UNIVERSITY OF LIVERPOOL

Department of Physics

INTRODUCTION TO AI IN PARTICLE ACCELERATORS

Dr. Andrea Santamaria Garcia
Lecturer

CI lectures CI-ACC-226
Lecture on 23/02/2026

OUTLINE

- **Artificial intelligence, machine learning…what are they?** (12 slides)
  - *AI & ML, learning paradigms, game: is it ML?, optimisation paradigms*

- **Motivation: why AI for particle accelerators (now & future)?** (18 slides)

- **Historical context: the birth and evolution of AI** (15 slides)
  - *AI history in a nutshell, why exactly is AI booming now?, new buzzwords*

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING…WHAT ARE THEY?

GenAI

Algorithms that learn the underlying distribution of data to generate new samples resembling the training data

text generation, image synthesis, code generation, audio generation, data augmentation

Deep Learning (DL)

Large neural networks that learn hierarchical representations from raw data

speech recognition, natural language processing, machine translation, recommender systems, computer vision, generative modelling

Neural Networks (NNs)

A family of parameterised function approximators of interconnected neurons that learn complex, often non-linear mappings from data through gradient-based optimisation

MLP, CNN, RNN, GNN, AE, VAE, GAN, PINN

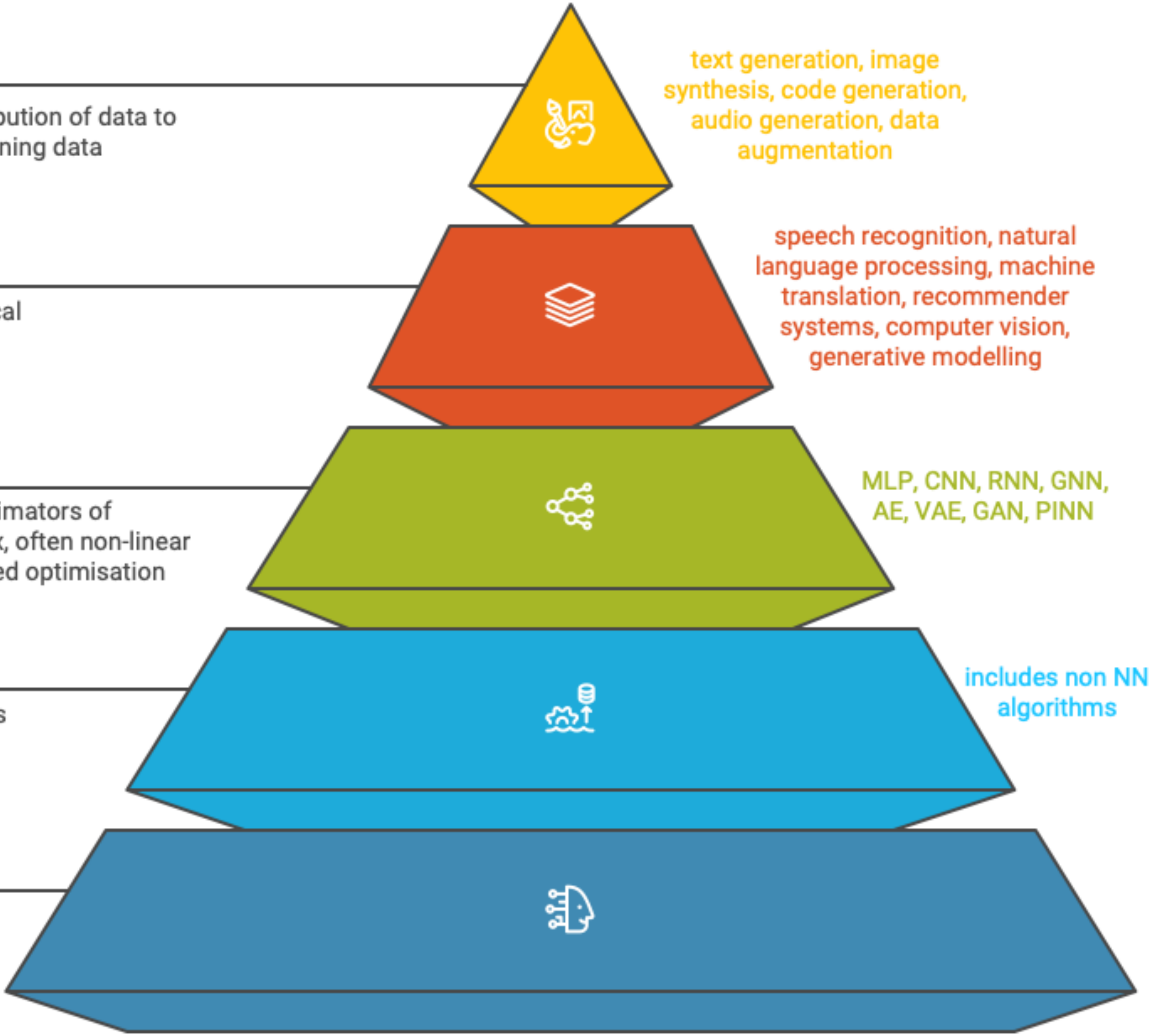
Machine Learning (ML)

Algorithms learning patterns or parameters from data enabling them to generalise to unseen inputs and improve performance

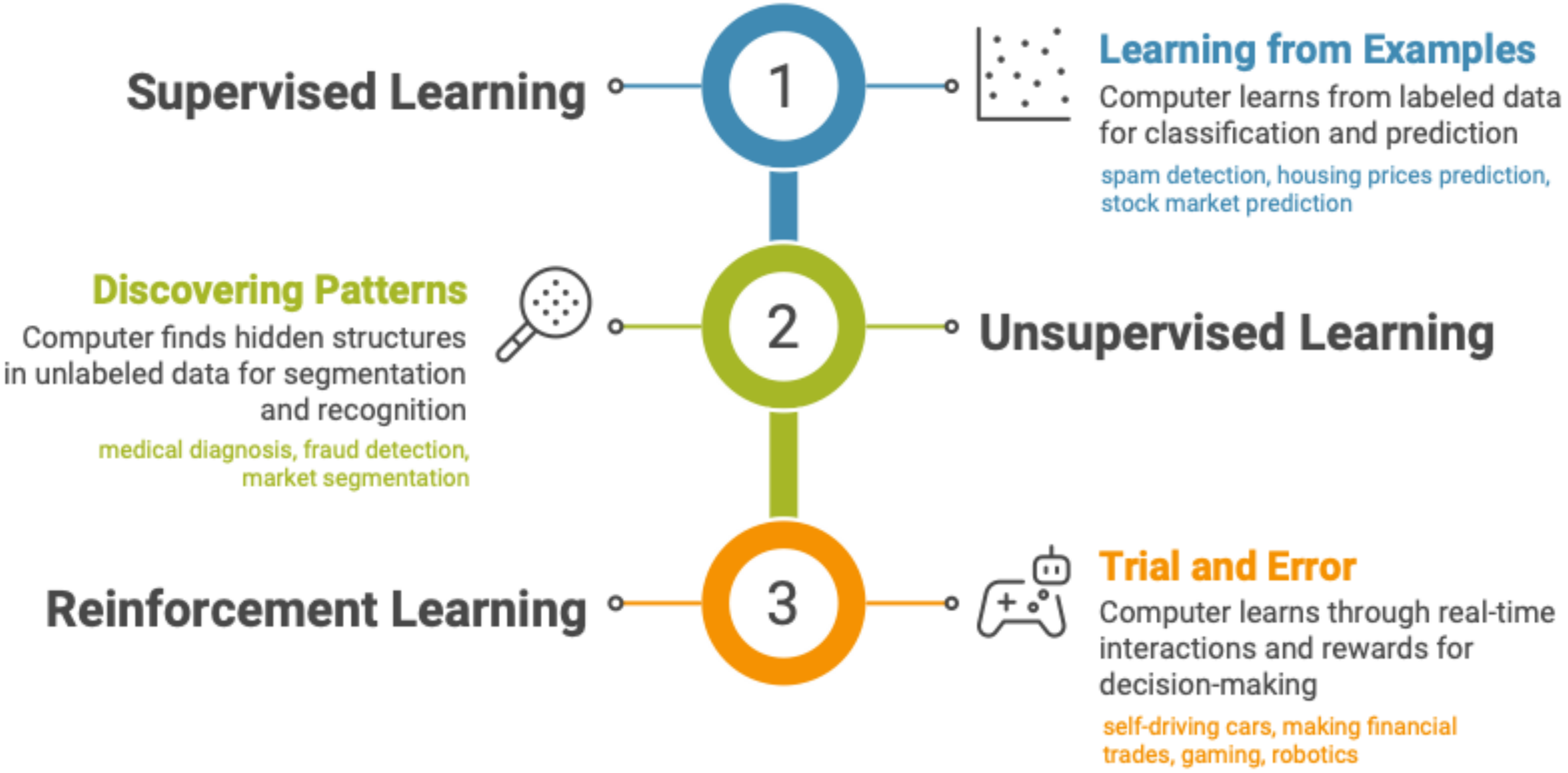
includes non NN algorithms

Artificial Intelligence (AI)

Systems performing human cognitive tasks (language, reasoning, perception, planning)



Learning paradigms (ML) *Ways of learning from data*



Let’s play: is it machine learning? 🤖

For each method family presented, we'll vote whether it qualifies as proper machine learning under these **four criteria**:

01

Parameter Learning

Learns parameters or structure directly from data through training

02

Objective Optimisation

Optimises an objective or loss function whose definition depends on the data

03

Mapping Generation

Produces a predictive or structural mapping that compresses data structure into a general form

04

Generalisation

Generalises beyond the specific problem instance to handle unseen inputs

Apply these criteria rigorously when voting.

Vote: Yes or No?

https://pollev.com/ansantam

Linear/kernel methods

Models that assume linear structure in the input space or feature space, often trained via convex optimisation or eigenvalue decompositions.

Examples:

- Linear regression (OLS)
- Ridge regression
- Lasso
- Elastic net
- Logistic regression
- Linear and kernel support vector machines (SVM)
- Kernel ridge regression (e.g. RBF)
- Linear discriminant analysis (LDA)
- Principal component analysis (PCA)
- Partial least squares (PLS)

Projecting correlated variables onto principal components

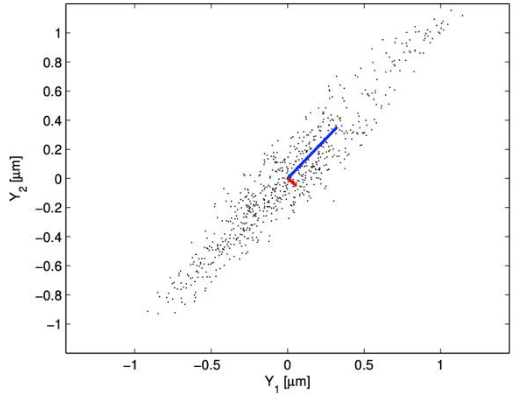


Figure 1: PCA for two highly correlated variables: two adjacent beam position monitors at Diamond Light Source. The blue and red lines indicate the first and second principal components respectively.

[Using principal component analysis to find correlations and patterns at diamond light source, C. Bloomer, 2014]

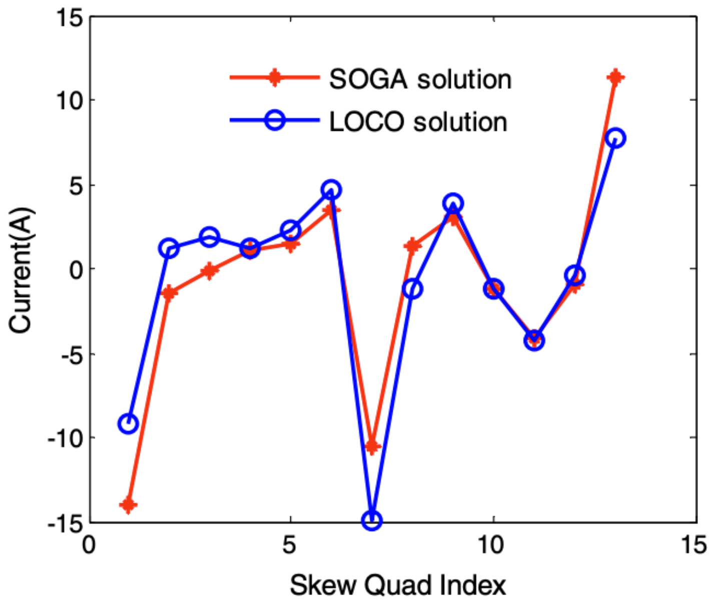
Population-based / evolutionary methods

Stochastic optimisation methods that evolve a population of candidate solutions using selection, mutation, and recombination.

