Day 10: Current Challenges

Presenter: Auralee Edelen

Day 10

AI Safety

Major area of open research in AI/ML more broadly → some good techniques already well-established for accelerators

lower-consequence

Safe / constrained optimization applied in cases that are not really safety-critical

- Explore while avoiding simply undesirable of beam parameter space → *faster tuning*
- Avoiding regions a human wouldn't go to or by better juggling various needs
- **Faster/better tuning = more scientific output per dollar spent in operation**

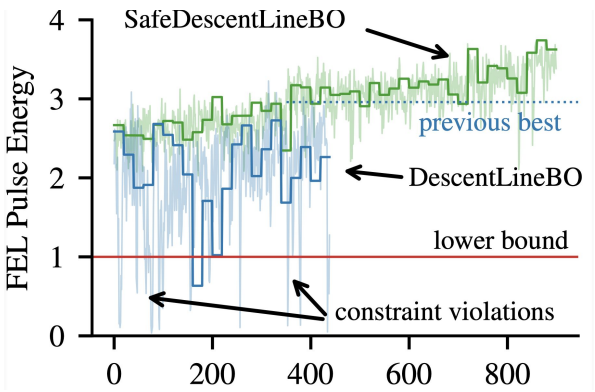
Damage to Instruments

- Radiation, heat, rf power, high power beams, etc. can damage components
- Can be very expensive (multiple millions of dollars) for a single component
- Can be indirect: e.g. a hole in a beam pipe can cause huge loss of time due to need to patch/replace and pump back down to vaccuum → *can also damage other equipment*
- **Machine protection system is not guaranteed to help** *(many examples where issues due to human error aren't caught by MPS → e.g. manual steering into beam pipe, not seeing sparking pattern during rf conditioning)*

Human Safety

- Accidents could have severe consequences in some cases (if future MPS systems use ML, it needs to be robust)
- Industrial/medical applications: proton therapy, x-ray therapy, accelerator-driven reactors

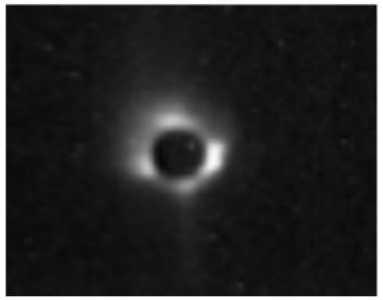
higher-consequence



J. Kirschner et al., ICML (2019)

pulse energy drops
→ *angry users!*

high beam losses
→ *radiation damage!*



beam halo

https://accelconf.web.cern.ch/IPAC10/papers/wepeb075.pdf

proton therapy machine at Children's Hospital of Philadelphia

Uncertainties and Time-Varying Behavior

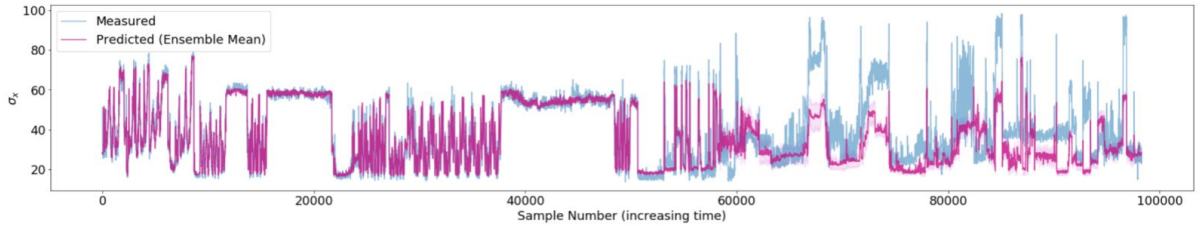
Major assumption in ML: i.i.d. (**independent and identically distributed)** distributions

→ ***How to handle out-of-distribution data and drift?*** *(major research question in AI/ML community)*

Accelerators are changing all the time

- Deliberate changes in state (e.g. new beam configurations)
- Unplanned changes in state (e.g. equipment failure)
- Drift (e.g. temperature/equipment), e.g. ion sources + cathodes can change responses over months
- Equipment changes + part replacement over time, not just “one accelerator”

