



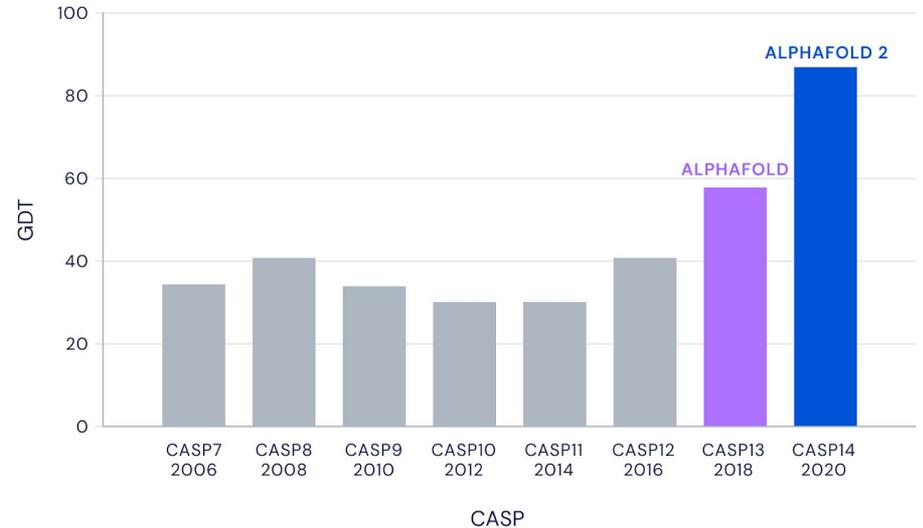
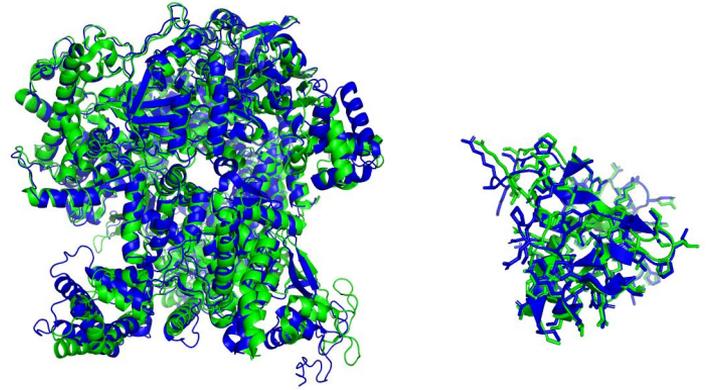
# Investigación e industria Una alianza necesaria

Bernardino Romera Paredes

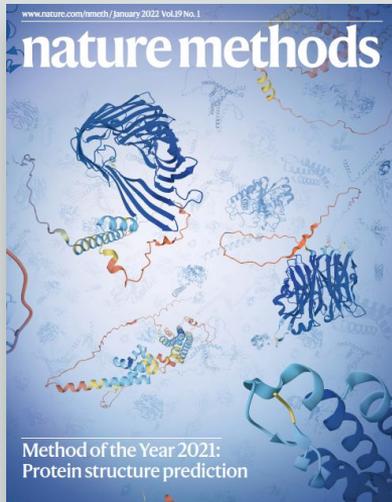
REDIT SUMMIT

# AlphaFold

- Una proteína es una cadena (molécula larga) de amino ácidos
- Las proteínas se pliegan formando una estructura tridimensional.
- La figura que forman determina en muchas ocasiones su función → Entender las estructuras de las proteínas es un problema fundamental en biología



# El Impacto de AlphaFold



## Breakthrough of the Year 2021, Science

## Method of the Year 2021, Nature

Methods



N I U S



PORTADA | CIENCIA

Noelia Ferruz, la científica española que diseña proteínas “revolucionarias” mediante Inteligencia Artificial



Irene Fernández · Madrid

04/10/2022 · 01:00h.