Examples:

- Genetic algorithms (GA)
- Evolution strategies (ES, CMA-ES)
- Differential evolution (DE)
- Genetic programming (GP)
- Particle swarm optimisation (PSO)
- Ant colony optimisation (ACO)

Evolutionary optimisation applied to accelerator tuning



Machine based optimization using genetic algorithms in a storage ring, K. Tian, 2014

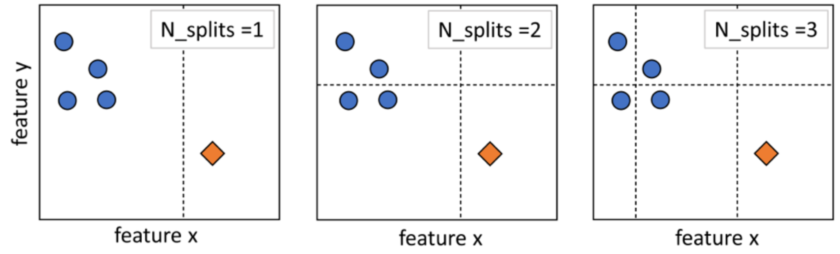
Tree-based methods

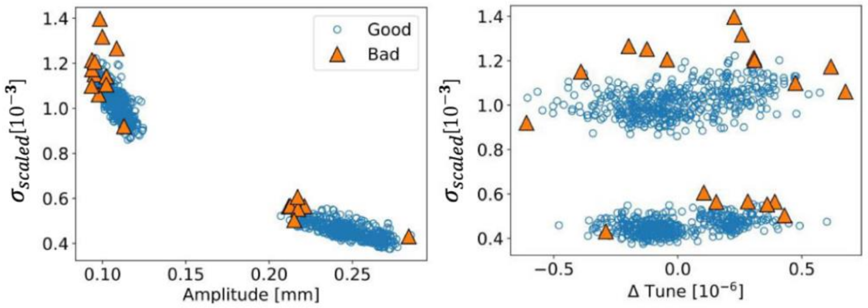
Models that partition the feature space recursively using decision rules, forming trees or ensembles of trees.

Examples:

- Decision trees
- Random forests
- Isolation forest
- Gradient boosting machines (GBM)
- XGBoost
- LightGBM
- AdaBoost
- Extremely randomised trees

Recursive feature space partitioning and anomaly detection using tree-based methods





[*Detection of faulty beam position monitors using unsupervised learning, E. Fol, 2020*]()

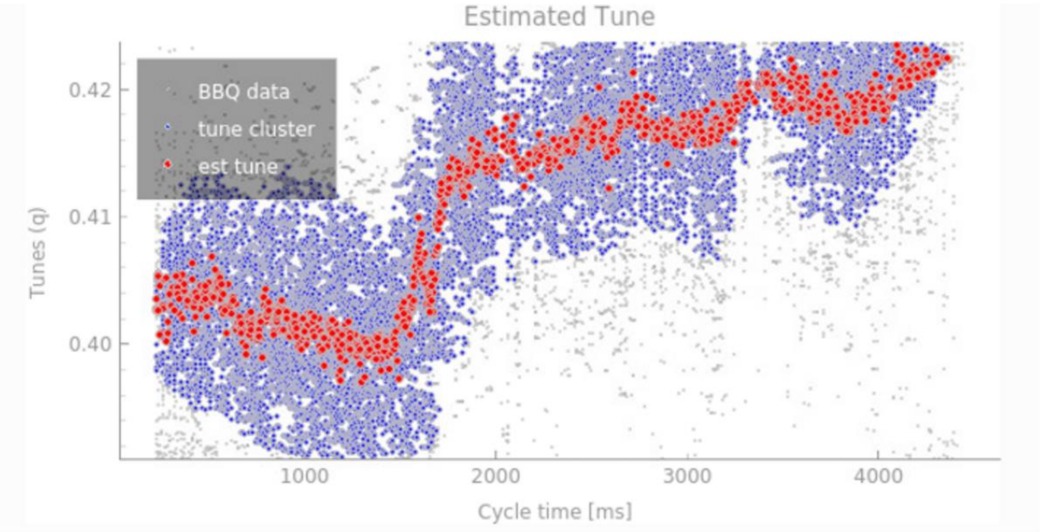
Geometry-based clustering

Data point clustering based on geometric relationships in feature space, typically using distances, neighbourhood structure or density.

Examples:

- DBSCAN
- OPTICS
- Mean shift
- k-means
- k-medoids
- Hierarchical clustering
- Spectral clustering
- Nearest-neighbour outlier scoring
- Local outlier factor (LOF)

Clustering in accelerator diagnostics



[Machine Learning for betatron tune diagnostics and control on the Super Proton Synchrotron at CERN, N. Gallou, 2023]

Geometry-based / deterministic direct search

Derivative-free optimisation methods that explore parameter space using geometric constructions or deterministic update rules.

Examples:

- Nelder-Mead simplex
- Pattern search
- Mesh adaptive direct search (MADS)
- Coordinate descent
- Powell's method
- Conjugate direction search (e.g. RCDS)
- Hooke-Jeeves
- Trust-region derivative-free methods

Comparison of derivative-free optimisation methods for accelerator tuning

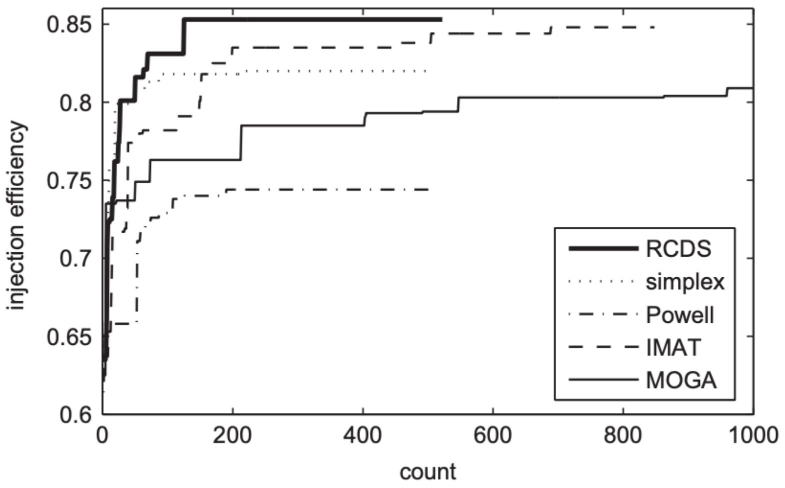


Fig. 7. The history of the best injection efficiency during optimization for the various algorithms.

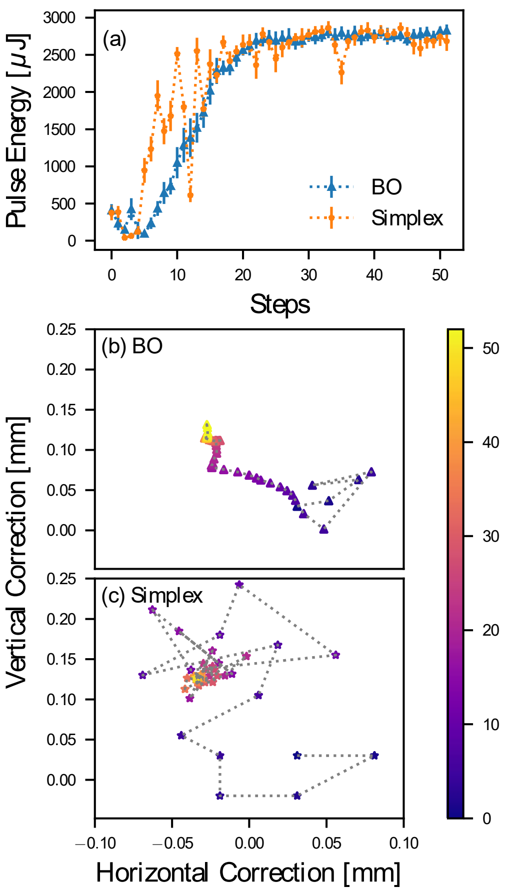
[An algorithm for online optimization of accelerators, X. Huang, 2013]

Probabilistic models

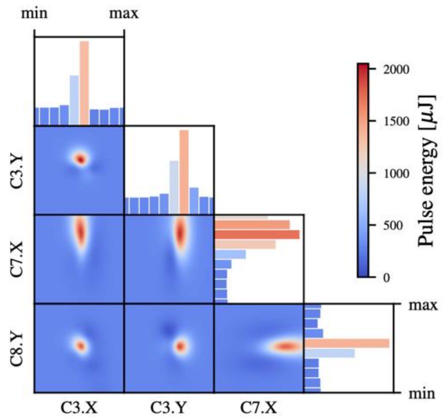
Models that represent uncertainty explicitly by learning probability distributions over variables, parameters, or latent structure.

Examples:

- Gaussian process regression (GP)
- Bayesian linear regression
- Bayesian neural networks
- Gaussian mixture models (GMM)
- Variational autoencoders
- Normalising flows
- Bayesian optimisation (probabilistic surrogate-based optimisation)



Gaussian process surrogate modelling for Bayesian optimisation



GP model can be used to visualize the sensitivity of actuators with respect to an objective and assist operators

[*Bayesian Optimization for SASE Tuning at the European XFEL, C. Xu, 2023*]

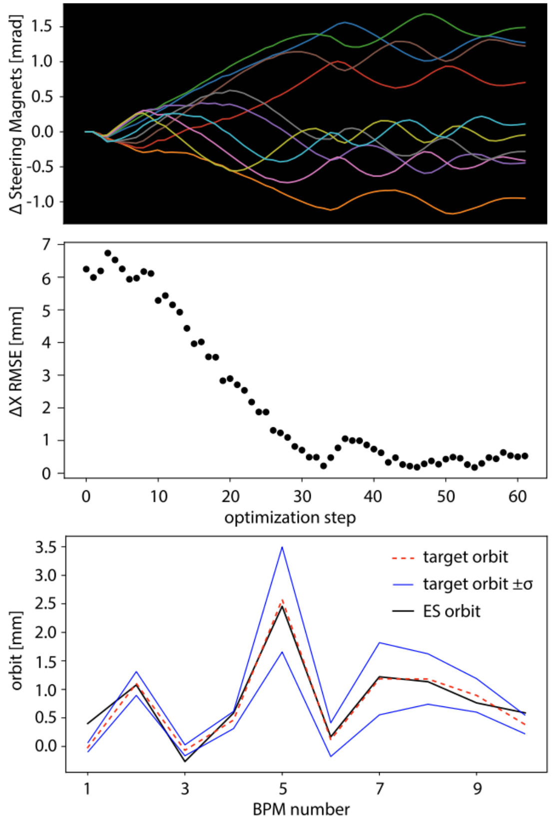
Perturbation-based feedback optimisation

Derivative-free optimisation methods that adjust parameters directly on the real system by applying small perturbations and using measured performance changes.

Examples:

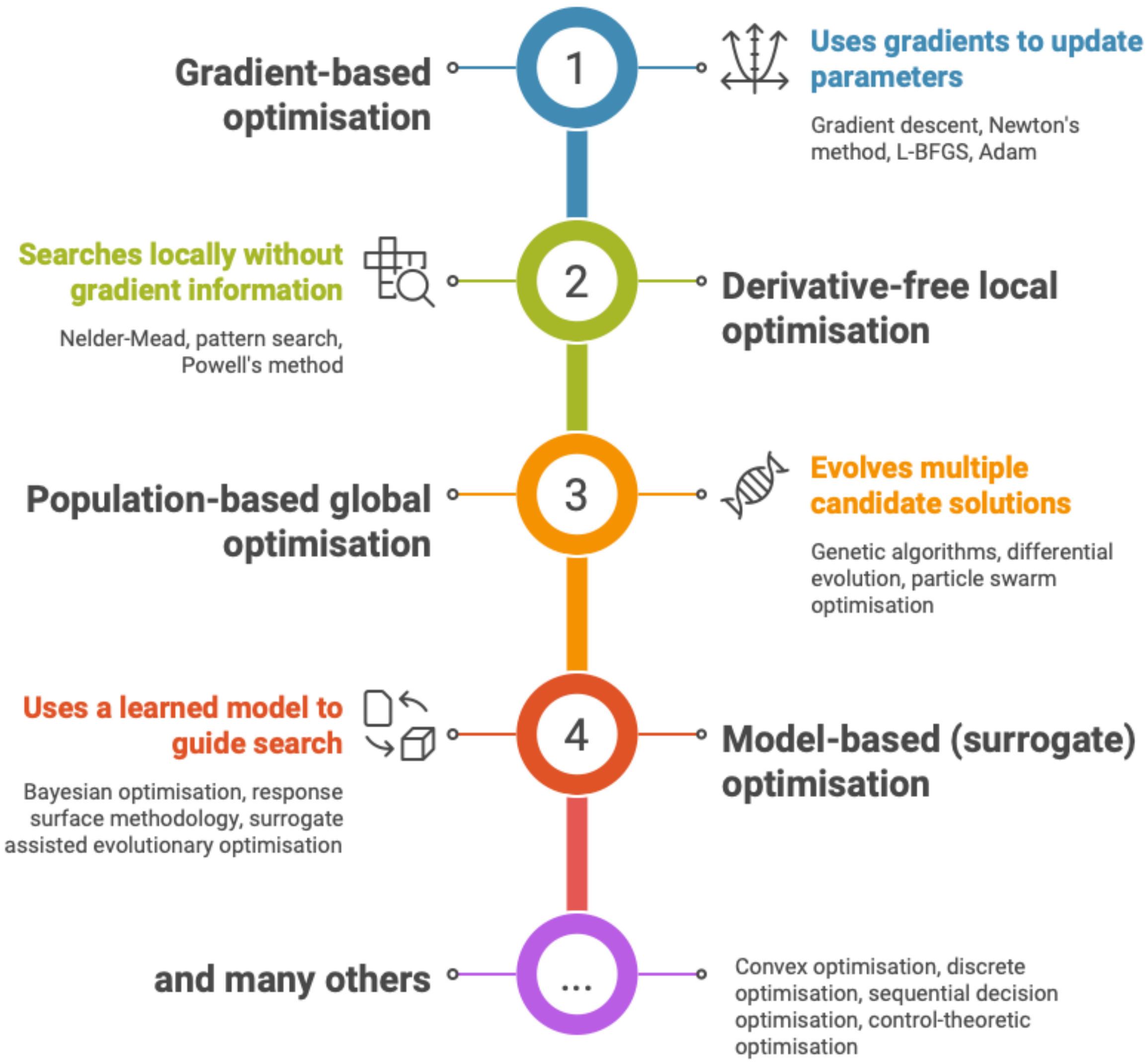
- Extremum seeking (ES) control
- Simultaneous perturbation stochastic approximation (SPSA)
- Finite-difference gradient estimation under noise (when used operationally rather than as model fitting)