Need accurate (calibrated!) uncertainty estimates + redundancy (e.g. multiple models + control methods)



Uncertainty estimates don’t always cover full error
In this case: substantial drift in several inputs to OOD

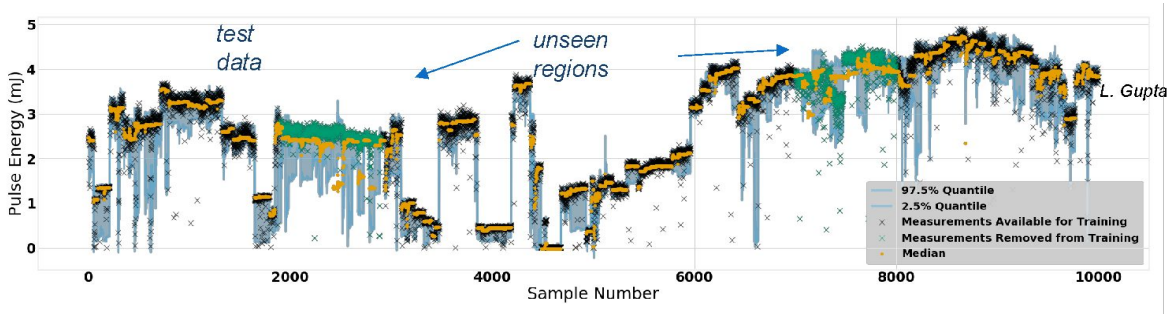
Need to be able to handle out-of-distribution samples and drift over time

- Identify when model is no longer accurate *(e.g. can monitor accuracy + estimated uncertainty, look at ranges of inputs to see if beyond training data, etc.)*
- Need good / robust procedures for re-training or re-calibration *(many promising options but not well-explored)*
- Switch between methods inherently better for new regimes *(e.g. handoff between data-hungry ML methods and ones better-suited for efficient exploration)*
- Automation of all this is a major challenge

Data Cleaning + Data Tagging

We have a lot of data, but we need to be able to make use of it!
GIGO

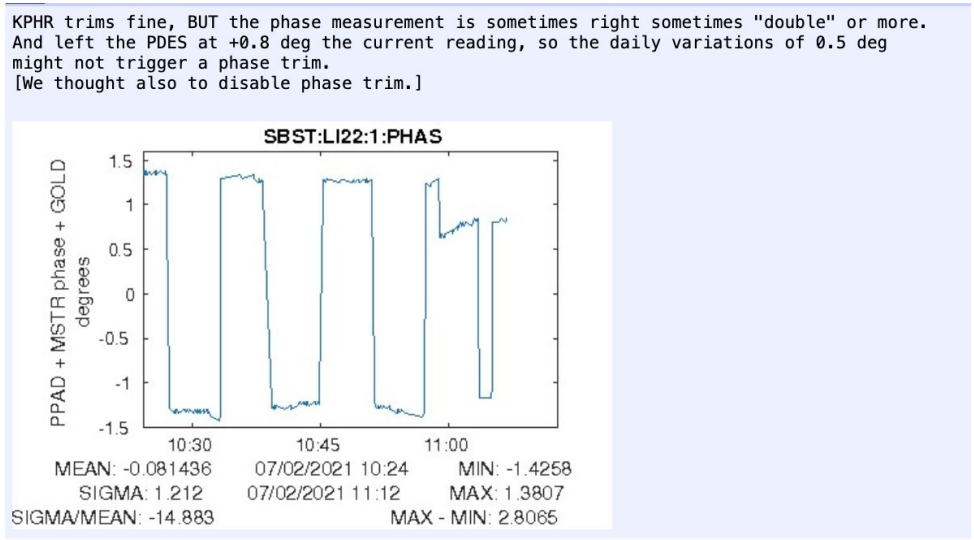
- Many variables → which to use? *"include them all" can backfire*
- Changing conditions *(e.g. different machine states, some of which are sub-optimal or have different metrics)*
- Not all variables are logged (often added over time) or capable of being logged
- Different archives with different timing systems (e.g. sometimes not even a centralized clock)



hundreds of possible inputs
→ many operating modes + changes to machine!

