on unfamiliar problems. This debate highlights an important distinction: a system may demonstrate impressive breadth while still lacking the reliability, adaptability, and robust generalization associated with human intelligence.

This tension between impressive breadth and inconsistent reliability remains one of the central unresolved questions in discussions surrounding AGI.

Levels of AGI		
	Narrow	General
Level 0	No AI	Emerging
Level 1	Competent	Competent
Level 2	Expert	Competent
Level 3	Virtuoso	Expert
Level 4	Superhuman	Virtuoso
Level 5	Superhuman	Superhuman

THE ENGINE BEHIND THE CLIMB: SCALE AND ITS LIMITS



Much of the recent progress in AI has been driven by scaling. OpenAI's 2020 paper, *Scaling Laws for Neural Language Models*, showed that performance generally improves as model size, dataset size, and training compute increase [5]. This helped make scaling a central strategy in modern AI research. However, scale alone has limitations.

Researchers now also use inference-time computation, allowing reasoning focused models to spend more computation on difficult problems. This has contributed to progress on reasoning benchmarks such as ARC AGI [3][4]. Other approaches, including neurosymbolic and hybrid systems, combine neural networks with symbolic reasoning, suggesting that achieving broader intelligence may require more than simply building larger models.

WHY WE'RE NOT THERE YET?

Despite rapid progress, no widely accepted scientific consensus establishes that current AI systems constitute AGI. Even reasoning focused models that have demonstrated significant progress on ARC AGI 1 face substantial challenges on ARC AGI 2, which was designed to reduce the effectiveness of shortcuts and require more robust forms of generalization [4]. Current AI systems also face limitations involving persistent memory, reliable understanding of real world environments, physical interaction, and the ability to consistently recognize the boundaries of their own knowledge. These limitations suggest that today's large language models demonstrate some characteristics associated with general intelligence without yet establishing the full range of capabilities commonly associated with AGI [1][2][4]. The distinction is important: demonstrating intelligence on a wide variety of benchmarks is not necessarily equivalent to possessing a reliable, and transferable form of intelligence.

WHY THIS MATTERS

ESPECIALLY FOR STUDENTS

For a computer engineering student, AGI research is connected to several active areas of technology and research, including machine learning, cognitive science, benchmarking, AI safety, and policy. The rapid improvement of AI capabilities also raises questions about reliability, misuse, economic disruption, and human control. In late 2023, a consensus paper titled *Managing AI Risks in an Era of Rapid Progress*, signed by researchers including Yoshua Bengio, Geoffrey Hinton, Stuart Russell, and Daniel Kahneman, argued that as AI systems become more capable and autonomous, the potential for large scale social harm may increase and that research into AI safety and risk management needs to keep pace with advances in AI capabilities [6]. The paper was subsequently published in *Science* under the title *Managing Extreme AI Risks Amid Rapid Progress* [6]. For students entering this field, the opportunity is therefore not limited to building more capable models. It also includes developing better evaluation methods, understanding the limitations of AI systems, improving safety techniques, and contributing to the responsible governance of increasingly capable technologies.

CONCLUSION

The journey toward AGI is not a single race with a clearly defined finish line. Recent advances in AI have demonstrated impressive progress in reasoning, learning, and generalization, but important challenges remain in reliability, adaptability, and real world understanding. If AGI emerges, it may not arrive through a single breakthrough, but through a gradual development of increasingly capable and general systems. Understanding systems and developing them responsibly will be as important as achieving greater intelligence itself.

REFERENCES

- [1] M. R. Morris et al., "[Levels of AGI: Operationalizing Progress on the Path to AGI](#)," [Proceedings of the 41st International Conference on Machine Learning \(ICML\)](#), PMLR 235, 2024.
- [2] S. Bubeck et al., "[Sparks of Artificial General Intelligence: Early Experiments with GPT 4](#)," [Microsoft Research](#), 2023.
- [3] F. Chollet, "[On the Measure of Intelligence](#)," 2019.
- [4] F. Chollet et al., "[ARC AGI 2: A New Challenge for Frontier AI Reasoning Systems](#)," 2025.
- [5] J. Kaplan et al., "[Scaling Laws for Neural Language Models](#)," [OpenAI](#), 2020.:
- [6] Y. Bengio, G. Hinton, A. Yao, et al., "[Managing AI Risks in an Era of Rapid Progress](#)"

ARTICLE 06

ARTICLE 06 PROMPT ENGINEERING

Crafting precise instructions to get smarter, more reliable results from AI.

BY NEHUL MAHALE
TY BTECH | COMPUTER | B

INTRODUCTION

Ask an AI chatbot a question, and you might get a brilliant answer. Ask it the same question a different way, and you might get something useless. What changed? Not the AI. Not the topic. Just the words you used to ask. That small but powerful skill — choosing the right words to get the best answer from an AI — is called prompt engineering, and it is quickly becoming one of the most useful skills a student can learn.

“The AI didn’t get smarter. You just learned how to ask better.”

In this article, we break down what prompt engineering actually means, why it matters for students in 2026, and the simple techniques you can start using today — no coding required.

WHAT EXACTLY IS PROMPT ENGINEERING?

- In the simplest terms, prompt engineering is the skill of writing clear, well-structured instructions for an AI model so that it gives you the most accurate and useful answer possible [1]. AI language models like ChatGPT, Claude, and Gemini are extremely powerful, but they are also very literal. They respond to exactly what you type — no more, no less. A vague question gets a vague answer. A clear, detailed question gets a sharp, useful answer.
- Think of an AI model as an incredibly talented new intern. The intern is brilliant, has read almost everything on the internet, and works at superhuman speed. But the intern cannot read your mind. If you say “make a poster,” you might get anything. If you say exactly what the poster is for, who will see it, and what tone it should have, you get something you can actually use. Prompt engineering is simply the practice of being that clear, every single time.

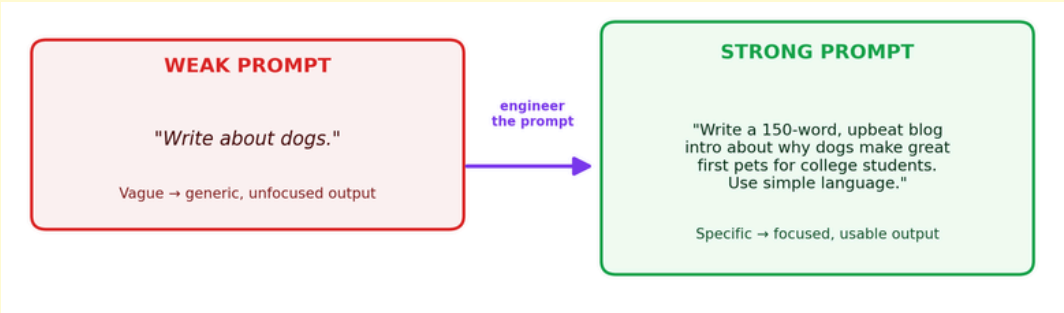


Figure 1: The same AI, given two different prompts, produces two very different results.