Online optimisation using extremum seeking control



[Online multi-objective particle accelerator optimization of the AWAKE electron beam line for simultaneous emittance and orbit control, A. Scheinker, 2020]

Optimisation paradigms *Ways of searching for optima*



Machine Learning

Learning paradigm	Learning objective	Representative algorithms
Supervised Learning Input output pairs (x, y) Labelled data	Classification (discrete variables)	Logistic regression, linear SVM, kernel SVM, decision trees, random forests, gradient boosting classifiers, naïve Bayes, k-nearest neighbor classifier, neural networks
	Regression (continuous variables)	Linear regression (OLS, ridge, lasso), support vector regression (SVR), random forest regression, neural networks (MLP)
Unsupervised Learning Input only x Unlabelled data	Clustering	K-means, Gaussian mixture models (GMM), DBSCAN, OPTICS, spectral clustering, mean shift, k-medoids
	Representation learning	PCA, ICA, autoencoders, variational autoencoders (VAE), t-SNE, UMAP
	Density estimation	Kernel density estimation (KDE), Gaussian mixture models (GMM), normalizing flows, variational autoencoders (VAE)
Reinforcement Learning Interaction with environment + rewards	Policy learning	REINFORCE (policy gradients), PPO, TRPO, A2C/A3C, DDPG, TD3, SAC (actor–critic variants)
	Value estimation	Q-learning, SARSA, deep Q-networks (DQN), double DQN, dueling DQN, TD learning
	Model learning	GP dynamics models, learned neural dynamics models, Dyna-style methods, learned dynamics used in MPC, PETS-style probabilistic ensembles

Model families (across paradigms)

- Linear / kernel models
- Tree-based models
- Neural networks
- Probabilistic models

WHY AI FOR PARTICLE ACCELERATORS NOW?

What is tuning and control?

Tuning and control are often conflated, but they are fundamentally different.
In accelerators we usually refer to:

Tuning: adjustment of system parameters across repeated runs to optimise performance metrics

Operator-level optimisation

- Performed across **iterations**, not continuously
- Parameters are typically **static during a run**
- Historically done by **human operators**
- Limited to **few knobs at a time**
- Feedback is **slow**

→ Beam setup
Commissioning
…

What is tuning and control?

Tuning and control are often conflated, but they are fundamentally different.
In accelerators we usually refer to:

Control: Real-time adjustment of inputs in response to evolving system state to maintain or optimise behaviour

Real-time decision making

- Operates **within a run**
- Actions depend on **current system state**
- Requires **low latency**
- Must handle **delays** and **disturbances**
- **Stability** and **safety** constraints dominate

→

Feedback systems in accelerators (control loops):

- Bunch-by-bunch feedback system
- Low level RF

Classical control and learning-based methods

Classical controllers

- Highly effective when **accurate models** exist
- Extremely reliable for **well-characterized regimes**
- **Low operational risk** and **strong interpretability**

Operational usability of models for real-time control

models are accurate and usable online

models are accurate but partially observable or slow

models are inaccurate or unavailable

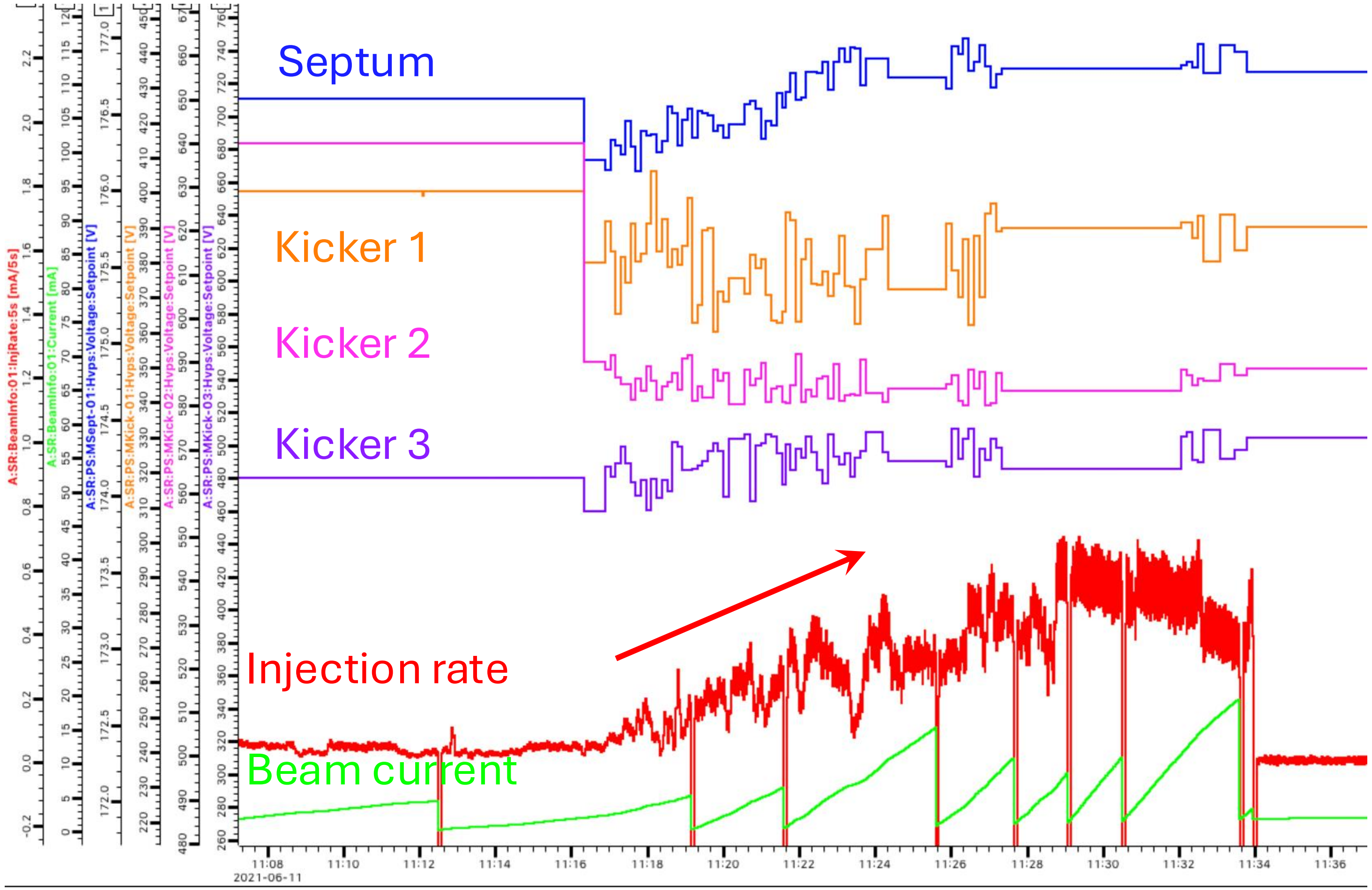
ML-based methods become relevant when

- System **models are incomplete** or hard to derive
- **Dynamics change** across operating regimes
- **Objectives go beyond stability** (e.g. performance trade-offs, multi-objective optimisation)
- **Adaptation matters** more than guarantees

As feedback becomes **richer** and more **delayed**, learning-based methods can **exploit information** that is **difficult to encode** in classical designs

Example of tuning with ML

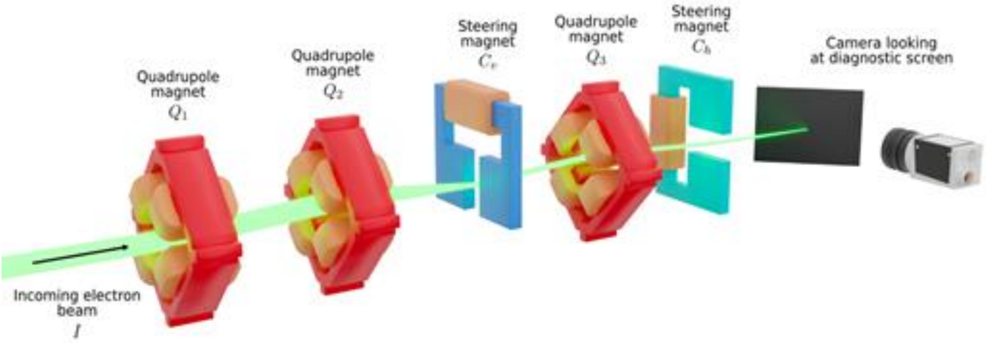
Injection efficiency optimization with Bayesian optimization at KARA (KIT)

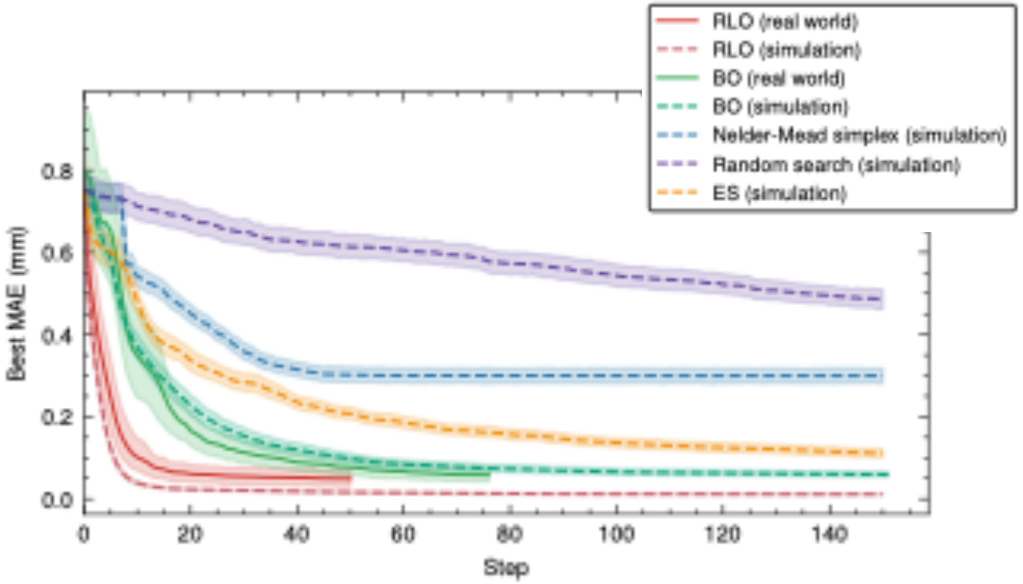


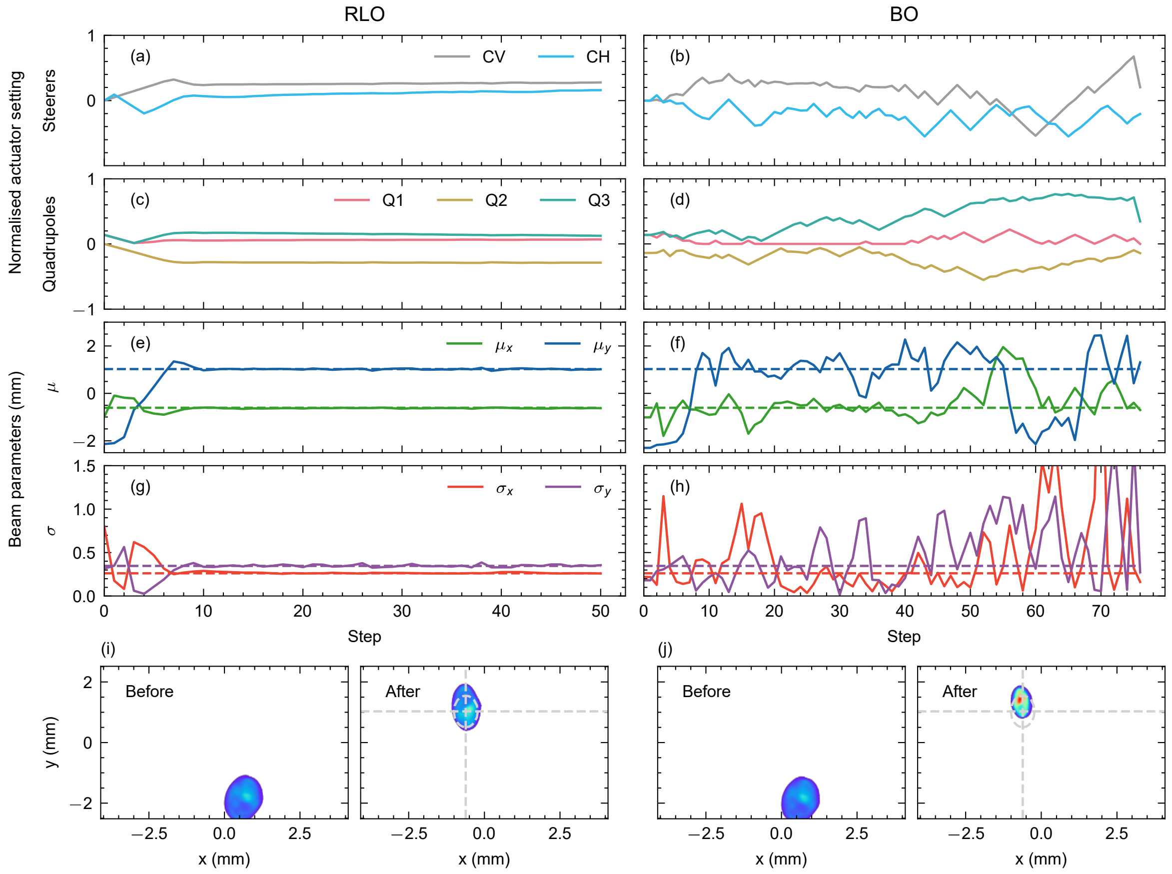
[Bayesian optimization of the beam injection process into a storage ring, C. Xu, 2023]()