Disparate sources of info

- E-log for machine, e-log for maintenance
- The data itself
- Unlogged information → *e.g. hardware change details, calibration changes not always logged (especially during commissioning)*


KPHR trims fine, BUT the phase measurement is sometimes right sometimes "double" or more.
And left the PDES at +0.8 deg the current reading, so the daily variations of 0.5 deg
might not trigger a phase trim.
[We thought also to disable phase trim.]

e-log entry

ML can help but should probably not be relied upon exclusively

- Anomaly detection
- Feature selection
- NLP for the logbook??? ☺️

Legacy Software and Hardware

(compute + data acquisition + accelerator components)

Data recording rate (and machine rep rate) + storage capabilities

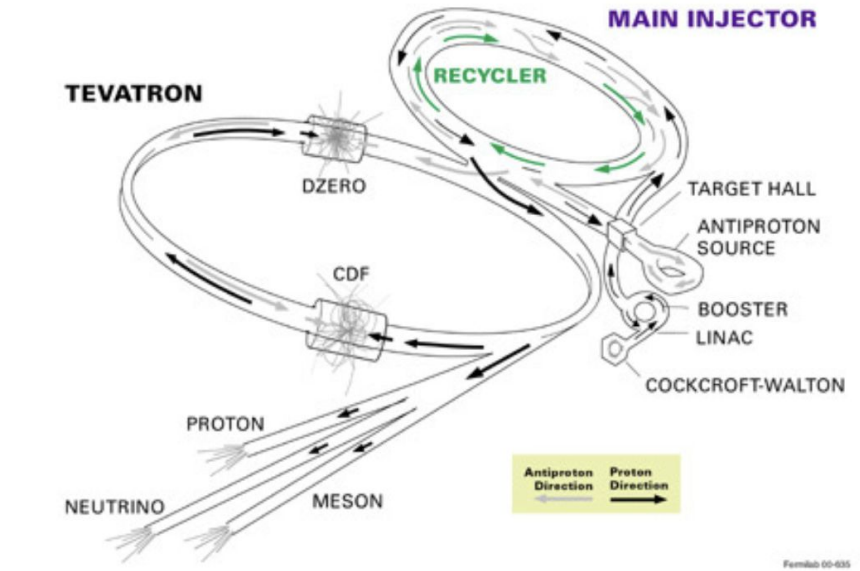
Matters a lot for data collection + feedback! *(think about 120Hz vs. 1Hz data)*
e.g. upgrade cameras 1 Hz to 10 Hz → could require change of hardware!
e.g. RF waveform data → MHz rate, difficult to store all data!

Some accelerators are very old!

Expensive to build + construction takes a long time
→ not using cutting-edge for data acquisition/storage/compute at build time

e.g. Fermilab Tevatron
- Started construction in 1968; operational 1987 – 2011
- Some components originally for Tevatron are still used

e.g. LCLS began operation in 2009
- Both LCLS and FACET-II use some components from an older accelerator at SLAC (e.g. RF cavities)



Compute resources + controls network

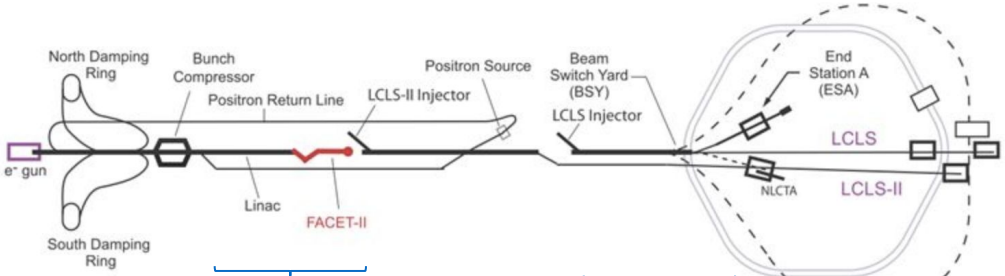
Old infrastructure + bandwidth between HPC compute resources and accelerator