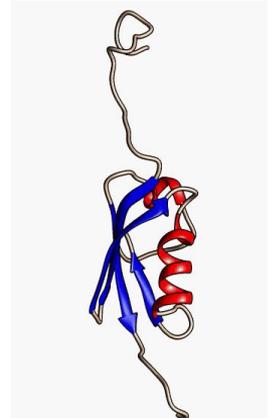
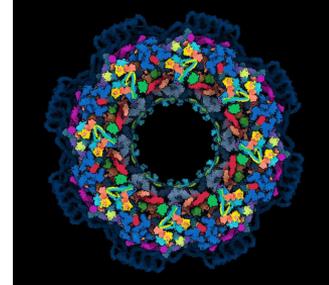


La investigadora lleva años trabajando en el diseño de proteínas con IA. NOELIA FERRUZ / NIUS

- Ferruz explica los avances que supone AlphaFold: “Es una herramienta brutal: nos ha cambiado la forma de trabajar”

# Ejemplos donde AlphaFold ya está acelerando la ciencia

- La Iniciativa [Drugs for Neglected Diseases Initiative](#) usó AlphaFold para avanzar su investigación sobre curas de enfermedades mortales que afectan desproporcionadamente a las partes más pobres del planeta (ej. leishmaniasis).
- El [Centre for Enzyme Innovation](#) están usando las predicciones de AlphaFold para ayudar a crear enzimas más rápidas para reciclar algunos de los plásticos de un solo uso más contaminantes.
- Descubrimiento de medicamentos: Insilico Medicine [describe](#) cómo AlphaFold aceleró la investigación para encontrar un medicamento contra el cáncer de hígado.
- Científicos del EMBL y el MPIBP usaron AlphaFold para [modelar y entender el complejo poro nuclear humano](#), la máquina molecular más grande en células humanas que los científicos han intentado resolver por más de una década.
- Científicos del Institute of Cancer Research y el EMBL demostraron que AlphaFold mejora el estado del arte en [predicción de desorden](#) en proteínas, considerado por los autores de ese estudio como una revolución.
- Científicos de varias universidades danesas usaron AlphaFold para [modelar la estructura de proteínas bacteriales](#) que promueven la formación de hielo. Esto podría contribuir al efecto enfriamiento del hielo en las nubes.
- Un equipo de la Universidad de California San Francisco [ha usado AlphaFold](#) para revelar aspectos clave en el funcionamiento del SARS-CoV-2.
- Numerosas historias de investigadores que durante años han estado intentando obtener la estructura de una proteína, y finalmente lo han logrado gracias a AlphaFold utilizando un proceso llamado reemplazamiento molecular.



Descubrir nuevos **algoritmos** verificablemente correctos que superen a los actuales en problemas impactantes