Example of tuning with ML

Automatic beam steering and focusing with ML at ARES, DESY



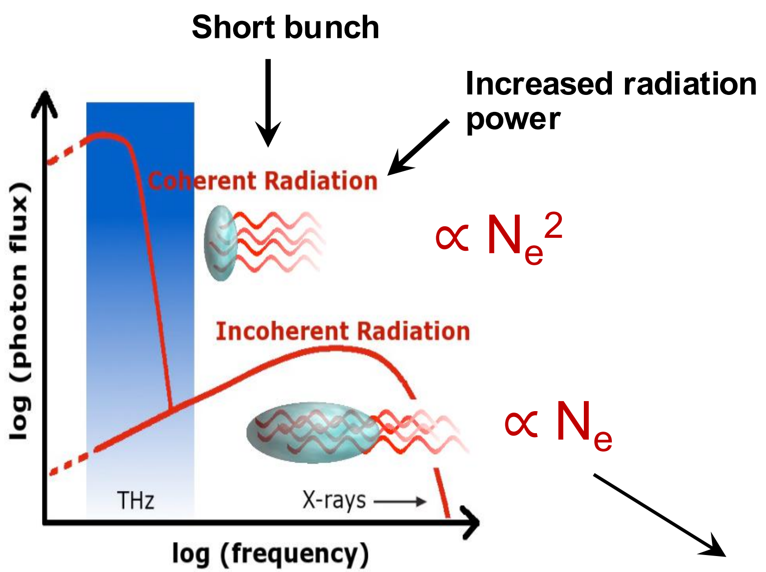


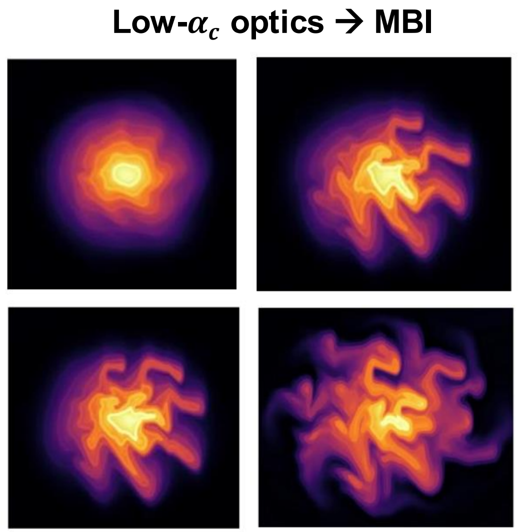


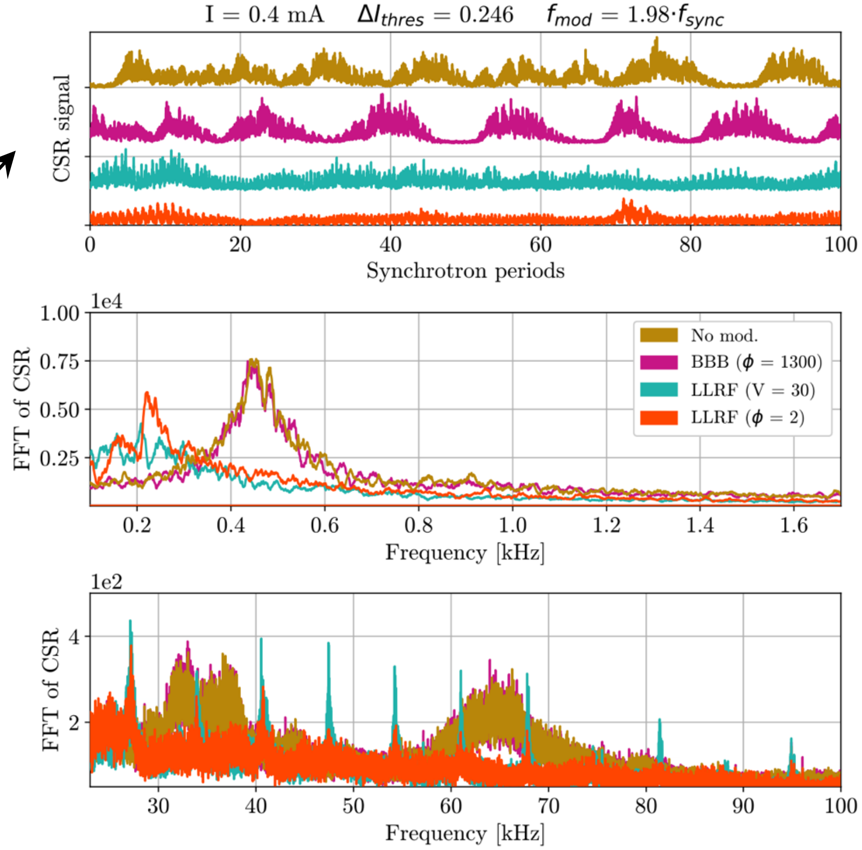
[Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning, J. Kaiser, 2024]

Example of control with ML

Control of the microbunching instability at KARA (KIT)

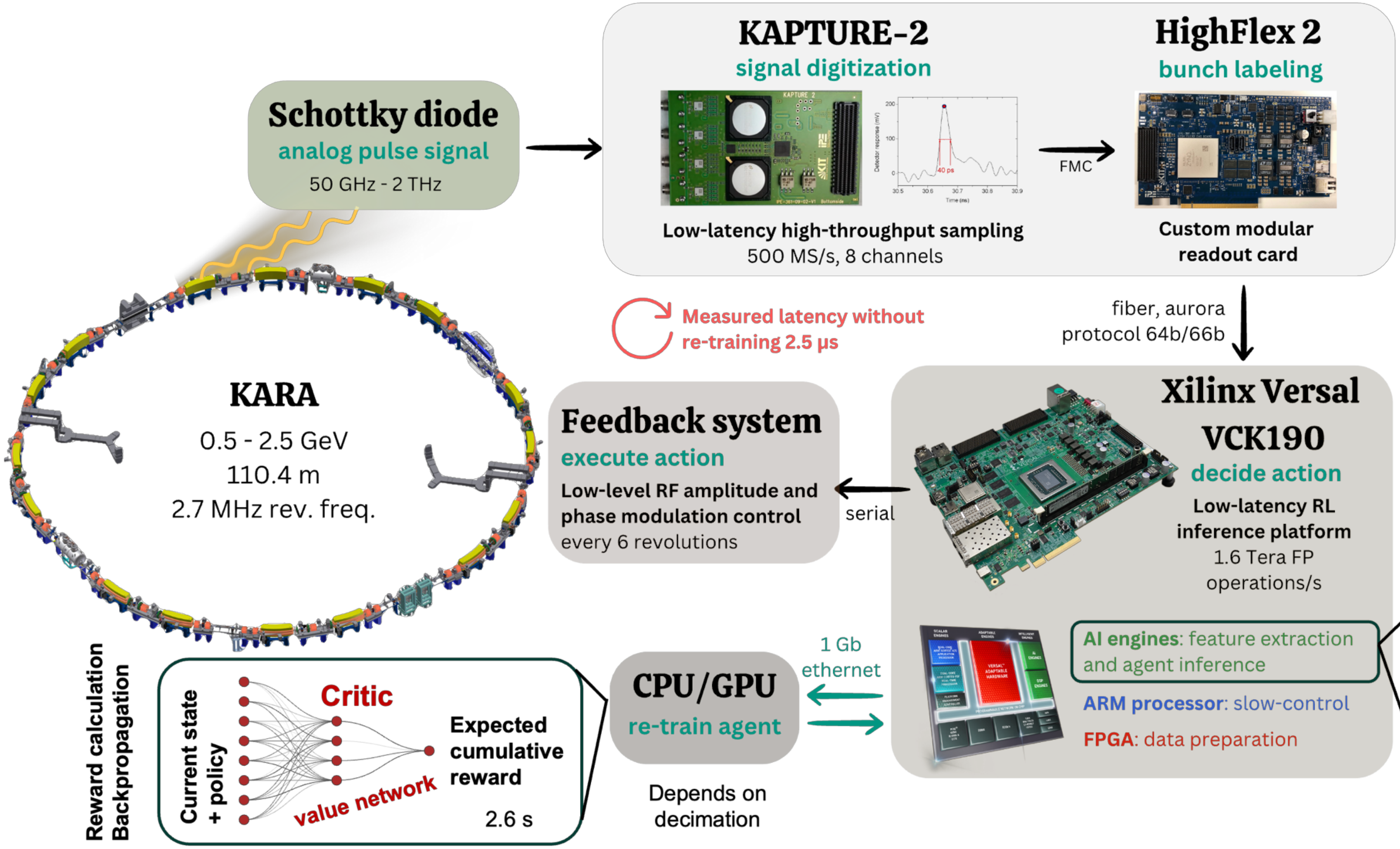






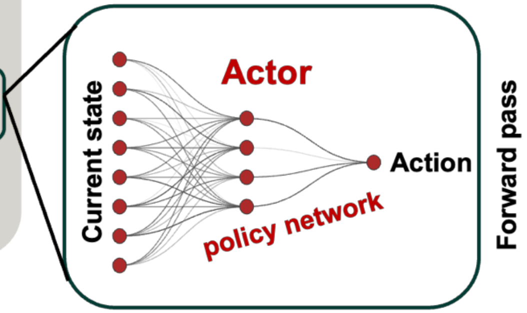
Example of control with ML

Control of the microbunching instability at KARA (KIT)



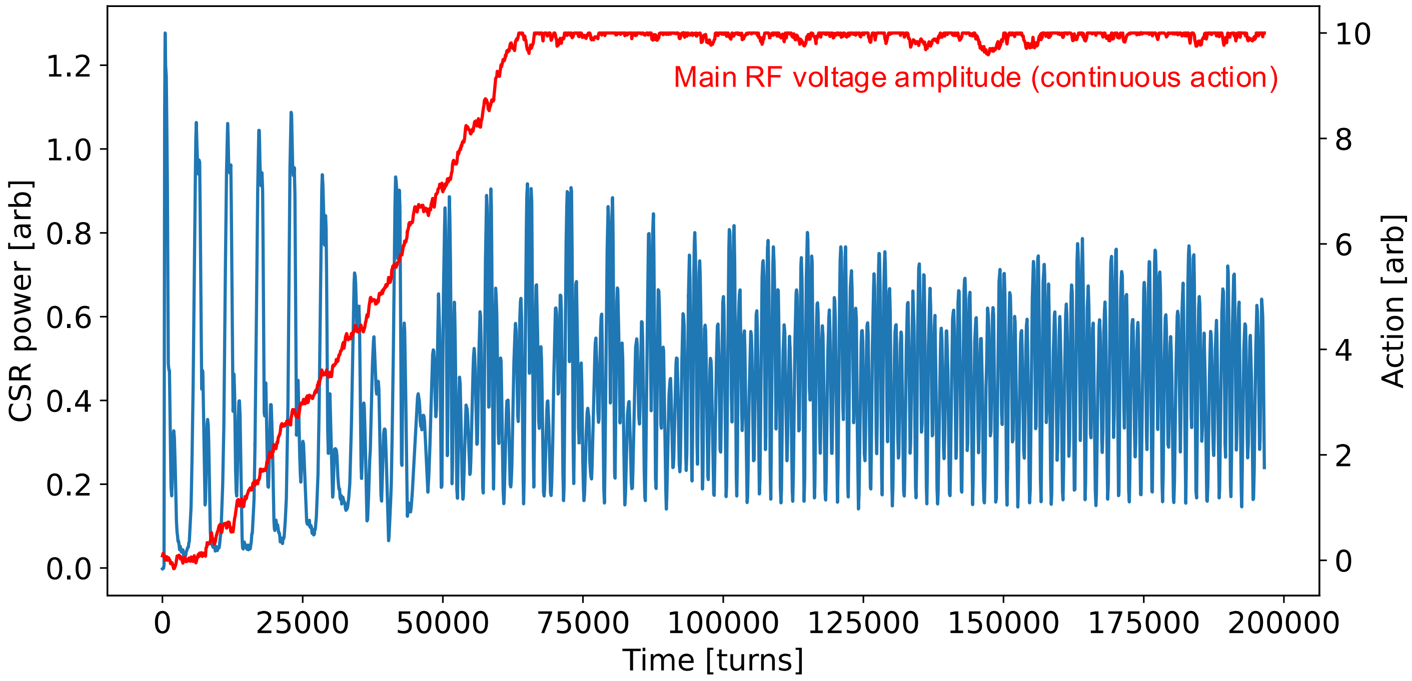
First real-time experiment-based RL learning on hardware in accelerators