Often have safety restrictions on entering/exiting controls network

Flexibility to change varies between facilities

High-demand user program → *risk/reward favors being conservative about change*

New accelerators (e.g. CLARA at Daresbury) inherently trying to support ML/AI data acquisition and storage needs in control system and data acquisition design



FACET-II accelerator

LCLS accelerator

Software Development Practices

All-too-common situation in accelerators: un-tracked code / lattices etc sent over email, copied into folders, local changes

→ *development nightmare*

Essential for stable development to track changes to code → github etc!

- Useful even if not working on community code (e.g. what changes did I make 6 months ago?) → *important for reproducibility*
- Many journals now require data or code to be available
- Some resources exist for publishing data sets

Need for community software and open-source code development

- Open-source code and data sets are what helped ML/AI research take off → we can improve how we do this in accelerators
- Also benefits to sharing data → standard ML test problems in accelerators

Reinventing the wheel often wastes resources

- Sometimes starting from scratch is needed (e.g. too old or convoluted code base, new needs to meet) → *not talking about those cases*
- **Easy to reinvent the wheel accidently if not communicating**
- Biggest challenges for community code: different design philosophy + different points of focus / needs + organizational barriers

Even small-scale code sharing and using repositories for individually-developed projects is helpful to accelerate progress and ensure reproducibility

Combining Physics + ML

We have good physics models

→ *should not discard all that info!*

→ *adding physics info can improve generalization and sample-efficiency in learning (need less data)*

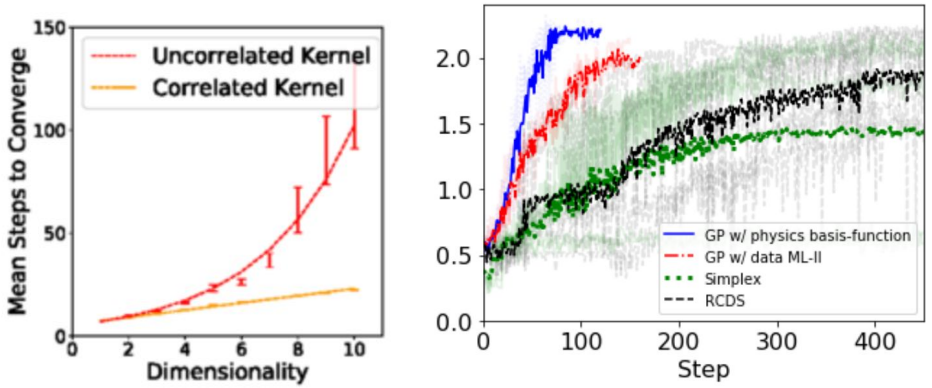
Various ways to combine theory + sim with measured data:

- Calibrated physics models
- Calibrated ML models
- Adding physics info to cost function
- Inductive biases
- Physics-based priors

→ *Saw examples of some of these in the class*

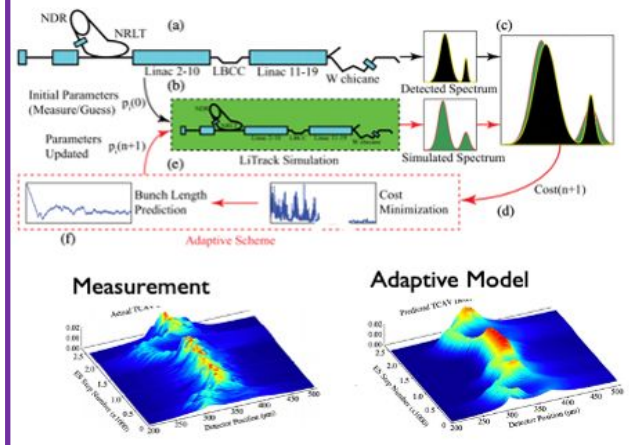
Open area of research in ML more generally