# Multiplicación de matrices en el mundo real



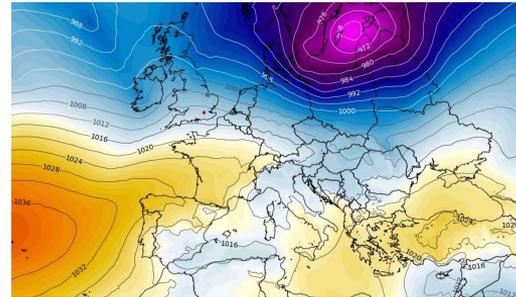
Móviles



Generación de gráficos para videojuegos



Reconocimiento del habla



Simulaciones y predicción del tiempo



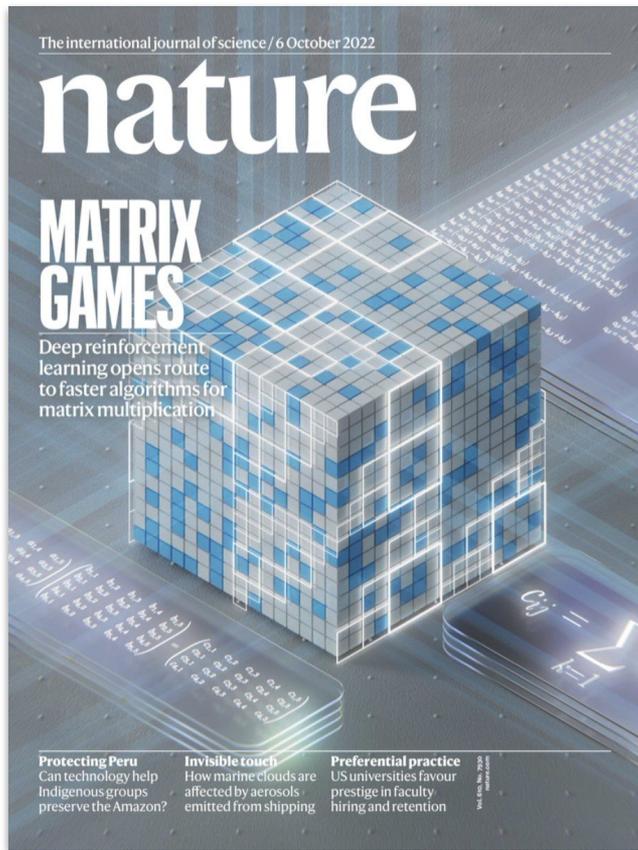
# AlphaTensor

The international journal of science / 6 October 2022

# nature

## MATRIX GAMES

Deep reinforcement learning opens route to faster algorithms for matrix multiplication



**Protecting Peru**  
Can technology help indigenous groups preserve the Amazon?

**Invisible touch**  
How marine clouds are affected by aerosols emitted from shipping

**Prestigious practice**  
US universities favour hiring and retention

ISSN 0028-0836

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{pmatrix}$$

$$h_1 = a_{1,1}b_{1,3}$$

$$h_2 = (a_{1,1} + a_{3,1} + a_{3,3})(b_{1,1} + b_{3,1} + b_{3,3})$$

$$h_3 = (a_{1,1} + a_{3,1} + a_{3,4})(b_{1,2} + b_{4,2} + b_{4,3})$$

$$h_4 = (a_{1,3} + a_{2,1} + a_{2,3})(b_{1,3} + b_{1,4} + b_{3,4})$$

$$h_5 = (a_{1,1} + a_{3,1})(b_{1,1} + b_{1,2} + b_{1,3} + b_{3,1} + b_{3,3} + b_{4,2} + b_{4,3})$$

$$h_6 = (a_{1,3} + a_{2,3})(b_{1,3} + b_{1,4} + b_{3,2} + b_{3,3} + b_{3,4} + b_{4,2} + b_{4,3})$$

$$h_7 = (a_{1,4} + a_{4,3} + a_{4,4})(b_{3,1} + b_{3,3} + b_{4,1})$$

$$h_8 = (a_{1,4} + a_{4,1} + a_{4,4})(b_{1,3} + b_{1,4} + b_{4,4})$$

$$h_9 = (a_{1,3} + a_{2,3} + a_{2,4})(b_{3,2} + b_{4,2} + b_{4,3})$$

$$h_{10} = (a_{1,4} + a_{4,4})(b_{1,3} + b_{1,4} + b_{3,1} + b_{3,3} + b_{4,1} + b_{4,3} + b_{4,4})$$

$$h_{11} = a_{3,3}(b_{1,1} + b_{2,2} + b_{2,3} + b_{3,1} + b_{3,2})$$

$$h_{12} = (a_{1,2} + a_{3,2} + a_{3,3})(b_{2,2} + b_{2,3} + b_{3,2})$$

$$h_{13} = a_{3,4}(b_{1,2} + b_{2,1} + b_{2,3} + b_{4,1} + b_{4,2})$$

$$h_{14} = (a_{1,2} + a_{3,2})(b_{2,1} + b_{2,2} + b_{2,3} + b_{3,2} + b_{4,1})$$

$$h_{15} = (a_{1,2} + a_{3,2} + a_{3,4})(b_{2,1} + b_{2,3} + b_{4,1})$$

$$h_{16} = a_{2,1}(b_{1,2} + b_{1,4} + b_{2,2} + b_{2,3} + b_{3,4})$$

$$h_{17} = h_{15} + h_{26} + h_2 + h_{30} + h_{32} + h_{39} + h_{40} + h_{42} + h_{45} + h_{47}$$

$$h_{18} = (a_{1,2} + a_{2,2})(b_{1,2} + b_{2,2} + b_{2,3} + b_{2,4} + b_{4,4})$$