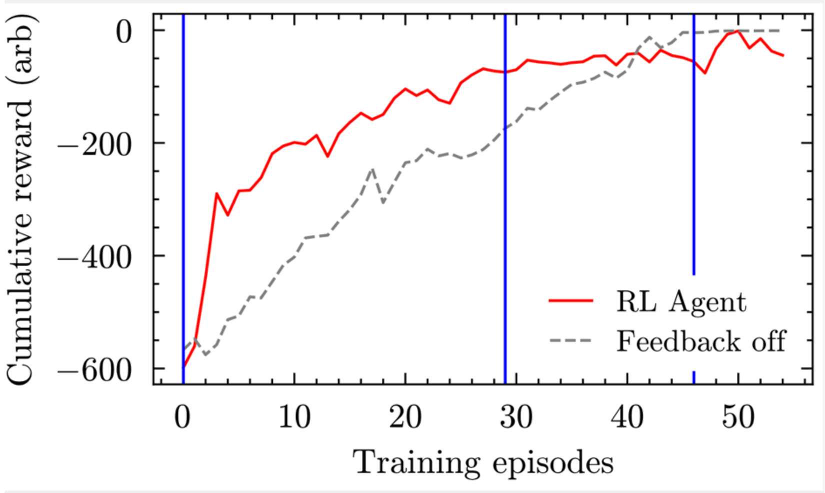
[Doctoral thesis L. Scomparin]



Example of control with ML

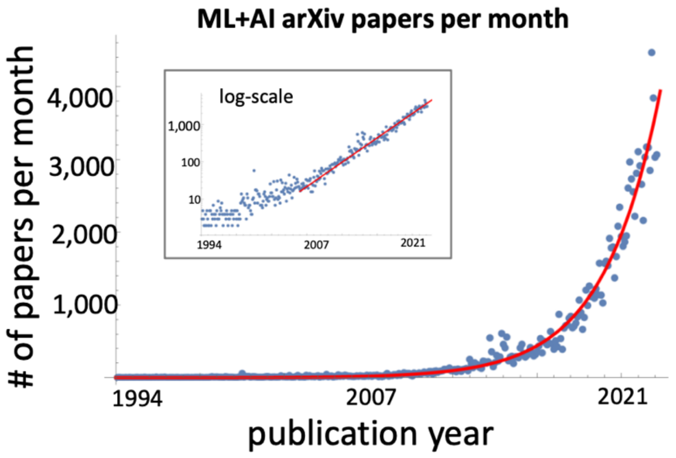
Control of the microbunching instability at KARA (KIT)





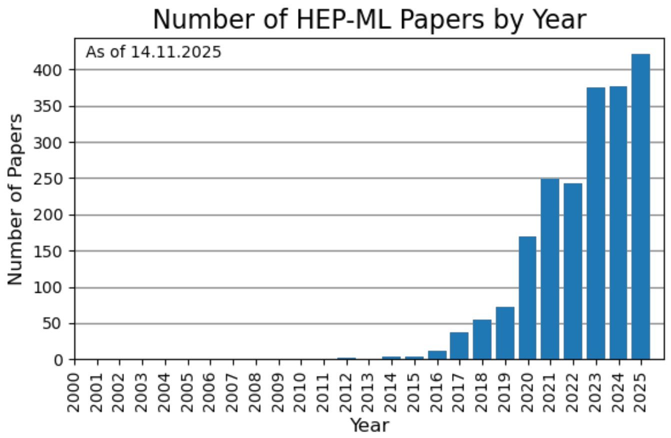
A look into AI publication numbers

Total



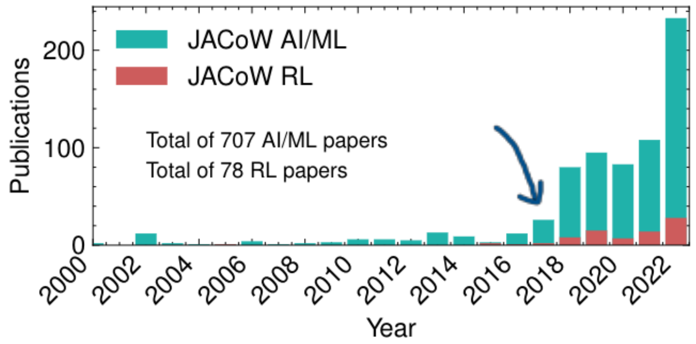
https://arxiv.org/pdf/2210.00881

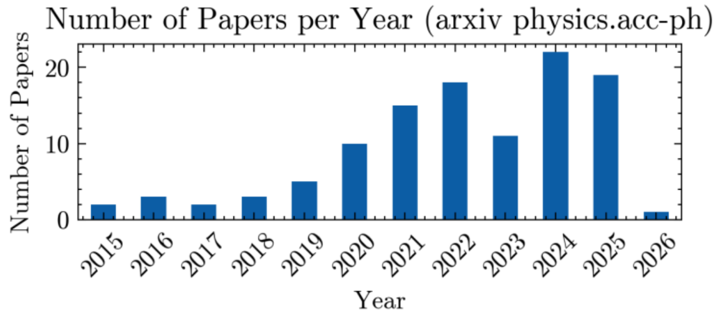
High energy physics



https://iml-wg.github.io/HEPML-LivingReview/

Accelerator physics





https://github.com/MALAPA-Collab/AccML-LivingReview/tree/main

AI in particle accelerators

- Particle accelerators are **large, nonlinear, data-rich, and difficult to model in their entirety**, making them a natural candidate for ML
- Initial attempts (1980s-1990s) included rule-based AI and early neural nets for beam control, orbit correction, and fault detection
  - ➢ **Limited by poor hardware, algorithms, and lack of data infrastructure**

SLAC-PUB-4091
September 1986
(A)

Some Applications of AI to the Problems of Accelerator Physics*

T. Higo†, H. Shoaee and J. E. Spencer
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94305

Abstract

Failure of orbit correction schemes to recognize betatron oscillation patterns obvious to any machine operator is a good problem with which to analyze the uses of Artificial Intelligence and the roles and relationships of operators, control systems and machines. Because such error modes are very common, their generalization could provide an efficient machine optimization and control strategy. A set of first-order, unitary transformations connecting canonical variables through measured results are defined which can either be compared to design for commissioning or to past results for 'golden orbit' operation. Because these relate directly to hardware variables, the method is simple, fast and direct. It has implications for

AI in particle accelerators

Opportunities in Machine Learning for Particle Accelerators

Editors

A. Edelen and C. Mayes
SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

D. Bowring
Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Contributors

D. Ratner
SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

A. Adelmann, R. Ischebeck, and J. Snuverink
Paul Scherrer Institut, Villigen, Switzerland

I. Agapov and R. Kammering
Deutsches Elektronen-Synchrotron, Hamburg, Germany

J. Edelen
Radiasoft, LLC, Boulder, Colorado 80301, USA

I. Bazarov
Cornell University, Ithaca, NY 14853, USA

G. Valentino
University of Malta, Msida, Malta

J. Wenninger
CERN, Geneva, Switzerland

arXiv:1811.03172v1 [physics.acc-ph] 7 Nov 2018

Current and emerging applications

Anomaly detection and machine protection

- Detect subtle system instabilities before failures (e.g. magnet quenches, cavity faults).

➢ <u>Examples</u>: *LHC anomaly detection for beam monitors and collimator alignment.*

System modelling and surrogates

- ML can create fast, accurate surrogate models blending physics simulations and real data
- Enables real-time control and faster design optimisation.

➢ <u>Example</u>: *FAST facility reduced simulation time from 20 min → <1 ms using neural nets*.

AI in particle accelerators

Opportunities in Machine Learning for Particle Accelerators

Editors

A. Edelen and C. Mayes

SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

D. Bowring

Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Contributors

D. Ratner

SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

A. Adelmann, R. Ischebeck, and J. Snuverink

Paul Scherrer Institut, Villigen, Switzerland

I. Agapov and R. Kammering

Deutsches Elektronen-Synchrotron, Hamburg, Germany

J. Edelen

Radiasoft, LLC, Boulder, Colorado 80301, USA

I. Bazarov

Cornell University, Ithaca, NY 14853, USA

G. Valentino

University of Malta, Msida, Malta

J. Wenninger

CERN, Geneva, Switzerland

arXiv:1811.03172v1 [physics.acc-ph] 7 Nov 2018

Current and emerging applications

Virtual diagnostics (virtual instrumentation)

- Predict measurements where diagnostics are destructive, slow, or unavailable.
- Neural networks at LCLS used fast signals to predict photon energy and spectra.

➢ Example: *Surrogate imaging diagnostics used for phase space reconstruction.*

Tuning and control

- ML models reduce time to switch machine settings and optimise parameters.

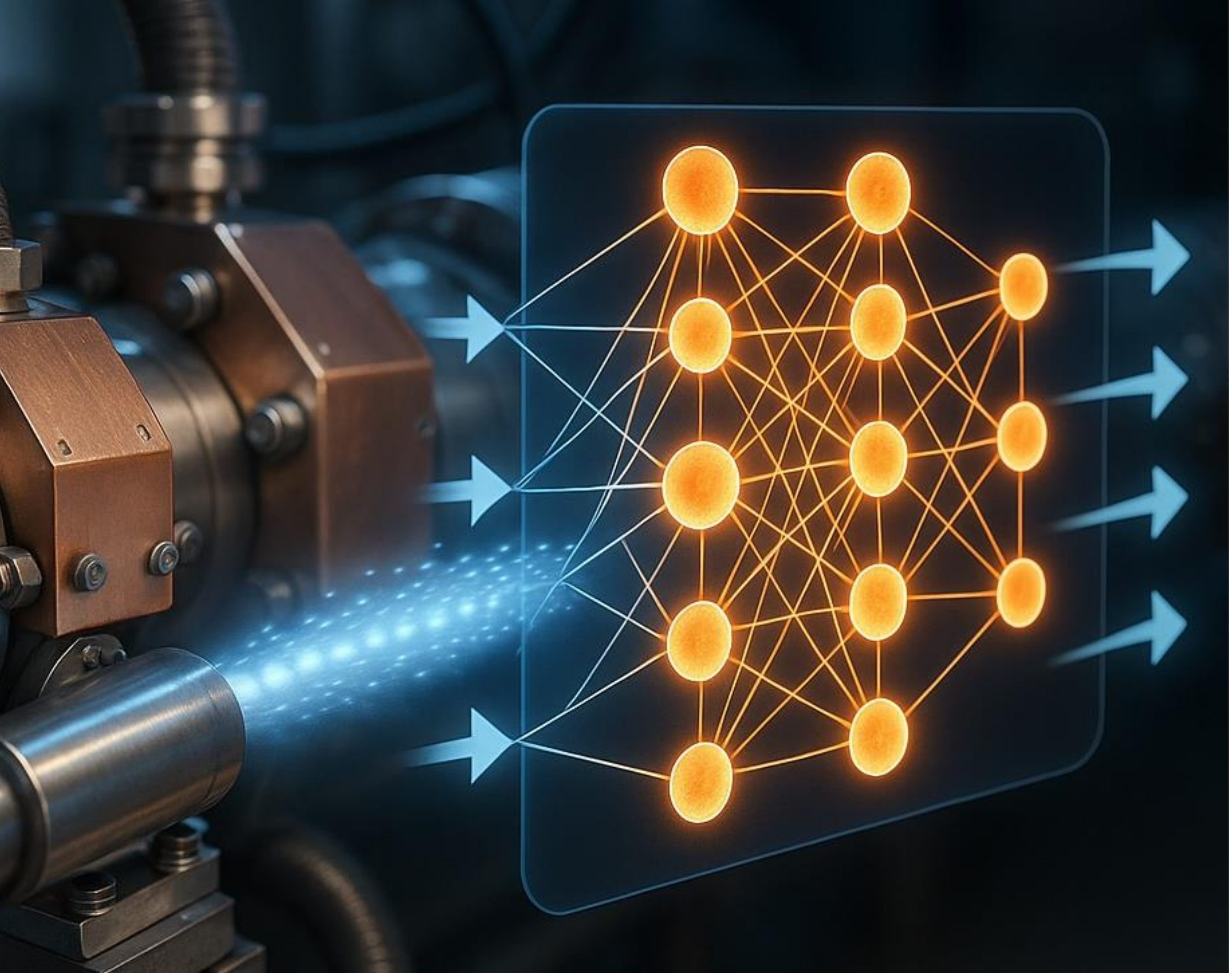
➢ Examples: *Reinforcement learning (RL) doubled FEL power at LCLS. Bayesian optimisation tuned quadrupoles efficiently while avoiding damaging states.*

AI in particle accelerators

Task	Goal	Methods/Concepts	Examples[1]
Detection	Detect outliers and anomalies in accelerator signals for interlock prediction, data cleaning	• Anomaly detection • Time series forecasting • Clustering	• Collimator alignment • Optics corrections • SRF quench detection
Prediction	Predict the beam properties based on accelerator parameters	• Virtual diagnostics • Surrogate models • Active learning	• Beam energy prediction • Accelerator design • Phase space reconstruction
Optimization	Achieve desired beam properties or states by tuning accelerator parameters	• Numerical optimizers • Bayesian optimization • Genetic algorithm	• Injection efficiency • Radiation intensity
Control	Control the state of the beam in real time in a dynamically changing environment	• Reinforcement learning • Bayesian optimization • Extremum Seeking	• Trajectory steering • Instability control