WHY SHOULD STUDENTS CARE?

You might be thinking: “I just type my question and the AI figures it out. Why do I need to learn anything extra?” Here is why it matters. Across workplaces already using AI daily, people who know how to prompt well report major gains in speed and quality compared to those who type the first thing that comes to mind [3]. For a student, this translates directly into saved time and better grades. A well-engineered prompt can turn a confusing textbook chapter into a simple explanation, turn a rough idea into a polished essay outline, or turn messy notes into a clean study guide. The difference is rarely the AI tool you use — it is almost always how you ask.

Experts describe prompt engineering as part craft and part science — a mix of intuition for what works and testable techniques you can learn and repeat, much like learning to write a good search query or a clean line of code [2].

A Skill Worth Learning Early

Some experts now compare prompt engineering to learning to write a good database query: a small technical skill that quietly affects the quality of everything you build on top of it [2]. Learning it now, while it is still a fairly new skill for everyone, puts students ahead rather than behind.



Figure 2: Reported gains from using structured prompting techniques instead of casual, one-line questions [3].

FOUR CORE TECHNIQUES ANYONE CAN LEARN

You do not need to be a computer science major to use these techniques. They are simply different ways of structuring what you type. Researchers have catalogued more than fifty such techniques, but four of them cover most everyday situations a student will run into [4].

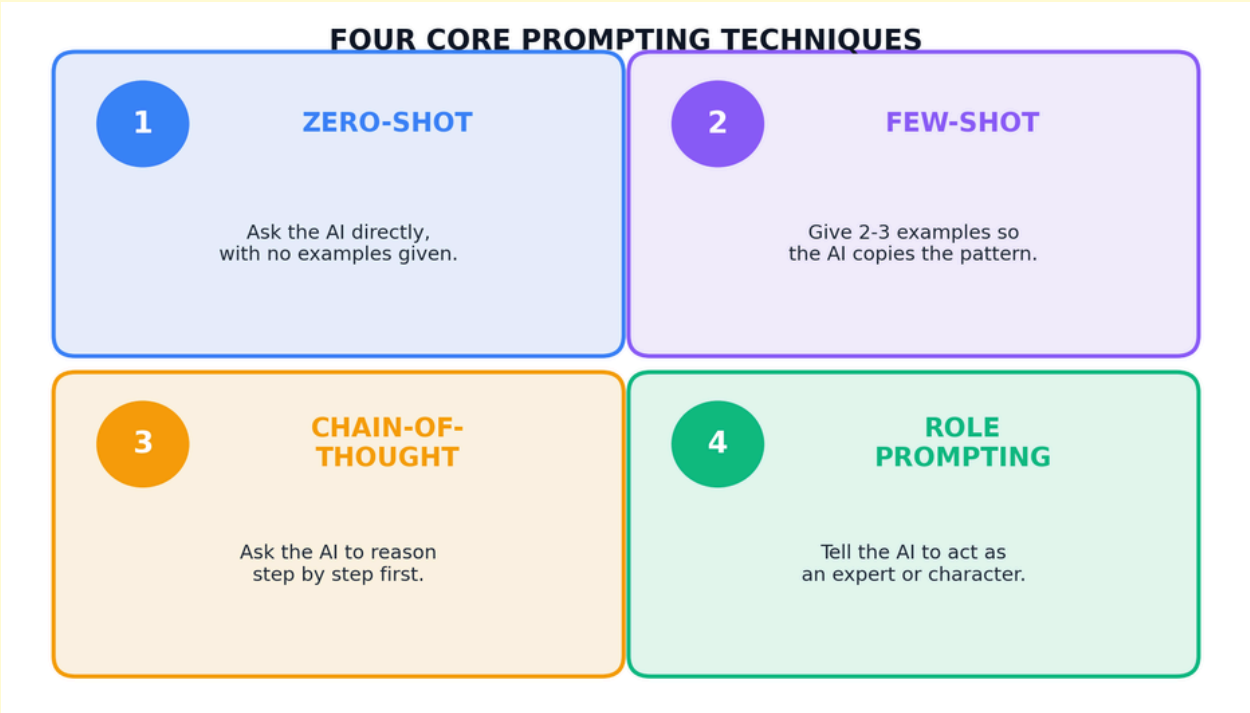


Figure 3: Four beginner-friendly prompting techniques you can start using immediately.

Zero-shot prompting is the simplest: you just ask directly, with no examples, and trust the AI to understand the task [1]. Few-shot prompting is one step up — you give the AI two or three examples of what you want, and it copies the pattern, which works well for things like formatting or tone. Chain-of-thought prompting means asking the AI to “think step by step” before answering, which helps enormously with maths, logic, and multi-part problems [1]. Finally, role prompting means asking the AI to respond as if it were a specific expert — for example, “act as a patient biology tutor” — which often makes explanations clearer and more focused.

Common Mistakes Beginners Make

Most disappointing AI answers trace back to one of a handful of avoidable habits. Watch out for these four before you blame the AI itself:



PROMPT ENGINEERING IN FOUR SIMPLE STEPS

Good prompting is not a one-shot event. It is a short loop: you write, you check the result, and you adjust. Professionals call this iterative refinement, and it is one of the core techniques behind every well-crafted AI interaction [2].

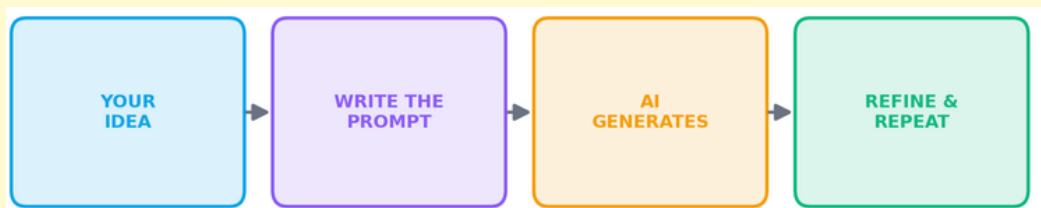


Figure 4: The basic prompt engineering loop — write, generate, and refine.

Quick Tips for Better Prompts

- Be specific: mention the length, tone, audience, and format you want.
- Give context: tell the AI what the content is for, and who will read it.
- Ask for structure: request headings, bullet points, or a table when useful.
- Iterate: if the first answer is not right, tell the AI exactly what to fix.
- Break big tasks down: split a complex request into smaller, clearer steps.

TRY IT YOURSELF — THE PROMPT FORMULA

“Act as a [ROLE]. Help me [TASK] for [AUDIENCE]. Keep it [TONE] and [LENGTH]. Format the answer as [FORMAT].”

Example: “Act as a friendly maths tutor. Help me understand integration by parts for a first-year student. Keep it simple and under 200 words. Format the answer as three numbered steps with one example.”

WHERE IS PROMPT ENGINEERING HEADED?