$$h_{19} = a_{2,4}(b_{2,3} + b_{2,4} + b_{3,2} + b_{4,2} + b_{4,4})$$

$$h_{20} = (a_{1,2} + a_{2,3} + a_{2,4} + a_{3,2} + a_{3,3})b_{3,2}$$

$$h_{21} = (a_{1,2} + a_{2,2} + a_{2,4})(b_{2,3} + b_{2,4} + b_{4,4})$$

$$h_{22} = a_{4,3}(b_{2,3} + b_{2,4} + b_{3,1} + b_{3,4} + b_{4,1})$$

$$h_{23} = (a_{1,1} + a_{1,3} + a_{1,4} + a_{2,3} + a_{2,4} + a_{3,1} + a_{3,4})(b_{4,2} + b_{4,3})$$

$$h_{24} = (a_{1,2} + a_{4,2} + a_{4,3})(b_{2,3} + b_{2,4} + b_{3,4})$$

$$h_{25} = (a_{1,2} + a_{4,2})(b_{1,1} + b_{2,1} + b_{2,3} + b_{2,4} + b_{3,4})$$

$$h_{26} = (a_{1,2} + a_{4,1} + a_{4,2})(b_{1,1} + b_{2,1} + b_{2,3})$$

$$h_{27} = a_{1,4}b_{4,3}$$

$$h_{28} = (a_{1,2} + a_{2,1} + a_{2,2} + a_{3,1} + a_{3,4})b_{1,2}$$

$$h_{29} = (a_{1,2} + a_{2,1} + a_{2,3} + a_{4,2} + a_{4,3})b_{3,4}$$

$$h_{30} = (a_{1,2} + a_{3,1} + a_{3,3} + a_{4,1} + a_{4,2})b_{1,1}$$

$$h_{31} = a_{4,1}(b_{1,1} + b_{1,4} + b_{2,1} + b_{2,3} + b_{4,4})$$

$$h_{32} = (a_{1,2} + a_{3,2} + a_{3,4} + a_{4,3} + a_{4,4})b_{4,1}$$

$$h_{33} = (a_{1,2} + a_{2,2} + a_{2,4} + a_{4,1} + a_{4,4})b_{4,4}$$

$$h_{34} = (a_{2,1} + a_{3,1} + a_{4,1})(b_{1,1} + b_{1,2} + b_{1,4})$$

$$h_{35} = (a_{1,2} + a_{2,1} + a_{2,2} + a_{3,2} + a_{3,3})(b_{2,2} + b_{2,3})$$

$$h_{36} = (a_{1,2} + a_{2,4} + a_{3,2} + a_{4,3})(b_{2,3} + b_{2,4} + b_{3,2} + b_{4,1})$$

$$h_{37} = (a_{1,2} + a_{2,1} + a_{3,3} + a_{4,2})(b_{1,1} + b_{2,2} + b_{2,3} + b_{3,4})$$

$$h_{38} = (a_{2,2} + a_{3,2} + a_{4,2})(b_{2,1} + b_{2,2} + b_{2,4})$$

$$h_{39} = a_{1,3}b_{2,3}$$

$$h_{40} = a_{1,3}b_{3,3}$$

$$h_{41} = (a_{1,1} + a_{1,3} + a_{1,4} + a_{2,1} + a_{2,3} + a_{4,1} + a_{4,4})(b_{1,3} + b_{1,4})$$

$$h_{42} = (a_{1,2} + a_{3,2} + a_{3,4} + a_{4,1} + a_{4,2})(b_{2,1} + b_{2,3})$$

$$h_{43} = (a_{2,4} + a_{3,4} + a_{4,4})(b_{4,1} + b_{4,2} + b_{4,4})$$

$$h_{44} = (a_{2,3} + a_{3,3} + a_{4,3})(b_{3,1} + b_{3,2} + b_{3,4})$$

$$h_{45} = (a_{1,1} + a_{1,3} + a_{1,4} + a_{3,1} + a_{3,3} + a_{4,3} + a_{4,4})(b_{3,1} + b_{3,3})$$

$$h_{46} = (a_{1,2} + a_{2,2} + a_{3,4} + a_{4,1})(b_{1,2} + b_{2,1} + b_{2,3} + b_{4,4})$$

$$h_{47} = (a_{1,2} + a_{2,2} + a_{2,4} + a_{4,2} + a_{4,3})(b_{2,3} + b_{2,4})$$

$$c_{1,1} = h_{15} + h_{26} + h_2 + h_{30} + h_{32} + h_{39} + h_{40} + h_{42} + h_{45} + h_{47}$$