More efficient online optimization with Bayesian optimization and physics-informed ML



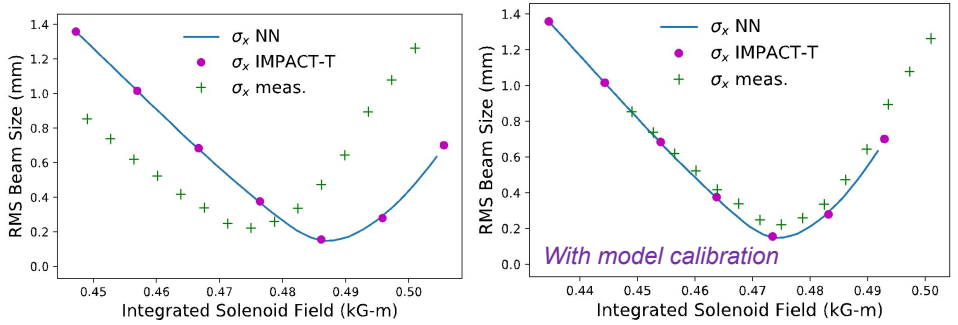
J. Duris, et al, PRL 124, 124801 (2020)
A. Hanuka, et al, PRAB 24, 072802 (2021)

Adaptively tuned physics model

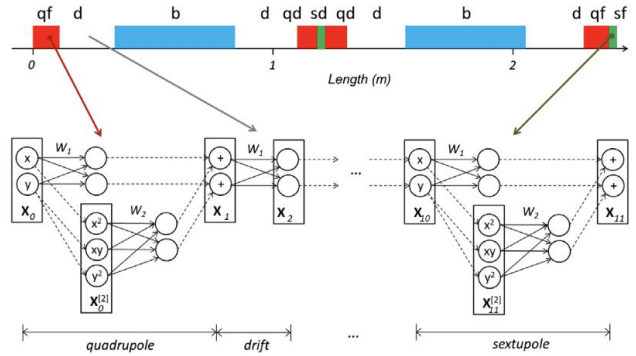


A. Scheinker, S. Gessner, PRAB 18, 102801 (2015)

Constrained calibration of standard neural network



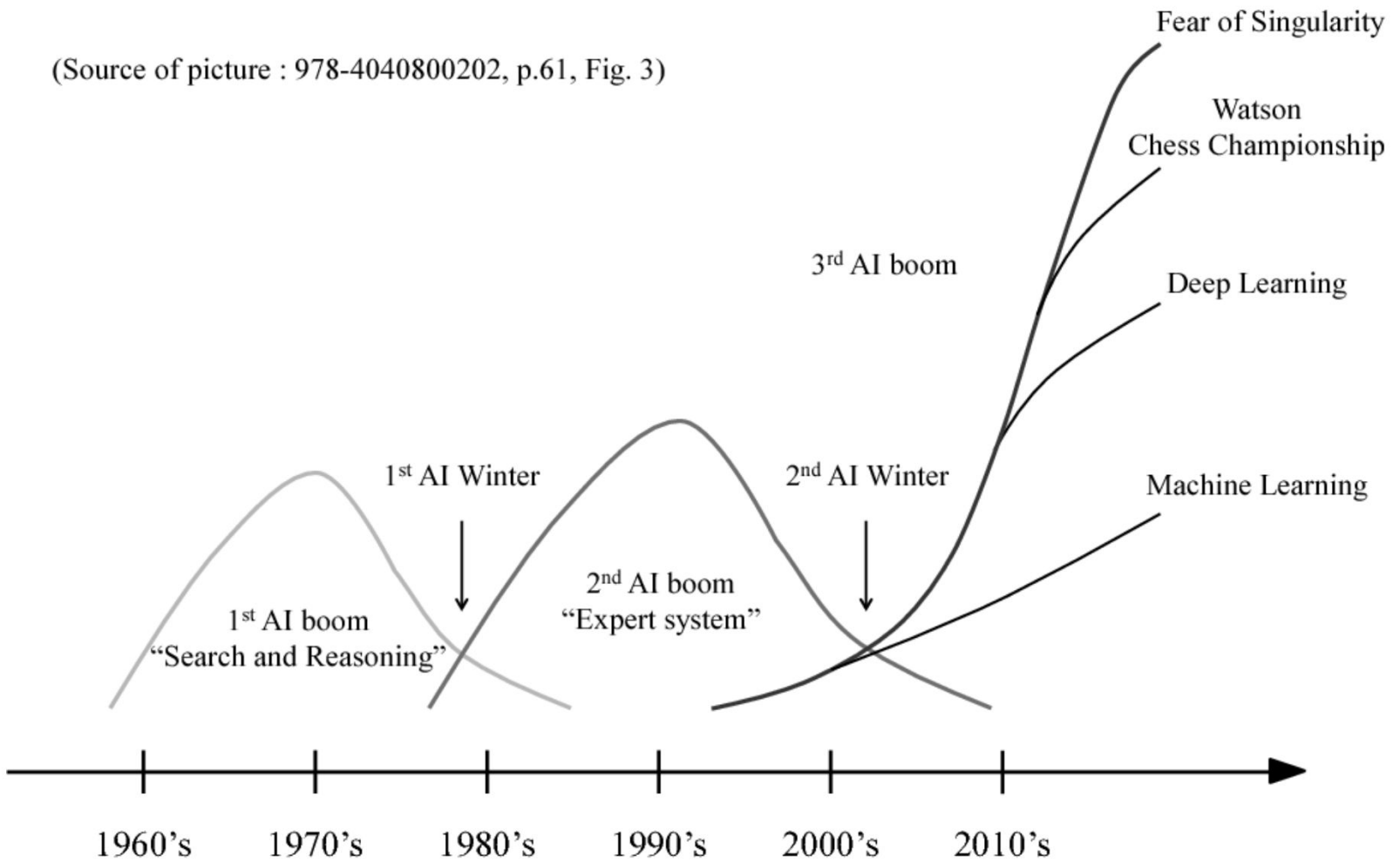
Included physics structure (e.g. Lie map) + autodiff for calibration



Andrei Ivanov, et al., PRAB 23, 074601 (2020)

History of Neural Networks and “AI winters”

- 1950s - 1960s: reasoning, search etc
- 1970s: AI winter
- 1980s: “connectionism” i.e. neural networks, knowledge representation
- 1990s: AI winter
- 1997: Deep Blue beats Gary Kasparov in chess
- 2006: Deep learning breakthroughs at University of Toronto
- 2011: IBM Watson wins Jeapordy
- 2015: Deep learning on GPUs
- 2016: Alpha-Go deep learning software beats best players



History of AI/ML in Particle Accelerators

Excitement about new areas is good, but it can backfire due to hype

ML for accelerators had brief popularity in the 1990s then died out again

e.g. can see in publication history for in accelerators (not exhaustive):

Early 1990s → excitement + a lot of early studies with mixed results, then mostly abandoned

2008 – 2010 → NN studies at LCLS and Australian Synchrotron

2015 – 2017 → NN studies at Fermilab, Bayesian optimization and RL at SLAC (+DESY)

2018 → First ICFA ML workshop on ML for particle accelerators → *international community that was starting to work on AI/ML came together to discuss opportunities and challenges, resulting in a community white paper: https://arxiv.org/pdf/1811.03172.pdf*