The field is evolving fast. In 2026, prompting has split into two related skills: casual prompting for everyday chatting with an AI, and production prompting for building systems that use AI behind the scenes in apps and tools [4]. For most students, casual prompting — the skill this article focuses on — is already forgiving enough that a little structure goes a long way. As newer “reasoning” models get better at figuring out steps on their own, some older habits, like forcing step-by-step thinking on every question, matter less than they used to [4]. What still matters, and probably always will, is clarity: knowing exactly what you want before you ask for it. Prompt engineering is not about tricking the AI. It is about communicating clearly — a skill that has always mattered.

There is also a growing awareness that prompt engineering overlaps with something bigger: knowing what information, tools, and context to give an AI system in the first place, not just which words to use [4]. For now, though, students do not need to worry about that production-level complexity. Mastering the basics covered in this article already puts you well ahead of the average user.

QUICK FAQ

Is prompt engineering only for programmers?

Not at all. Anyone who types a question into an AI chatbot is already prompting it. The “engineering” part just means doing it a little more deliberately — the same way anyone can learn to write a clearer email or a sharper essay question, no coding required.

Will AI eventually get so smart that this skill becomes useless?

Newer models are getting better at filling in gaps on their own, which is why some older habits are fading [4]. But clearly explaining what you want, for whom, and in what format is a communication skill, not a workaround for a weak AI — it will likely stay useful for a long time.

What is the single fastest way to improve a prompt?

Add one sentence of context the AI would not otherwise know: who the answer is for, what it will be used for, or how long it should be. That one addition usually improves the result more than any other single change.

Final Thought

You do not need a technical degree to benefit from prompt engineering. You need curiosity, a little practice, and the willingness to rewrite a question when the first answer falls short. The next time an AI tool gives you a disappointing response, do not give up on it — try re-asking with more detail, more context, and a clearer goal. That one habit is, in essence, the entire skill.

Glossary: Jargon Explained Simply

LLM: Large Language Model — the type of AI behind ChatGPT, Claude, and Gemini, trained on huge amounts of text.

Token: A small chunk of text (often part of a word) that an AI reads and generates one piece at a time.

Context window: How much text an AI can “remember” and consider at once during a conversation.

Hallucination: When an AI confidently states something that is false or made up.

REFERENCES

- [1] AI Fundamentals. “What Is Prompt Engineering? Techniques & Tips 2026.” aiweekly.co, 2026.
- [2] Rutan, R. “Prompt Engineering.” Launch Rock Library, 27 May 2026.
- [3] “Prompt Engineering Techniques for Developers in 2026.” fungies.io, 2026.
- [4] “Prompt Engineering Guide.” techsy.io, 2026.
- [5] “Prompt Engineering.” Prefactor Glossary, reviewed 9 April 2026.

ARTICLE 07

ARTICLE 07 DEEPFAKES AND THE NEW TRUST PROBLEM

Deepfakes blur reality, making digital content harder to verify and trust.

BY PARTH GAUD
TY BTECH | COMPUTER | B

INTRODUCTION

Imagine receiving a voice message from a teammate asking you to send a project file urgently. The voice sounds familiar. A few minutes later, you discover that the teammate never sent it. Generative AI has made convincing media easier to produce, so recognizing a face or voice can no longer be the only check before we act. The engineering challenge is to establish where a piece of media came from, what changed, and whether a request is genuine [1][2].

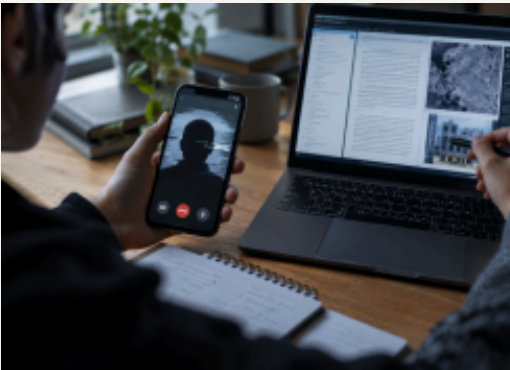


Figure 1. Conceptual workflow for creating an AI-altered image with disclosure.

WHAT IS A DEEPPFAKE?

A deepfake is synthetic or manipulated image, audio, or video that convincingly depicts a person doing or saying something they did not do or say. The term is often used for identity impersonation. It is useful to distinguish that from harmless synthetic media, such as consented dubbing or a clearly labeled fictional scene. The harm depends on context, disclosure, consent, and use [1].

Modern generators learn statistical patterns from examples. A face replacement system may map one person’s appearance onto another video; a voice cloning system can model speech features and produce new audio from text. The result may be edited again, compressed, cropped, or shared through messaging apps. These changes make simplistic visual checks, such as counting fingers or looking for odd blinking, unreliable.

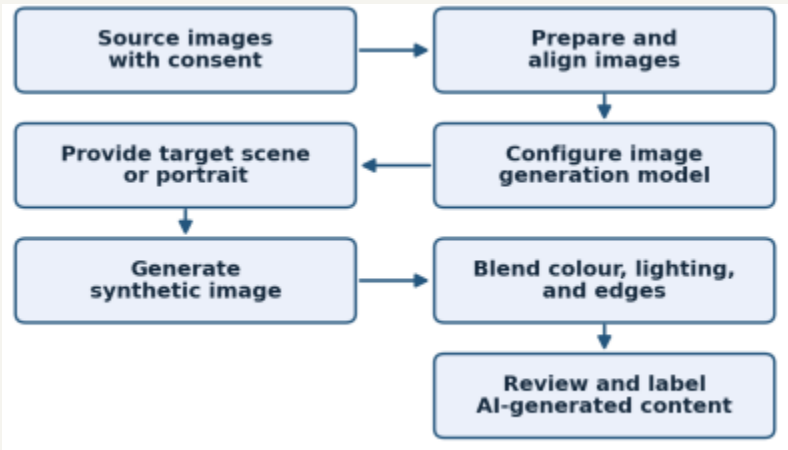


Figure 2. A familiar voice or face should prompt source verification. Illustrative AI-generated image. [1][2]

There is no single recipe behind all deepfakes. Some systems synthesize an entire image from a text prompt. Others replace a face in existing footage, alter lip movements to match a new script, or generate speech that resembles a target speaker. The attacker may combine ordinary editing with AI-generated elements. This matters during investigation: finding a genuine background, a real timestamp, or a familiar setting does not establish that the voice or central action is authentic.