[1] non-exhaustive

From automated and efficient accelerator operation to faster simulations, ML is enabling new ways of **designing** and **operating** particle accelerators



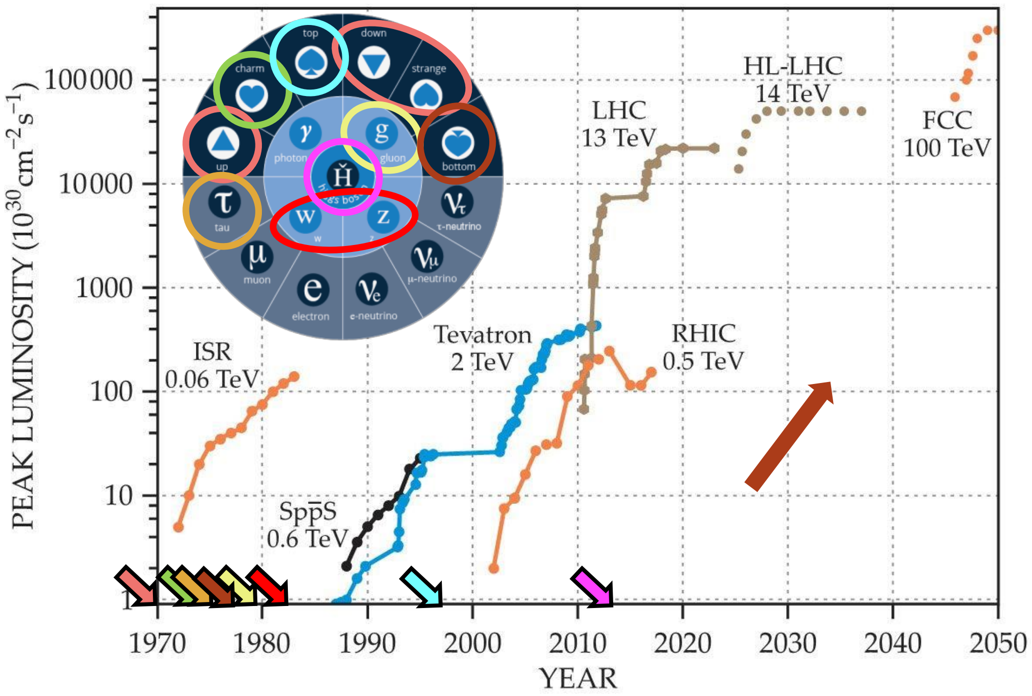
Today, AI is ready to become a core tool for accelerator design and operation, but success depends on **data quality, infrastructure, and institutional support**

WHY AI FOR FUTURE PARTICLE ACCELERATORS?

The particle accelerator roadmap

Ability to generate new particles via high-energy collisions

Energy frontier



Ability to probe atomic structures

Photon science

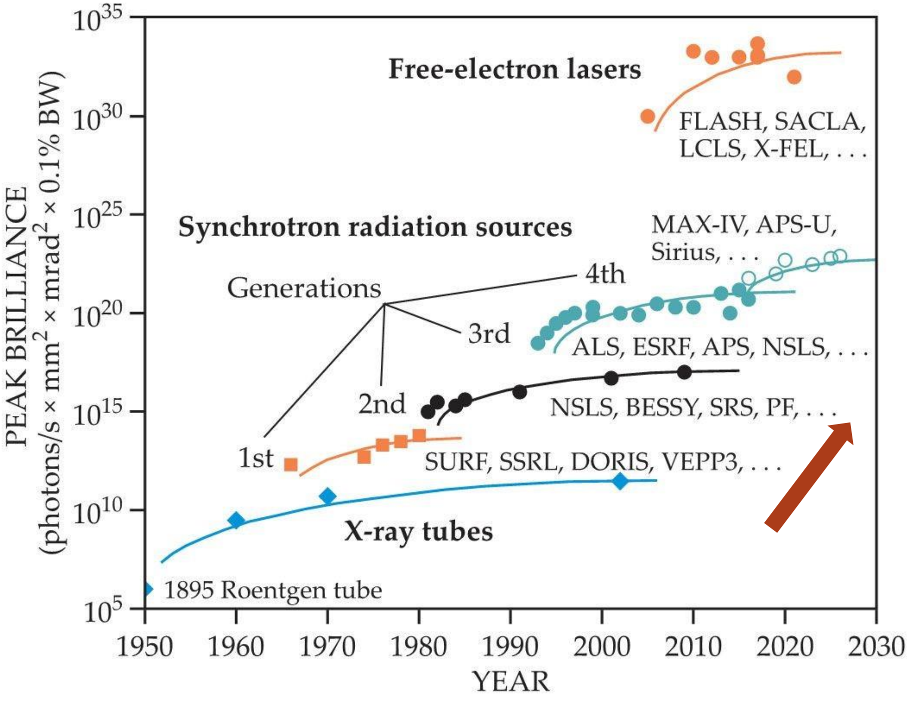


Image source: "Particle beams behind physics discoveries" (Physics Today)

Technological innovation is needed to keep up with the **upcoming challenges**!

Trends and challenges of frontier accelerators

Denser beams for higher luminosity and brilliance

- Complex beam dynamics
- Complex accelerator design and operation

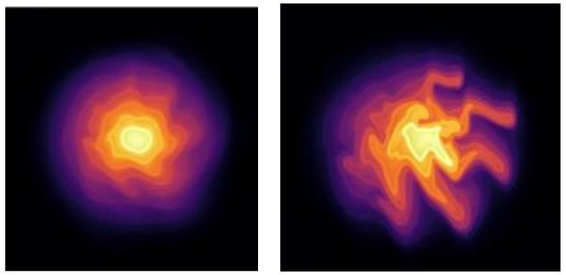


Image: KIT

Larger circular colliders for higher energies

- Orders of magnitude more signals
- Machine protection limits

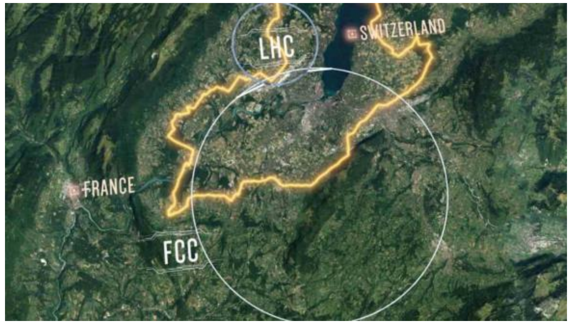


Image: CERN

Compact plasma accelerators with higher gradients

- Tight tolerances
- High-quality beams required

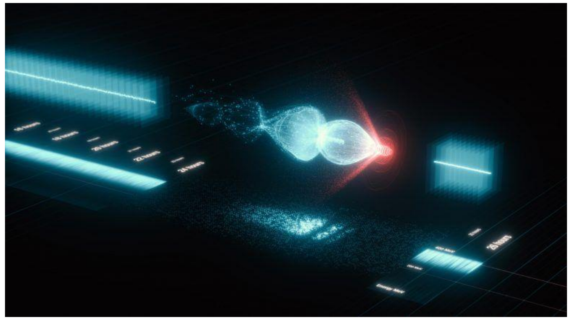
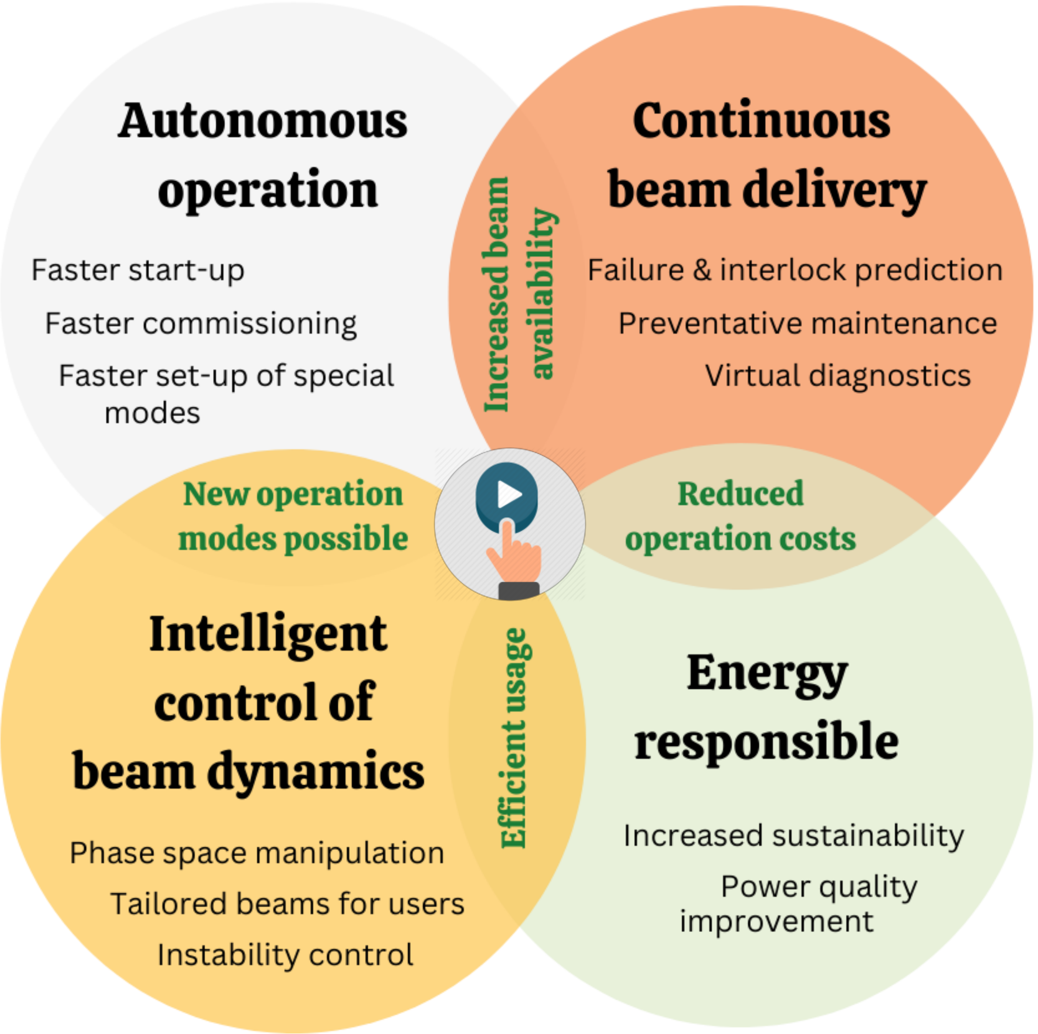


Image: DESY

[How can machine learning help future light sources?, A. Santamaria Garcia, 2023]

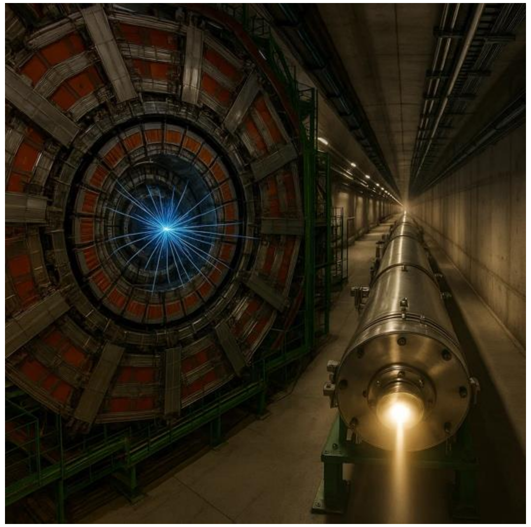
A vision for future accelerators, driven by AI



[How can machine learning help future light sources?, A. Santamaria Garcia, 2023]

New Capabilities

- Large-scale particle physics experiments
- Next-generation accelerators
- Advanced detectors and instrumentation.



+

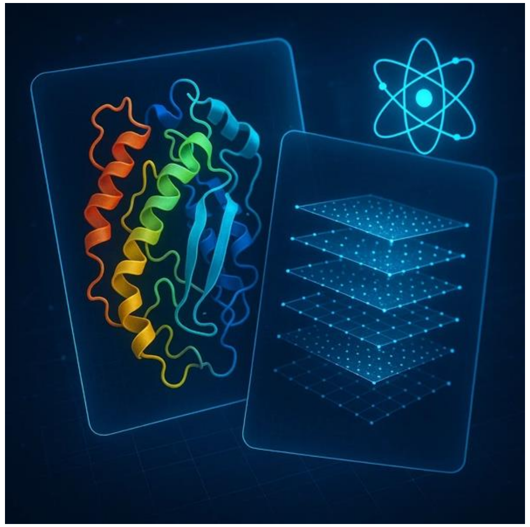
Technological Advances

- AI for data analysis, optimisation, and control
- Exascale computing (unprecedented simulation power)
- Cloud computing (scalable and collaborative workflows)
- Quantum computing (novel approaches to hard physics problems)

=

Scientific Progress

- New discoveries in particle and nuclear physics
- Breakthroughs in materials science and biology
- Deeper understanding of fundamental processes



HISTORICAL CONTEXT:
THE BIRTH AND EVOLUTION OF AI

AI history in a nutshell

Early foundations (1940s-1970s)

So far, so good

Pure academic curiosity & optimism
Beginning of connectionism

Turing test invented

1943 Perceptron model of a neuron
W. McCulloch, W. Pitts

First mathematical model of computation inspired by biology

1951 "A Stochastic Approximation Method"
H. Robbins, S. Monro

Mathematical foundation of stochastic gradient descent (core optimisation method)

1958 Perceptron implementation
F. Rosenblatt

First physical neural net machine with analog circuitry. Neural nets → tangible!