$$c_{2,1} = h_{11} + h_{12} + h_{14} + h_{20} + h_{22} + h_{24} + h_{25} + h_{29} + h_{35} + h_{36} + h_{37} + h_{38} + h_{44} + h_{47}$$

$$c_{3,1} = h_{11} + h_{12} + h_{14} + h_{15} + h_{26} + h_{30} + h_{39} + h_{42}$$

$$c_{4,1} = h_{15} + h_{22} + h_{24} + h_{25} + h_{26} + h_{32} + h_{39} + h_{42}$$

$$c_{1,2} = h_{12} + h_{17} + h_{20} + h_{23} + h_{28} + h_{35} + h_{39} + h_3 + h_{19}$$

$$c_{2,2} = h_{12} + h_{17} + h_{18} + h_{19} + h_{20} + h_{21} + h_{35} + h_{39}$$

$$c_{3,2} = h_{12} + h_{13} + h_{14} + h_{15} + h_{17} + h_{28} + h_{35} + h_{39}$$

$$c_{4,2} = h_{13} + h_{14} + h_{15} + h_{18} + h_{19} + h_{21} + h_{32} + h_{33} + h_{36} + h_{38} + h_{42} + h_{43} + h_{46} + h_{47}$$

$$c_{1,3} = h_1 + h_{27} + h_{39} + h_{40}$$

$$c_{2,3} = h_{16} + h_{17} + h_{18} + h_{19} + h_{21} + h_{39} + h_{40} + h_4 + h_6 + h_9$$

$$c_{3,3} = h_{11} + h_{12} + h_{13} + h_{14} + h_{15} + h_1 + h_2 + h_{39} + h_3 + h_5$$

$$c_{4,3} = h_{10} + h_{22} + h_{24} + h_{25} + h_{26} + h_{27} + h_{31} + h_{39} + h_7 + h_8$$

$$c_{1,4} = h_1 + h_{21} + h_{24} + h_{29} + h_{33} + h_{39} + h_{41} + h_{47} + h_4 + h_8$$

$$c_{2,4} = h_{16} + h_{17} + h_{18} + h_{21} + h_{24} + h_{29} + h_{39} + h_{47}$$

$$c_{3,4} = h_{16} + h_{17} + h_{18} + h_{25} + h_{26} + h_{28} + h_{30} + h_{31} + h_{34} + h_{35} + h_{37} + h_{38} + h_{42} + h_{46}$$

$$c_{4,4} = h_{21} + h_{24} + h_{25} + h_{26} + h_{31} + h_{33} + h_{39} + h_{47}$$

# Problemas del aprendizaje por refuerzo

- Estos sistemas tienen que aprender desde 0 → requieren una enorme cantidad de recursos
- No son flexibles: problemas distintos requieren arquitecturas distintas
- La solución es una red neuronal → no son interpretables ni verificables, y difíciles de implementar
- El código de estos sistemas basados en RL es muy difícil de mantener!

# LLMs

Si le preguntas a un LLM que proporcione nuevas soluciones a un problema:

- Alucinará una solución
- O versionará algún trozo de código de su training set que tenga alguna relación con el problema

## Los grandes modelos de lenguaje de inteligencia artificial, cada vez menos fiables

Según un estudio de la Universitat Politècnica de València, ValgrAI y la Universidad de Cambridge, que publica la revista Nature

[ 26/09/2024 ]