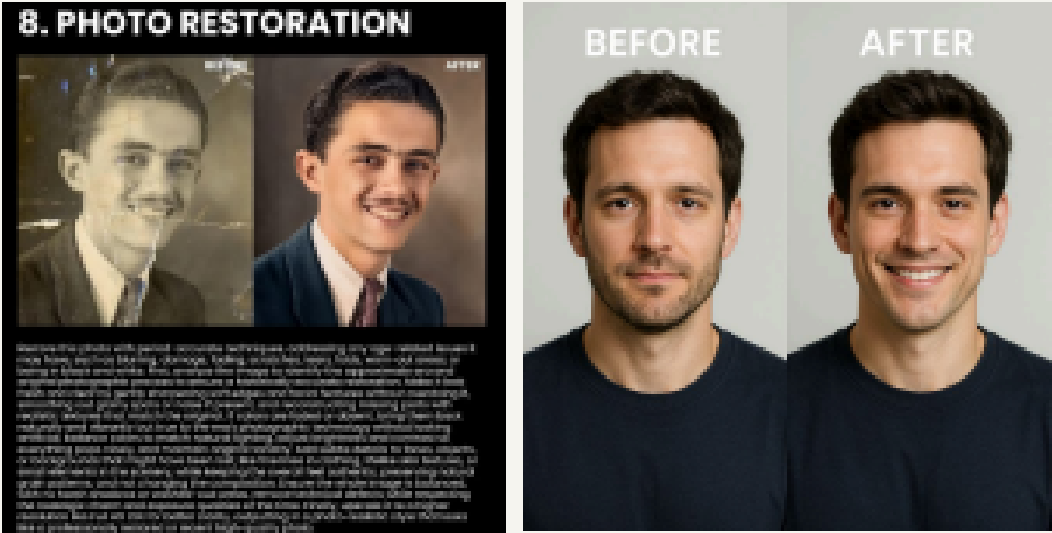


Figure 3. Photo restoration (left) and altered portrait (right). Editing alone does not imply deception. [1]

A clip also loses context as it travels. Reposting may remove its caption; a screenshot may strip embedded metadata; compression may blur telltale details. A viewer who sees only the final social media upload may have no access to the original file. Verification therefore depends on tracing the distribution path and finding corroborating evidence, rather than treating pixel inspection as a complete solution [1][3].

WHY ENGINEERS SHOULD CARE ?

A fake clip can mislead people about a public event, damage someone’s reputation, or support impersonation and financial fraud. The US Federal Trade Commission warns that scammers can clone a familiar voice from a short online sample and use an invented emergency to request money [2]. In a college setting, the same pattern could be used to impersonate a faculty member, recruiter. Those campus examples are plausible scenarios, not reported incidents.

DETECTION AND ITS LIMITS



Figure 4. Media editing and provenance require careful review. Illustrative AI-generated image. [1][3][4]

A PRACTICAL VERIFICATION WORKFLOW

First, pause before forwarding or acting on an urgent clip. Find the original upload and check whether a trusted source published the same event. Next, compare independent reporting, dates, location clues, and the full recording rather than a short excerpt. For a request involving money, credentials, or private files, contact the person through a number or channel you already know; do not use only the contact details in the suspicious message [2].

Detection systems examine signals such as inconsistent image detail, audio artifacts, or patterns learned from labeled examples. Their output is a probability or classification under particular test conditions, not a final verdict. A tool can fail on unfamiliar generation methods or on media that has been heavily edited or compressed. NIST describes detection, watermarking, and provenance as complementary approaches and calls for further testing and research [1].

A watermark can mark content at creation; provenance records can document a file's source and editing history. The C2PA standard defines cryptographically bound Content Credentials for such records. A valid credential can help verify that recorded provenance belongs to a particular asset and has not been tampered with. It does not prove that the scene itself is truthful. Likewise, a missing credential does not by itself prove that a file is fake [3][4].

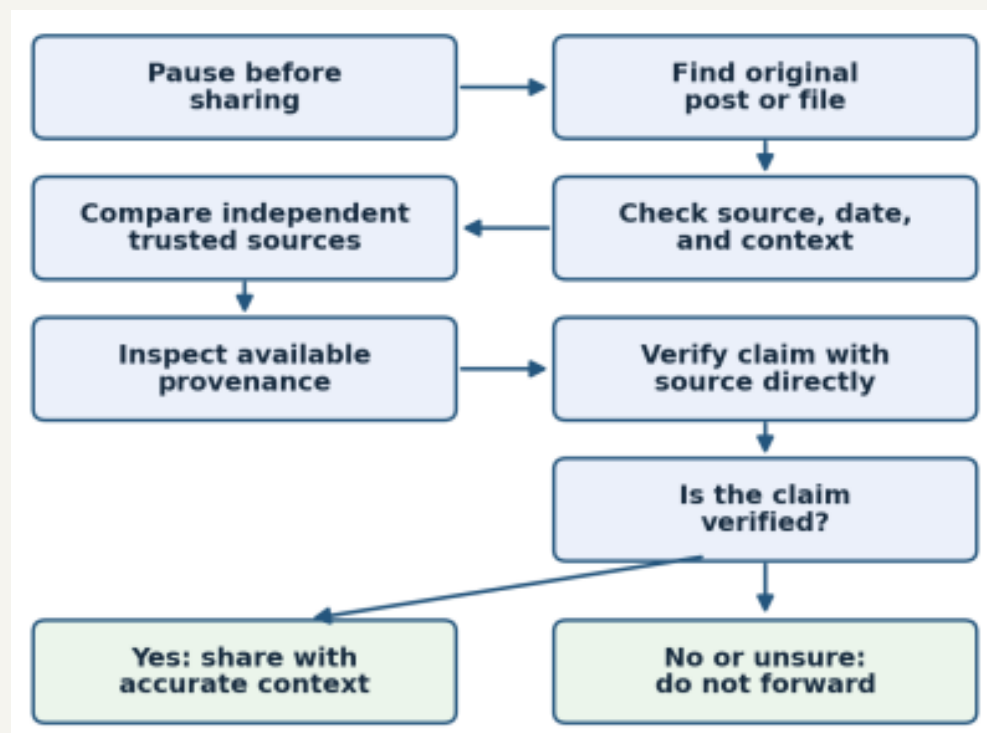


Figure 5. A verification workflow for suspicious images; no single visual clue proves authenticity. [1] [3]

If Content Credentials are available, inspect their source and edit history as one piece of evidence. Use detection tools as supporting signals and record uncertainty honestly. When building systems that process sensitive media, add human review, clear appeal routes, and consent controls. A cautious workflow protects people better than a confident label produced from a single model [1][3].

DESIGNING FOR TRUST

A campus application could make verification part of its normal workflow. Official announcements should appear on a known website or authenticated app, with a consistent record of updates. Requests for payment, account changes, or confidential documents should require confirmation in a second channel. A team can also designate who is allowed to publish official media and keep an archive of original announcements. These controls reduce dependence on a person’s ability to spot a convincing fake.

CONCLUSION

Deepfakes expose a weakness in our online habits: familiar faces and voices can trigger trust before we verify context. Engineers can respond with better provenance tools, careful detection, and workflows that make independent confirmation easy. For everyone else, the most useful habit is simple: verify the source and the request before you share, pay, or provide access.