1959 "Machine Learning" term coined
A. Samuel

Demonstrates self-learning checkers, the first program to improve with experience!

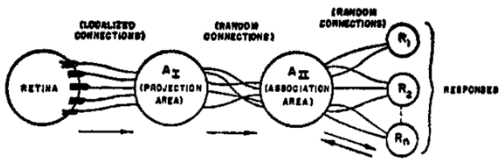
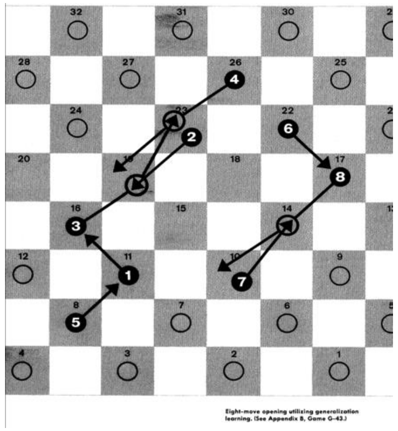


FIG. 1. Organization of a perceptron.



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

The first AI hype

AI history in a nutshell

Early foundations (1940s-1970s)

“Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed”

Arthur Samuel (1959)

AI history in a nutshell

Early foundations (1940s-1970s)

No path forward 👎

Lack of methods and resources
Lack of funding

1969 — Perceptron (book)
M. Minsky, S. Papert

Rigorous mathematical proof of what perceptrons couldn't do! which is solving nonlinear problems

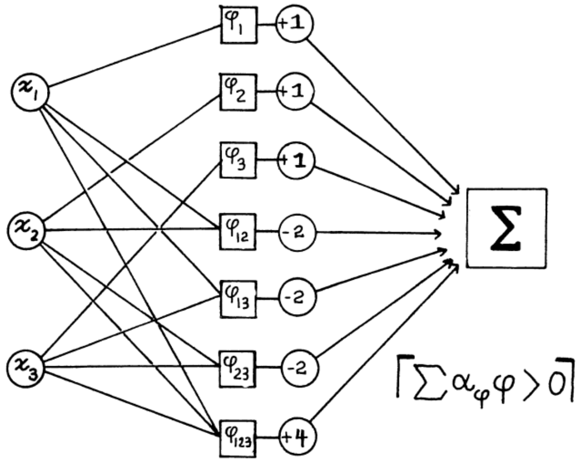
1973 — The Lighthill report
J. Lighthill

Government-commissioned review of AI research in the UK and presented in Parliament. Declared AI progress "disappointing" and concluded that AI research had failed.

1974 — AI winter

Collapse of all AI research due to funding collapse (symbolic and connectionism)

First PC
Altair 8800



Group-invariant coefficients for the $|R| = 3$ parity predicate.

AI history in a nutshell

Recovery and statistical era (1980s-1990s)

Recovering

We got the core methods
We got a practical example
Still slow, data scarce

World Wide Web invented(1990)

1986 — Backpropagation popularised
D. Rumelhart, G. E. Hinton, R. J. Williams

Finally! A way to efficiently train multilayer perceptrons

1993 — Handwritten digit recognition with CNNs
Y. LeCun et al.

First large-scale, commercially deployed neural network system solving a real-world problem

1995 — Support Vector Machines
V. Vapnik et al.

Max-margin classifiers that only use “support vectors” to define a nonlinear decision boundary. Dominant ML method and benchmark

What was actually wrong with backpropagation in 1986?

- We all drew the wrong conclusions about why it failed. The real reasons were:

1. Our labeled datasets were thousands of times too small.
2. Our computers were millions of times too slow.
3. We initialized the weights in a stupid way.
4. We used the wrong type of non-linearity.

AI history in a nutshell

Web + data era (2000s)

2001 Google image search
AI became consumer-facing

2006 Deep Belief Nets revive neural networks
G. E. Hinton

Early deep learning

2009 ImageNet dataset released
J. Deng et al.

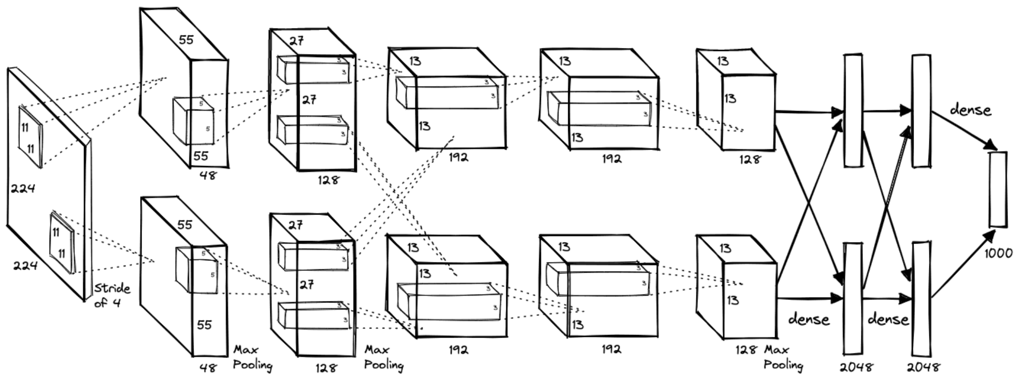
Large-scale visual database. Benchmark dataset that enabled measurable progress in object recognition

2006-2010 Statistical Machine Translation (Google Translate v1)

AI history in a nutshell

Deep learning revolution (2010s)

2012 AlexNet wins ImageNet
A. Krizhevsky, G. E. Hinton



2014 Generative Adversarial Networks (GANs)
I. Goodfellow

2015 ResNet
K. He et al.

2016 AlphaGo defeats Lee Sedol
DeepMind

2017 Transformer architecture introduced
Vaswani et al.

2017 AlphaZero
DeepMind

2018-2019 BERT, GPT-2

AI history in a nutshell

Foundation model era (2020s→ now)

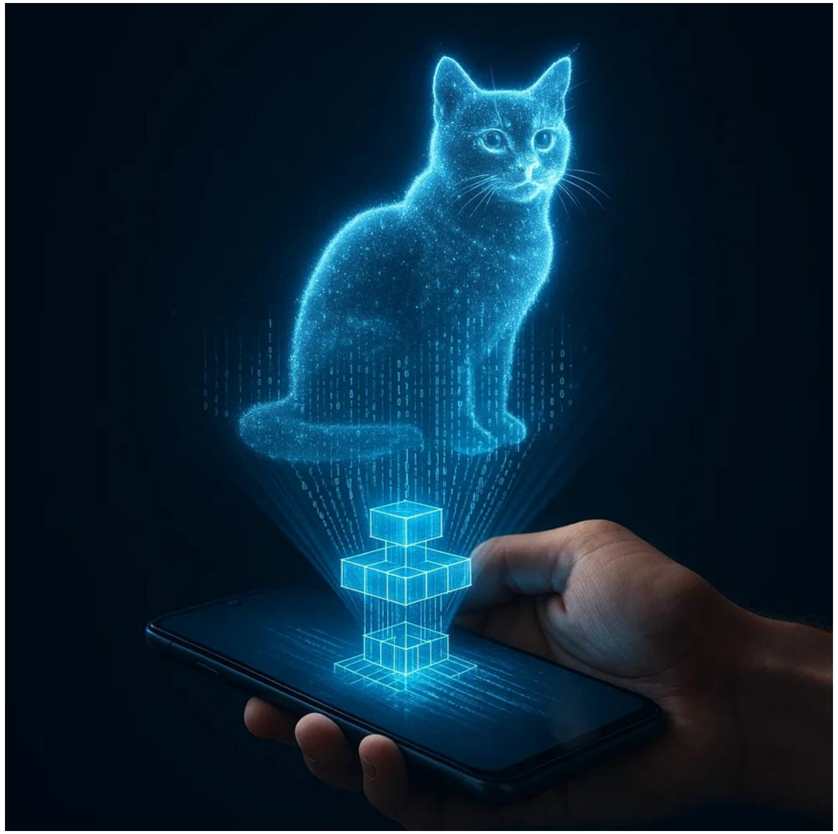
The rest is history! 🙌

2020 GPT-3
Scaling laws, few shot learning

2022 ChatGPT
Mass adoption of LLMs

2022 Stable Diffusion, DALL.E 2
Diffusion models for images

2023-2024 GPT-4, Claude, Gemini, open source LLMs
Foundation models



Created with Sora

Why exactly is AI booming now?

What was missing in the first AI wave?

Big Data

- Internet-scale datasets (text, images, video, code)
- Automated data collection pipelines (web scraping, sensors, logs)
- Cheap cloud storage (petabytes)

Software

- Open-source frameworks lowered entry barriers (PyTorch, Jax, etc.)
- New training techniques (self-supervision, transfer learning, RLHF)
- Transformer architecture unlocked scalable models

Hardware

- GPUs & TPUs enable massive parallelism
- Cloud computing democratised access to supercomputing
- Specialised chips (AI accelerators, NVIDIA H100) keep pushing scale

New buzzwords

LLMS, Foundation models, GenAI, Agentic AI

Unfortunately, all referred to as "AI"

- *Foundation and LLM: GPT-4, Claude 3, Gemini, LLaMA 3*
- *Non-foundation and LLM: early GPT-2, domain specific LLMs, customer support*
- *Foundation and not LLM: Stable diffusion, DALL.E*

Training and reuse

How it was trained and why it exists

Foundation model

- *Foundation model: static artefact*
- *Agency: feedback, memory, goals*

Model/architecture

What kind of model it is

Large language model

ChatGPT

System behaviour

How it is orchestrated to act

Agentic AI

- *LLMs ∈ generative models*
- *Not all uses of LLMs are GenAI (output can be labels, structured prediction, vector representations, scoring functions)*

Capability

What is it used to produce

GenAI

- *GenAI with no agency (just generate)*
- *Non generative agentic systems*

Generation ≠ Agency

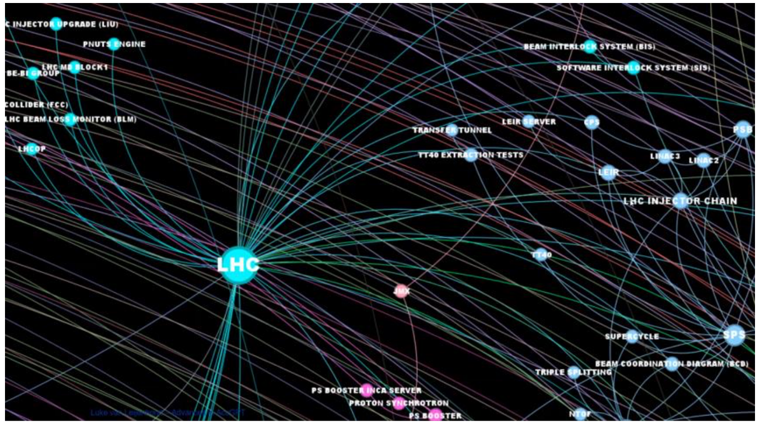
New buzzwords

LLMS, Foundation models, GenAI, Agentic AI

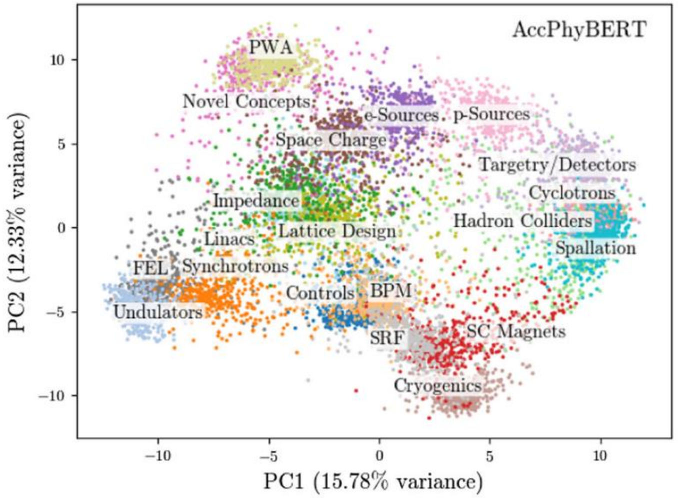
Large language model (LLM):
language-generating model

→ a deep neural network, typically a **transformer**, trained on large text corpora to model the **probability distributions** of **token sequences** and generate language.

Is a model class. It does not act, decide, or pursue goals, not inherently "intelligent"



[AccGPT = a chatbot pilot for CERN specific knowledge retrieval]()



[Domain-specific text embedding model for accelerator physics, T. Hellert, 2025]()

New buzzwords

LLMS, Foundation models, GenAI, Agentic AI

Foundation model: reusable, general-purpose pretrained model (training/reuse level)

→ large, general-purpose model trained on broad data at scale once at high cost and reused many times (**unimodal or multimodal**).