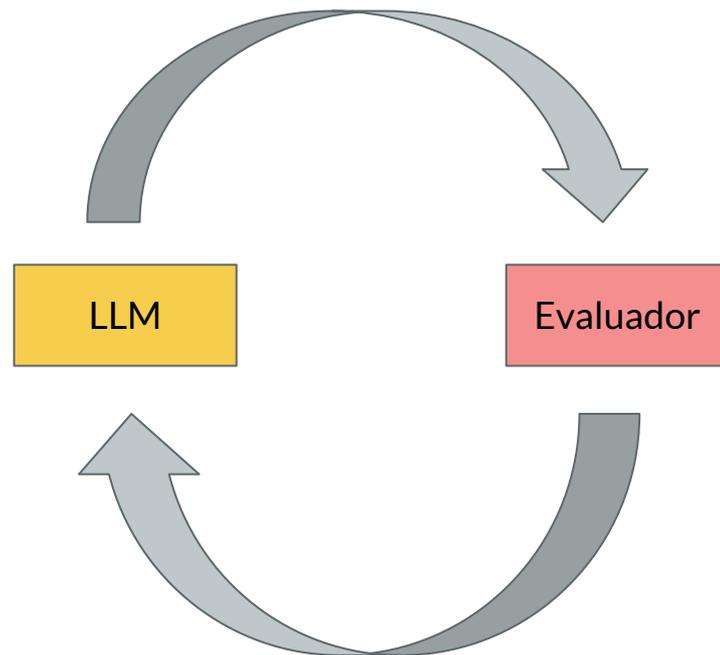


# LLMs para descubrir algoritmos

Como trabajamos con código, podemos combinar:

- un LLM preentrenado, cuyo objetivo es proporcionar soluciones **creativas** en forma de código informático
- con un "evaluador" automatizado, que elimina alucinaciones e ideas incorrectas.

Al iterar de entre estos dos componentes, las soluciones iniciales "**evolucionan**" hacia nuevos conocimientos.



# LLMs para descubrir algoritmos



nature

Explore content ▾

About the journal ▾

Publish with us ▾

[nature](#) > [articles](#) > article

Article | [Open access](#) | Published: 14 December 2023

## Mathematical discoveries from program search with large language models

[Bernardino Romera-Paredes](#) , [Mohammadamin Barekatin](#), [Alexander Novikov](#), [Matej Balog](#), [M. Pawan Kumar](#), [Emilien Dupont](#), [Francisco J. R. Ruiz](#), [Jordan S. Ellenberg](#), [Pengming Wang](#), [Omar Fawzi](#), [Pushmeet Kohli](#)  & [Alhussein Fawzi](#) 

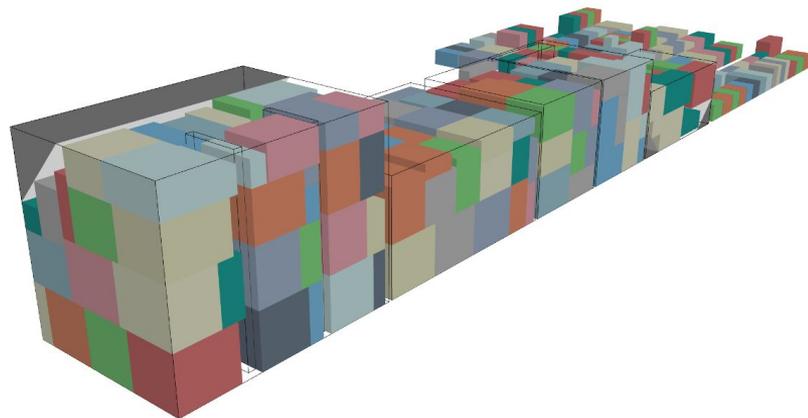
*Nature* **625**, 468–475 (2024) | [Cite this article](#)

# Problema del empaquetamiento online

Cuál es la forma más eficiente de empaquetar objetos de distintos tamaños en cajas?

Es clave en muchos problemas del mundo real, e.j:

- cargar contenedores con cajas
- asignar trabajos computacionales a servidores en centros de datos para minimizar costos.



# FunSearch se adapta a cada caso de uso

Funsearch puede **automáticamente** producir programas / estrategias **adaptadas** a un caso de uso específico

A diferencia de soluciones basadas en redes neuronales, lo que produce FunSearch es código:

- Es más fácil de implementar (no necesitas hardware especializado, como GPUs)
- No es una caja negra: es más fácil de debuggear
- Es más predecible
- Es más escalable

# Casos de uso

Toma de decisiones en tiempo real, como:

- Optimizar un centro de datos
- Asignación de taxis a clientes
- Automatización de líneas de ensamblaje

Heurísticas de búsqueda, útiles en tareas como:

- Transporte y logística
- Diseño de circuitos electrónicos
- Optimización de la red eléctrica



# Conclusión

- Muchas oportunidades a nivel de investigación para llevar a la IA al siguiente nivel
- Enormes posibilidades para crear valor en casos de uso reales usando IA