REFERENCES

- [1] NIST, Reducing Risks Posed by Synthetic Content (2024).
- [2] US Federal Trade Commission, Scammers Use Fake Emergencies To Steal Your Money.
- [3] C2PA, Content Credentials Explainer.
- [4] C2PA, Guiding Principles. [View source](#)

ARTICLE 08

ARTICLE 08 BEYOND CODING THE SKILLS EVERY ENGINEERING STUDENT NEEDS IN 2026

Engineering careers now demand more than coding—critical thinking, communication, creativity, and AI fluency matter too.

BY PRACHI JADHAV
TY BTECH | COMPUTER | B

INTRODUCTION

You can write code that passes every test and still build the wrong thing. Perhaps the team never asked users what they needed, or no one explained a security tradeoff before deployment. Engineering in 2026 asks students to connect technical work with problem definition, communication, teamwork, and responsible decisions. These skills do not replace coding; they make coding useful in real settings [1][2].

START BY DEFINING PROBLEM

Before opening an editor, write down who has the problem, what happens now, and what success would look like. For a campus helpdesk app, “make an AI chatbot” is a solution proposal. “Students wait two days for routine answers and cannot track unresolved requests” is a problem statement that can be tested. Interview users, map the existing process, and choose one measurable outcome, such as resolution time or the fraction of queries answered accurately. Human-centered design likewise begins with the people a solution is intended to serve [3]. Analytical thinking remains one of the core skills employers identify in the World Economic Forum’s Future of Jobs Report 2025. Its survey covers more than 1,000 employers across 55 economies; its findings describe employer expectations rather than a guarantee about any individual student’s career [1]. Good problem definition also involves constraints. Ask what can be built with the available time and data, what a failure would cost, and how performance will be measured. An ETA prediction project, for example, may achieve a low average error while performing poorly during disruptions. Testing only routine trips would hide the case in which passengers need the estimate most.



Figure 1. Career planning and critical thinking are part of engineering growth.

EXPLAIN YOUR WORK CLEARLY



Figure 2. Communication and teamwork help ideas move into practice.

A good engineer can describe a design to a teammate, a mentor, and a nontechnical user without giving all three the same explanation. Practice writing a short project brief with the problem, approach, evidence, limitations, and next step. In a demo, show what the system actually does and state what has only been simulated. ABET's 2025–2026 engineering criteria explicitly include effective communication as a student outcome [2].

Communication also means asking precise questions, keeping a decision log, and documenting assumptions. A concise README that lets another student run the project is often more valuable than an impressive screenshot with no reproducible setup.

WORK WELL IN A TEAM

Hackathons and class projects expose a common failure: parallel work that never integrates. Agree on ownership, interface contracts, deadlines, and a shared definition of done. Use code review to explain reasoning, not just to approve changes. When a teammate finds a defect, focus on reproducing it and fixing the system. ABET's criteria call for effective teamwork that sets goals, plans tasks, and meets objectives [2].



Figure 3. Career planning and critical thinking are part of engineering growth.

USE AI AS A TOOL YOU CAN AUDIT



Figure 4. Examples of AI assistants used in engineering workflows.

AI assistants can help outline an approach, generate a test case, or explain unfamiliar code. Their answers can also be wrong, insecure, or unsuitable for your constraints. Read generated code, run tests, check the cited sources, and make sure you can explain every important design decision. Avoid pasting private data or credentials into a tool without understanding the rules that govern it. The World Economic Forum projects strong growth in the importance of AI and big data skills, alongside creative thinking, resilience, and lifelong learning [1]. UNESCO also emphasizes critical judgement and responsible use in its AI competency framework for students [4].

MAKE ETHICAL AND SECURITY DECISIONS EARLY

If a project ranks job candidates, predicts train arrivals, or processes student complaints, its outputs can affect real people. Ask what data is collected, who may be excluded, how errors are corrected, and who can challenge a decision. Collect only what is needed, protect access, and report known limits. ABET includes ethical and professional responsibilities and the wider impacts of engineering solutions among its student outcomes [2]. NIST identifies reliability, privacy, transparency, and fairness as characteristics to consider when managing AI risks [5]. For web applications, OWASP also lists broken access control among its leading security risks [6].

BUILD EVIDENCE OF THE SKILLS

Choose one project this semester and keep a short record: user interviews, initial assumptions, design alternatives, test results, feedback, and revisions. Present the result in five minutes to someone outside your team, then ask what was unclear. Contribute a small improvement to a shared project and leave a useful review for someone else. This creates concrete examples for a portfolio or interview while improving the project itself.

CONCLUSION

The students who stand out can do more than implement a feature. They can frame a real problem, work with others, explain tradeoffs, check their tools, and take responsibility for the result. Those habits are trainable in ordinary coursework and small projects. Start with the next assignment: understand its user, document one decision, and test whether the solution actually helps.

REFERENCES

- [1] World Economic Forum, The Future of Jobs Report 2025, Skills Outlook.
- [2] ABET, Criteria for Accrediting Engineering Programs, 2025–2026.
- [3] IDEO.org, What Is Human-Centered Design?
- [4] UNESCO, AI Competency Framework for Students (2024).
- [5] NIST, AI Risk Management Framework 1.0 (2023).
- [6] OWASP, Top 10:2025.

ARTICLE 09

ARTICLE 09 RANSOMWARE ATTACKS

Ransomware is malicious software designed to deny access to systems or data, commonly through encryption, while attackers demand payment.

BY ANUJA MUTHAL
TY BTECH | COMPUTER | B

THE DIGITAL THREAT

WHEN A COMPUTER ATTACK BECOMES A HUMAN PROBLEM

Imagine arriving at a hospital and discovering that doctors can no longer access electronic patient records, diagnostic

images cannot be retrieved, scheduled operations must be cancelled, and emergency services have to work with paper records.

This is the reality that ransomware can create. Healthcare organisations are particularly sensitive to these attacks because modern medicine depends heavily on digital infrastructure. Electronic health records, medical imaging, laboratory systems, pharmacy systems, appointment platforms and communication networks are interconnected.

WHY HEALTHCARE IS A HIGH-VALUE TARGET

- Patient medical records
- Personal identification and insurance information
- Prescriptions and medication information
- Diagnostic reports and medical images
- Staff credentials and administrative data
- Research and operational information

More importantly, hospitals cannot simply stop operating while an IT problem is investigated. A hospital still has patients who require treatment.

IMPACT BOX

A ransomware incident can affect three interconnected areas: **CONFIDENTIALITY** — sensitive information may be accessed or stolen; **INTEGRITY** — data may be modified or corrupted; **AVAILABILITY** — systems and information may become inaccessible.