Most modern LLMs are foundation models, but not all foundation models are LLMs (e.g., stable diffusion, AlphaFold)

Describes how the model is trained and reused, but not the output

Foundation Models

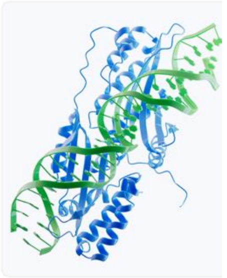
Language
- GPT-5
- Claude 3
- LLaMA
- DeepSeek

Vision
- DALL-E 3
- Stable Diffusion

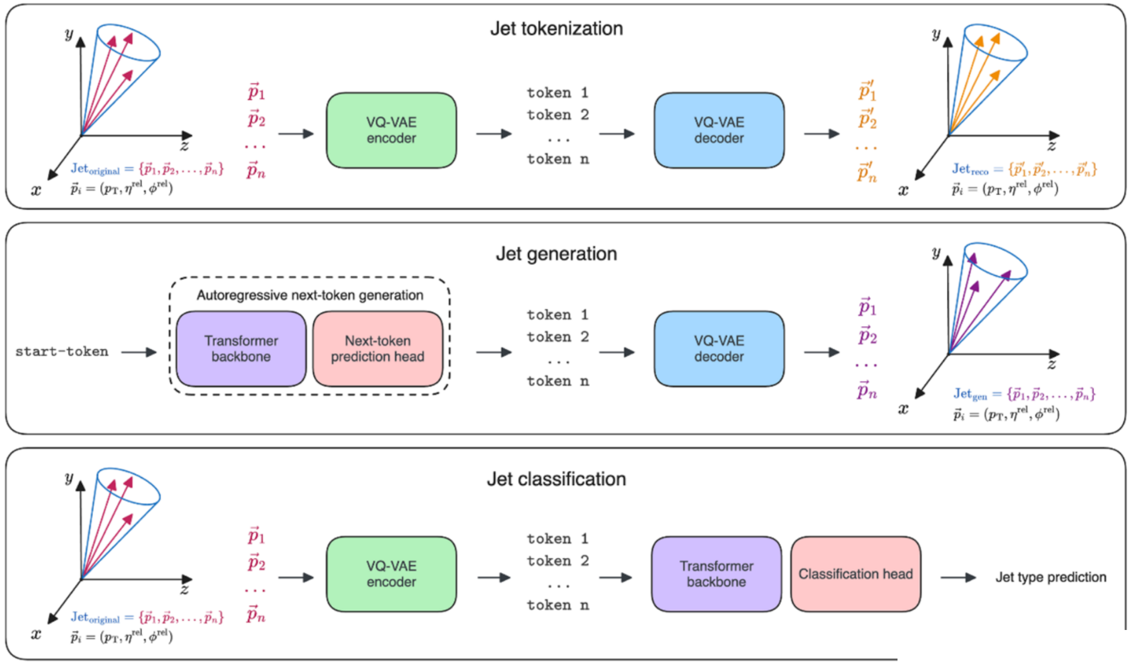
Audio
- Whisper

Multimodal
- GPT-4
- Claude 3
- Gemini
- Mistral

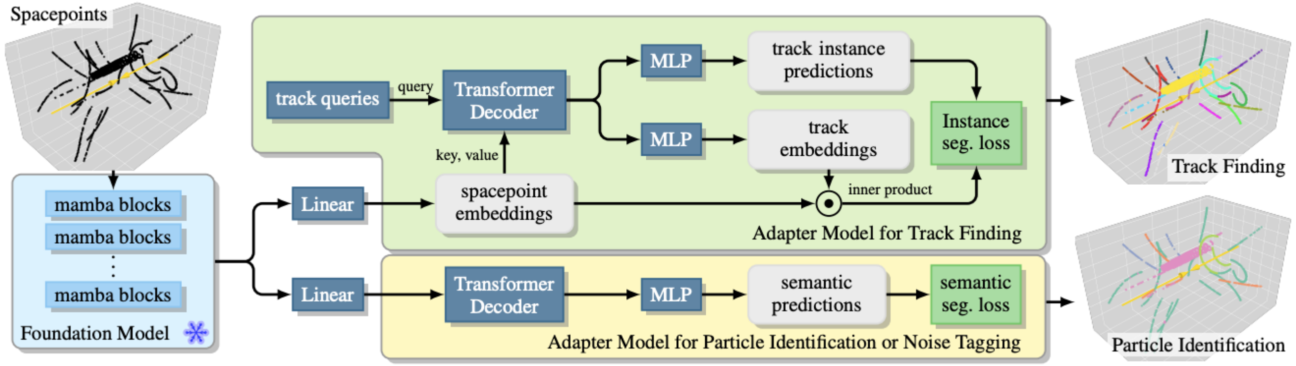
AlphaFold
Accelerating breakthroughs in biology with AI



Foundation models in particle physics



[OmniJet-α: the first cross-task foundation model for particle physics, J. Birk, 2024]



[FM4NPP: A Scaling Foundation Model for Nuclear and Particle Physics, D. Park, 2025]

New buzzwords

LLMS, Foundation models, GenAI, Agentic AI

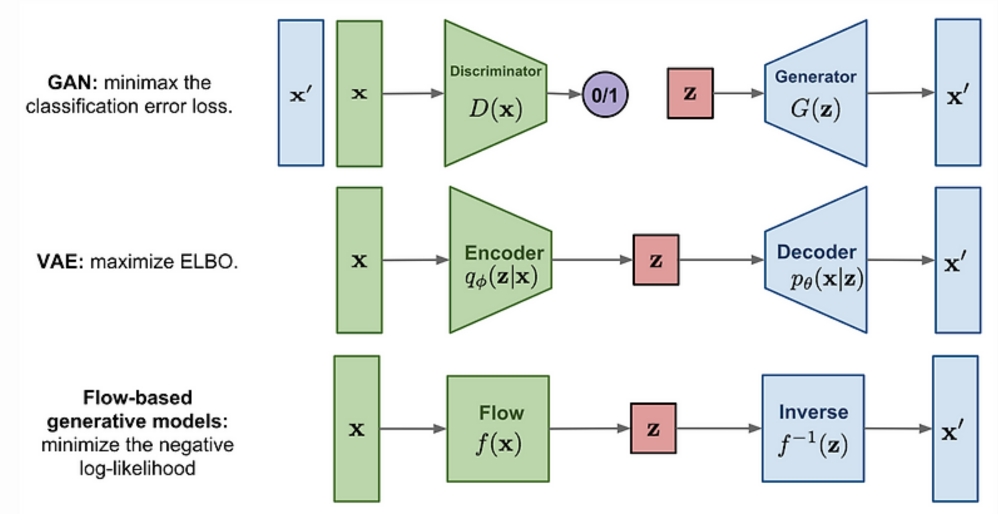
Generative AI: systems that generate content (capability level)

→ AI system whose primary function is to **generate new content resembling the training data** (text, images, audio, video, code)

Includes LLMs, diffusion models, audio and video generating models, any DNN developed for that purpose

GenAI is a capability category, not a model architecture or learning paradigm

- Generative adversarial networks (GANs)
- Variational autoencoders (VAEs)
- Flow-based models, normalizing flows
- Diffusion models



New buzzwords 🐝

LLMS, Foundation models, GenAI, Agentic AI

Agentic AI: AI systems focused on autonomous decision-making, designed to pursue goals through sequences of actions, typically by combining models with memory, tools, planning, and feedback loops.

→ LLM-driven planners with external control logic (LLM inference, heuristics for planning, external control logic, tool calling, search)

Agentic AI can be implemented with RL, but usually isn't

ChatGPT, perplexity: can decompose high-level goals into steps, decide to call external tools (code, web, file analysis), execute those actions, observe the results, and adapt subsequent steps until the task is completed.

GitHub Copilot: can analyse a repository, propose a multi-file change, apply edits, run tests, observe failures, and revise the solution.

Notion: can manage tasks, update documents, maintain project state, and take follow-up actions based on user instructions

Agentic systems in particle accelerators

LLMs and Agentic Interfaces

- Serve as operator interfaces
- Parse logs, discover correlations
- Act as collaborative agents during design
- Vision: 'LLM-based expert pools' for collective knowledge and feedback

Tuning Prompt

Human: Now you will help me optimise the horizontal and vertical position and size of an electron beam on a diagnostic screen in a particle accelerator.

You are able to control five magnets in the beam line. The magnets are called Q1, Q2, CV, Q3 and CH.

Q1, Q2 and Q3 are quadrupole magnets. You are controlling their k1 strength in m^-2. Their range is -30.0 to 30.0 m^-2.

CV is vertical steering magnet. You control its steering angle in mrad. Its range is -6.0 to 6.0 mrad.

CH is horizontal steering magnet. You control its steering angle in mrad. Its range is -6.0 to 6.0 mrad.

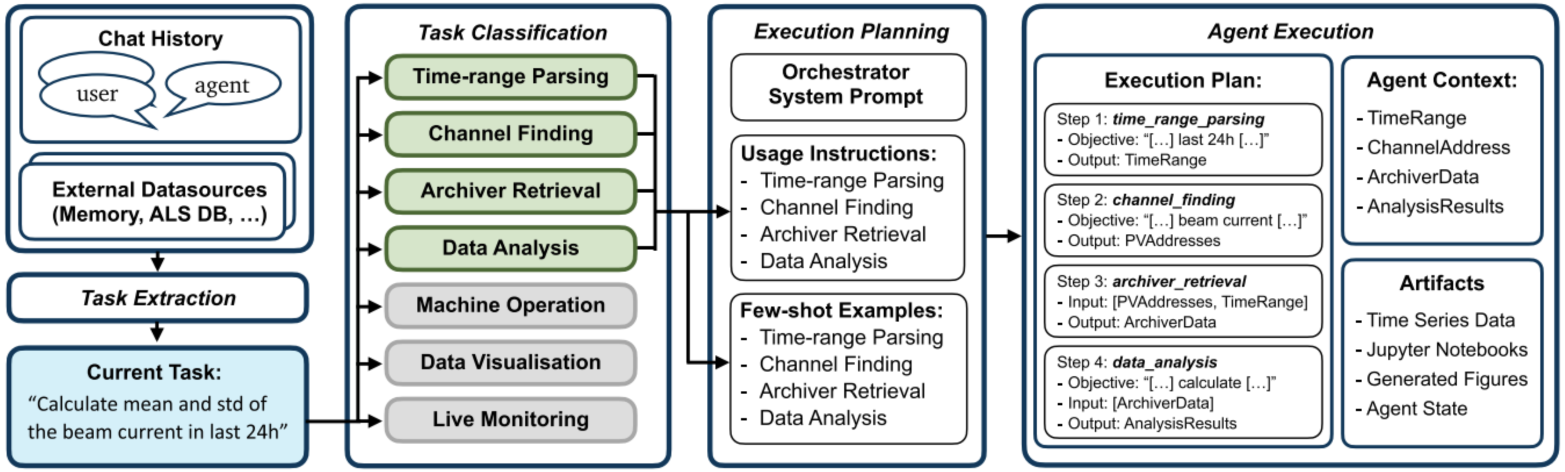
You are optimising four beam parameters: mu_x, sigma_x, mu_y, sigma_y. The beam parameters are measured in millimetres (mm). The target beam parameters are:

Target beam parameters:
```json
{
   "mu_x": 1.20,
   "sigma_x": 0.11,
   "mu_y": 1.25,
   "sigma_y": 0.06
}
```

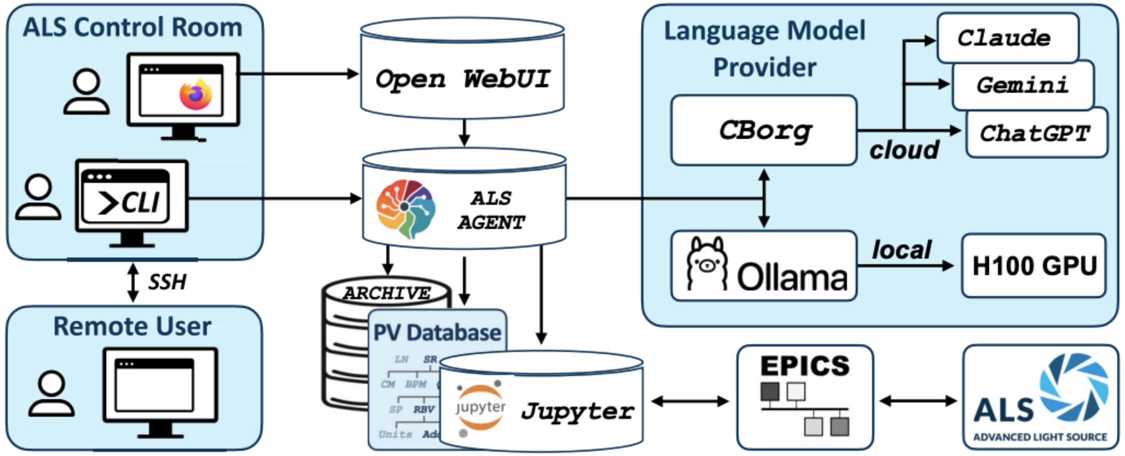
Task description

[Large language models for human-machine collaborative particle accelerator tuning through natural language, J. Kaiser, 2025]

Agentic systems in particle accelerators



Agentic workflow @ALS



System architecture of Accelerator Assistant @ALS

[Agentic artificial intelligence for multistage physics experiments at a large-scale user facility particle accelerator, T. Hellert, 2026]

THANK YOU FOR YOUR ATTENTION!

What questions do you have for me?

Dr. Andrea Santamaria Garcia
Lecturer at University of Liverpool
Cockcroft Institute

ansantam@liverpol.ac.uk

https://www.linkedin.com/in/ansantam/

https://github.com/ansantam

https://instagram.com/ansantam

https://www.liverpool.ac.uk/people/andrea-santamaria-garcia