ANATOMY OF A RANSOMWARE ATTACK

FROM INITIAL ACCESS TO SYSTEM LOCKDOWN

A ransomware attack is rarely just the moment when files become encrypted. It is usually the final stage of a broader intrusion. [1]

INITIAL ACCESS → EXECUTION → PERSISTENCE → PRIVILEGE ESCALATION → LATERAL MOVEMENT → DATA COLLECTION / EXFILTRATION → ENCRYPTION → EXTORTION

1. INITIAL ACCESS — Phishing, stolen credentials, compromised remote access, vulnerable internet-facing systems, or third-party compromise. [1]
2. EXECUTION — Malicious code or attacker-controlled commands are executed on a compromised system.
3. PERSISTENCE — Attackers may attempt to maintain access so they can return even if the initial intrusion is discovered.
4. PRIVILEGE ESCALATION — The attacker attempts to obtain higher privileges, potentially gaining administrative control.
5. LATERAL MOVEMENT — The attacker moves from one compromised computer or account to other systems.
6. DATA THEFT — Sensitive information may be copied before encryption, enabling double-extortion tactics.
7. ENCRYPTION — Files or systems are encrypted, preventing normal access.
8. EXTORTION — A ransom demand may be presented to the victim.

SECURITY INSIGHT

Ransomware is not only an encryption problem. Stopping an attack early—before privilege escalation and lateral movement—can dramatically reduce its potential impact.

IMPACT, DETECTION & PREVENTION

THE COST GOES BEYOND THE RANSOM

AREA	POTENTIAL EFFECT
Patient Care	Loss of rapid access to histories, medications and clinical information.
Diagnostics	Disruption to imaging and laboratory workflows.
Emergency Services	Reduced capacity and possible patient diversion.
Administration	Scheduling, billing and communication disruption.

PREVENTION: BUILDING A RANSOMWARE-RESILIENT ORGANIZATION

1. Multi-Factor Authentication — Adds another authentication factor beyond a password. [1]
2. Patch Management — Identify and address critical vulnerabilities promptly. [1]
3. Network Segmentation — Limit unrestricted connectivity between clinical, administrative and internet-facing systems. [1]
4. Backups — Maintain protected, reliable backups and regularly test restoration. [1]
5. Employee Awareness — Train staff to recognize phishing, credential theft and social engineering.
6. Incident Response — Define who declares an incident, how care continues and how systems are safely restored.

RESILIENCE MODEL

PREVENT → DETECT → CONTAIN → CONTINUE CARE → RECOVER → LEARN

CASE STUDY

AZ MONICA HOSPITAL CYBER INCIDENT — BELGIUM, 2026

On 13 January 2026, AZ Monica, a Belgian hospital network operating campuses in Antwerp and Deurne, detected a serious cyber incident. At approximately 6:32 a.m., abnormalities were detected in the hospital's IT environment. The organization responded by shutting down servers as a containment measure. [2] [3]

Later reporting characterized the incident as a ransomware cyberattack, while early reporting described it more generally as a cyberattack and did not publicly establish the initial access method. This distinction is retained here for technical accuracy. [1][2][3]

• THE FIRST LINE OF DEFENCE: CONTAINMENT

The hospital disconnected its infrastructure and shut down servers to limit the possibility of further compromise. The defensive decision created an unavoidable trade-off: containing the cyber threat meant deliberately taking critical digital services offline. [2][3][5]

CONTAINMENT VS. CONTINUITY

Containment: disconnect affected infrastructure to limit further compromise.
Consequence: clinical and administrative systems may become unavailable.
Challenge: continue patient care while investigating and recovering the technology.

• THE CASCADE OF OPERATIONAL FAILURE

01 — Clinical Paralysis — Healthcare professionals lost access to electronic patient information and had to work with significantly reduced digital capabilities.

02 — Diagnostic Blackout — Reports concerning the incident described disruption to imaging-related systems, including CT and MRI workflows.

03 — Emergency Diversion — Emergency capacity was reduced and ambulances/emergency-response teams were redirected to other facilities. [4]

CYBER INCIDENT → IT INFRASTRUCTURE DISRUPTED → CLINICAL SYSTEMS UNAVAILABLE → HOSPITAL CAPACITY REDUCED → PATIENTS REDIRECTED → PRESSURE ON NEIGHBOURING HOSPITALS

RECOVERY, LESSONS & REFERENCES

FROM DIGITAL WORKFLOW TO MANUAL WORKFLOW

With electronic systems unavailable, hospital personnel had to rely on alternative procedures. Tasks normally completed digitally could require paper documentation, manual communication, physical transfer of information, additional verification, and later data entry after systems were restored. [4][5]

RECOVERY STRATEGIES AFTER A RANSOMWARE ATTACK

- 1. Isolate affected systems** — Disconnect compromised devices and network segments to prevent the attack from spreading further.
- 2. Maintain essential services** — Use emergency and manual procedures so critical patient-care activities can continue during the outage.
- 3. Restore from trusted backups** — Recover systems only after backups and restored environments are checked for security.
- 4. Verify systems before reconnecting** — Test applications, accounts and network connections before returning them to normal operation.
- 5. Review and improve** — After recovery, document what happened, identify weaknesses and update security controls and response procedures.

THE FUTURE OF HEALTHCARE CYBERSECURITY

Healthcare is becoming increasingly dependent on connected digital systems, cloud services, medical devices and electronic records. This creates major benefits for patients and staff, but it also increases the number of systems that must be protected. Future cybersecurity efforts will need to focus on continuous monitoring, stronger identity protection, secure medical devices, regular staff training and well-tested recovery plans. Cybersecurity should be treated as an ongoing process rather than a one-time technical activity. The goal is not only to prevent ransomware, but also to make sure that healthcare services can continue safely when a digital system becomes unavailable.

CONCLUSION

Ransomware has evolved from a technical nuisance into a major operational risk. The AZ Monica incident demonstrates how quickly a cyber event can become a healthcare continuity problem. Strong security, segmentation, reliable backups, tested incident-response plans and well-rehearsed downtime procedures help organizations remain resilient when technology fails.

Citation note: Numbered in-text citations [1]–[8] correspond to the sources listed below.

IMPACT BOX— THE HIDDEN COST

The visible impact of ransomware is often the locked computer screen. The less visible impact is the human workload created by the outage: SYSTEM OUTAGE → MANUAL PROCEDURES → ADDITIONAL STAFF WORK → SLOWER INFORMATION FLOW → RECOVERY + DATA RECONCILIATION.

REFERENCES

[1] Resillion, “The AZ Monica Hospital Cyber Attack and Belgian Threat Healthcare,” Cyber Incident Report, 2026.

[2] ThaiCERT, “AZ Monica Hospital in Belgium Shuts Down Servers Following Cyberattack,” 15 January 2026.

[3] BleepingComputer, “Belgian hospital AZ Monica shuts down servers after cyberattack,” 13 January 2026.

[4] The Register, “Cyber-stricken Belgian hospitals refuse ambulances, transfer critical patients,” 14 January 2026.

[5] SANS Institute, “AZ Monica Hospitals in Belgium Recovering from Cyber Incident,” January 2026.

[6] Citymesh, “AZ Monica joins forces with Citymesh for rapid recovery after cyberattack,” 24 March 2026.

[7] ICTMagazine, “AZ Monica restarts electronic patient record,” February 2026.

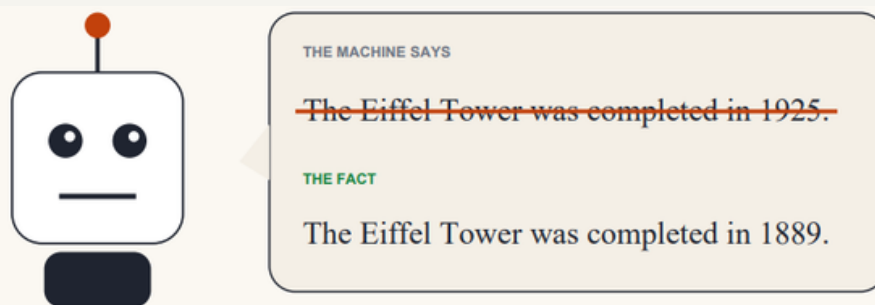
[8] Security.NL, “Systems Belgian hospital restored more than a month after ransomware attack,” February 2026.

ARTICLE 10

ARTICLE 10 WHEN THE MACHINE MAKES THINGS UP

A plain-spoken guide to AI hallucinations: why they happen, what they look like, and how to live with them.

BY TEJAS MORE
TY BTECH | COMPUTER | B



Look at the sentence in the speech bubble. It is fluent, confident and specific, and it is wrong. The correct year is 1889. Nothing in the wording gave the mistake away, and that is exactly the problem this article is about.

Inside: what a hallucination actually is, five reasons it happens, a field guide to the common kinds, three real cases with real consequences, what is being done about it, and a habit or two that will keep you out of trouble.

SO WHAT IS HALLUCINATION?

When an AI system produces something that sounds right but is false, or that is not supported by the material it was given, researchers call it a hallucination.[1] The word is borrowed from psychology, and I think it is slightly misleading. A person who hallucinates perceives something that is not there. A language model is doing something more mundane. It is producing the next most fitting piece of text, again and again, and sometimes the fitting piece is not a true one.

That distinction matters. A chatbot is not lying, because lying needs an intention to deceive, and there is no evidence of one here. It is also not confused in the way you or I get confused. It has no moment where it feels unsure and pushes on anyway. It simply generates text, and text that is fluent is not the same thing as text that is correct.



Figure 1. A language model ranks possible next words by how well they fit. A wrong answer can fit almost as well as the right one.

Look at the picture. Ask for the capital of Australia and the right word fits well, but so does a famous city that people often get wrong, because it appears constantly in the writing the model learned from. Now imagine the question is about an obscure court case, a minor historian or a niche software library. The right answer may barely appear in the training material at all, yet the model will still produce something that fits the shape of an answer.

That is the uncomfortable core of it. The confident tone you read in a hallucination is not evidence of anything. It is simply how the machine writes.

FIVE REASONS IT HAPPENS

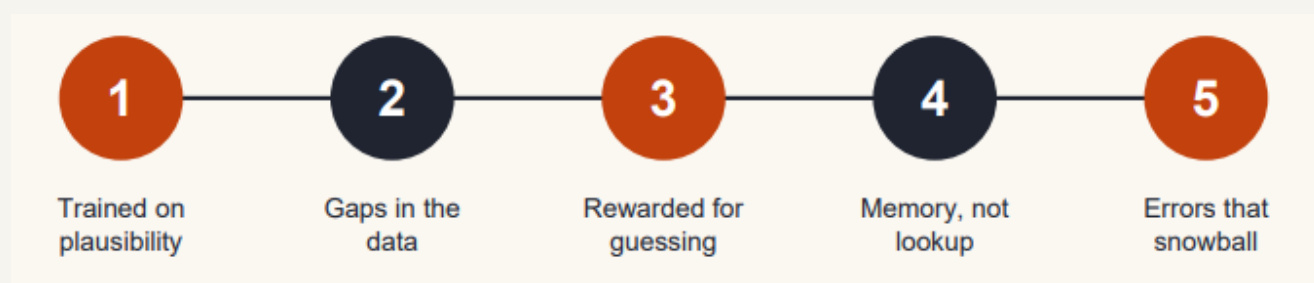


Figure 2. Five overlapping causes, discussed in the research literature.

1. It is trained to sound plausible. The basic training task is predicting what text comes next. Nothing in that task requires a claim to be true, only that it looks like something a human might write.[1]

2. The data has holes and errors. Models learn from enormous piles of text collected from many places. That text contains mistakes, outdated claims and contradictions, and on rare topics there may be very little of it. The model absorbs the errors along with everything else.[3]

3. Guessing gets rewarded. A recent line of research argues that the way models are tested and scored often gives credit for a lucky guess but none for saying "I don't know." A system tuned against those scores learns that bluffing pays better than honesty.[4] I find this the most persuasive explanation for why the problem has been so stubborn.

4. It works from memory, not from a lookup. A model does not keep a tidy library it can consult. What it learned is squeezed into numerical weights, and details such as exact dates, page numbers and names are the first things to blur. Ask for a precise detail and you may get a reconstruction rather than a recollection.

5. Mistakes snowball. Once a model has written a wrong statement, it tends to treat that statement as settled and builds on it. Researchers have shown that models can go on to make further errors they would not otherwise have made, simply to stay consistent with an earlier slip.[5] Leading questions push in the same direction, since a prompt that assumes something false invites the model to play along.

A FIELD GUIDE TO THE COMMON KINDS

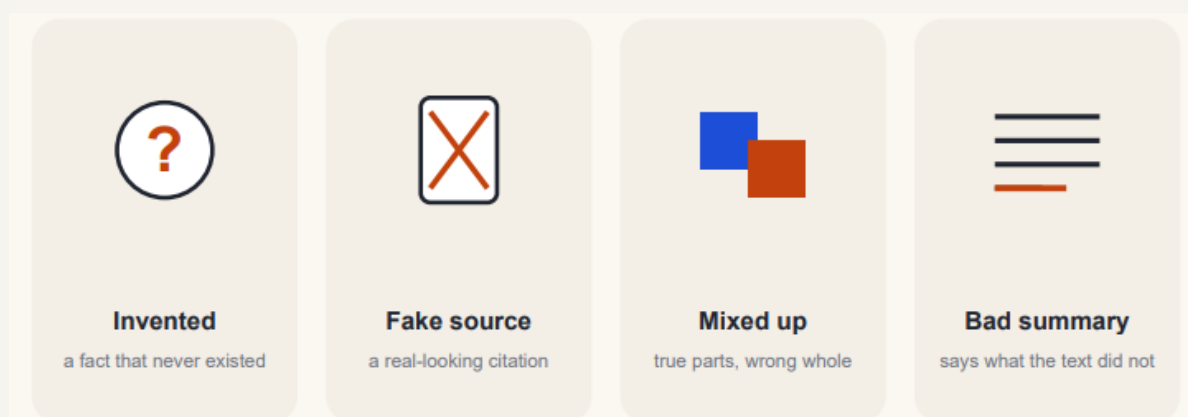


Figure 3. Four kinds of hallucination you are likely to meet.

The invented fact. The most familiar type. The system states a date, a statistic or a biographical detail that simply is not true, in the same calm voice it uses for everything else.

The fake source. Ask for references and you may receive titles, authors and journals that look entirely believable but do not exist. This one catches careful people out, because a citation has all the visual markers of credibility. The only defence is to look the source up.

The mixed-up truth. Here the pieces are real but they have been joined wrongly. A quote from one person gets attached to another, or two similar events blur into one. It is the hardest kind to spot, since a quick search will confirm every individual part.

The unfaithful summary. You paste in a document and ask for a summary, and the answer includes a point the document never made. Researchers treat this as its own category, because here the model had the correct material in front of it and still drifted from it.[6]

What ties these together is that none of them looks like an error. They read as smoothly as the correct answers around them. That is why the fix cannot be to trust your instincts about tone. It has to be a habit of checking.

FLUENT IS NOT THE SAME AS CORRECT.

WHEN IT LEFT THE LAB

Hallucinations would be a curiosity if they only appeared in demos. They have not stayed there. Three well documented cases show what the consequences look like.

The telescope answer

In February 2023, Google promoted its Bard chatbot with an example in which it was asked about discoveries from the James Webb Space Telescope. The answer claimed the telescope had taken the first pictures of a planet outside our solar system. Astronomers quickly pointed out that this was not correct, and the error drew wide coverage on the day of the launch.[7] A single wrong sentence in an advertisement was enough to embarrass one of the largest technology companies in the world.

The lawyers and the phantom cases

In the case of *Mata v. Avianca*, lawyers filed a brief that cited court decisions which did not exist. They had used ChatGPT to research it, and the tool had produced convincing but invented cases. When the court and opposing counsel could not find them, the judge sanctioned the lawyers and their firm and ordered a monetary penalty.[8] The lesson was not that the technology is useless. It was that a professional remains responsible for what they file.

The chatbot that made a promise

A customer of Air Canada asked the airline's website chatbot about bereavement fares and was given an inaccurate description of the policy. When the airline refused to honour it, a Canadian tribunal ruled in the customer's favour. The airline had argued that the chatbot was effectively a separate entity responsible for its own words, and the tribunal rejected that argument.[9]

A company answers for what its own website says, whether a person or a

program wrote it. Each case has the same shape. A confident answer was accepted without a check, and someone paid for it.

WHAT IS BEING DONE ABOUT IT

Nobody has eliminated hallucinations, and honest researchers say so. But there are real ways to reduce them, and a few of them matter to anyone who uses these tools.

Give the model the source. One widely used approach is retrieval-augmented generation. Instead of asking the model to remember, the system first fetches relevant documents and asks the model to answer from them.[10] It narrows the room for invention, though it does not close it, because a model can still misread or embellish what it retrieves.

Change what gets rewarded. If bluffing is rewarded, models bluff. The argument in recent research is that evaluations should stop punishing an honest "I don't know" and should penalise confident errors more than they penalise abstaining.[4] That is a fix aimed at the incentives, not just the symptoms.

Make checking part of the design. Better products show their sources, let you click through to the passage a claim came from, and flag when they are unsure. Surveys of the field treat detection and mitigation as separate, ongoing problems for exactly this reason. [3]

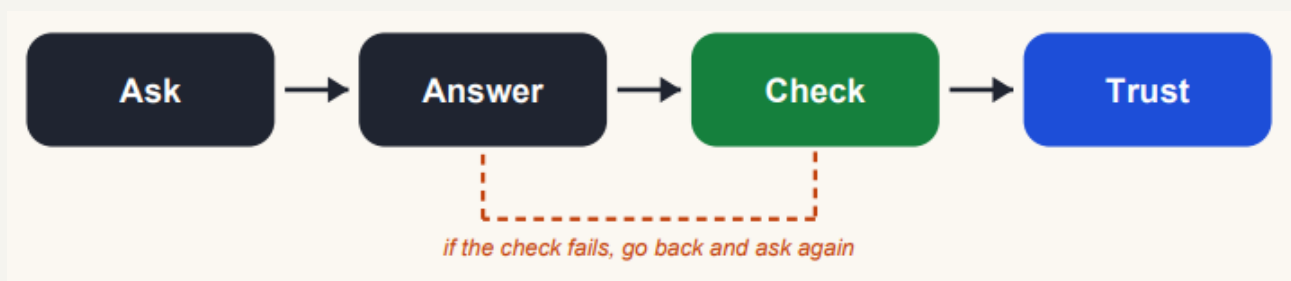


Figure 4. The loop that actually protects you: check before you trust, and go back if the check fails.

Notice that the most reliable safeguard in the figure is not in the software at all. It is the step in the middle, and it belongs to you.

USING IT WITHOUT GETTING BURNED

None of this is an argument for avoiding AI. It is an argument for using it the way you would use a well-read colleague who occasionally bluffs. Match the amount of checking to the cost of being wrong. A gift idea can be taken at face value. A legal filing, a medical decision or a published claim cannot.

Be most suspicious of the things that are easy to fake and easy to check: names, dates, numbers, quotations and citations. Open the source. If you cannot find it, treat it as not real. Give the model your own documents when accuracy matters, and tell it it may say it does not know. And avoid leading questions, since a false assumption in your prompt is an invitation to agree with it.

Above all, keep a person accountable. The tribunal and the court in the cases above both landed on the same idea, and I think it is the right one: the tool does not carry responsibility, so you do.

THE TOOL DOES NOT CARRY RESPONSIBILITY. YOU DO.

REFERENCES

- [1] Ji, Z. et al. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12). [arXiv:2202.03629](#).
- [2] Maynez, J. et al. (2020). On Faithfulness and Factuality in Abstractive Summarization. *Proceedings of ACL 2020*.
- [3] Huang, L. et al. (2023). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. [arXiv:2311.05232](#).
- [4] Kalai, A. T., Nachum, O., Vempala, S. S. and Zhang, E. (2025). Why Language Models Hallucinate. [arXiv:2509.04664](#).
- [5] Zhang, M. et al. (2023). How Language Model Hallucinations Can Snowball. [arXiv:2305.13534](#).
- [6] Maynez et al. (2020), as above, for the distinction between faithfulness to the source and factual accuracy.

- [7] Reuters (February 2023). Coverage of Google's Bard error in its launch advertisement about the James Webb Space Telescope.
- [8] Mata v. Avianca, Inc., No. 22-cv-1461 (S.D.N.Y. 2023), opinion and order on sanctions.
- [9] Moffatt v. Air Canada, 2024 BCCRT 149 (Civil Resolution Tribunal of British Columbia).
- [10] Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 2020. arXiv:2005.11401.