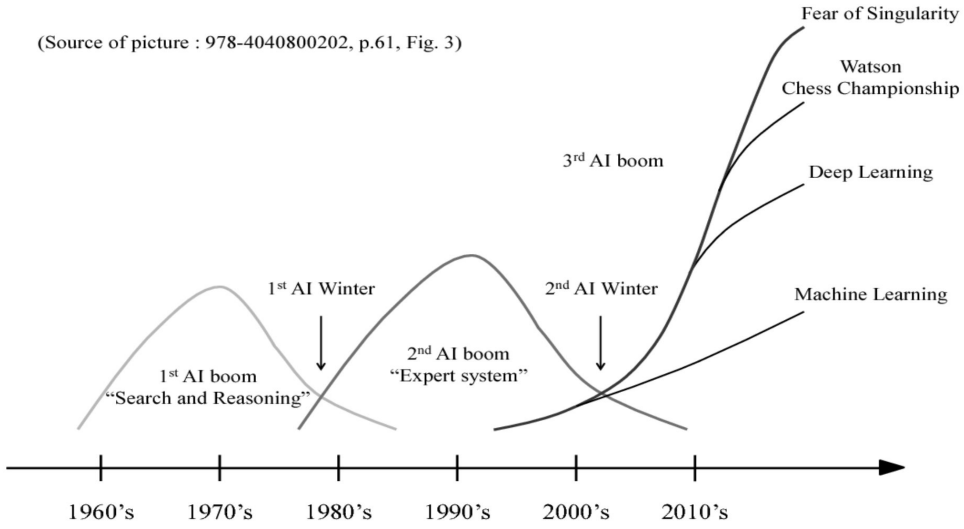
***2018 onward** → several publications each year (and growing)*

2018 - First ICFA workshop on ML for Particle Accelerators

- **Mid 1990s - appx. 2017 accelerator field was generally very skeptical of ML**
- **2017 was a major turning point as attitudes started to shift quickly**

  → *2019 DOE started getting ready for major funding initiative*

- **Great news, but also reason for caution:**
  - *First-hand accounts of people working in AI in the 80s and 90s describe how backlash after hype and aggressive funding drove the previous AI winter*
    *→ **opportunity for us to learn from this***
  - *Bringing solutions into regular operation in addition to doing cutting edge AI/ML R&D **might** help avoid the same issues*



Future Needs and Directions for ML/AI in Accelerators

Investments for Open Technical Challenges

Uncertainty quantification

- Detect when model may not be accurate (e.g. outside training range)
- Leverage for safe exploration of parameter space

Active learning

- Retraining to account for drift or adapt during search
- Sampling strategies to efficiently explore large parameter space + generate training data (maximize information with the least samples)

Efficient ways to handle high dimensional data:

- Images, point cloud, etc
- More variables (full accelerator vs. small test cases)

Physics-informed / constrained ML

- Improve robustness / generalization to unseen regions of parameter space
- Reduce need for additional data
- Extract physics from measured data

Interpretability

- Important for ML-based tuning, anomaly detection, modeling

Many shared challenges with other SciML domains

Investment in Shared Infrastructure

- Common data format standards (e.g. OpenPMD)
- Open software infrastructure for automation (e.g. Ocelot, xopt)
- Library of benchmark datasets + solutions

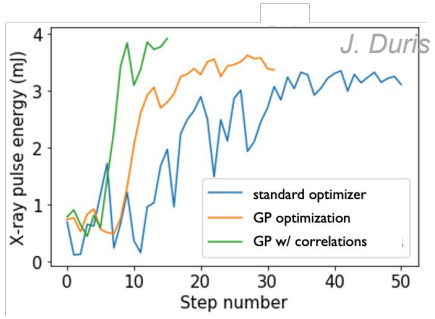
Cross-Communication and Coordination

- Tighter coupling between accelerator theorists and those doing ML
- Tighter integration with broader physics/ML community (e.g. those in particle / plasma physics doing ML)

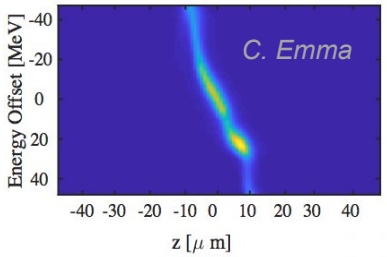
- Collaboration between facilities
  - Many shared challenges and similar problem structures for ML in accelerators
  - ML solutions can be readily transferred between facilities
  - Need funding structures that better facilitate or directly encourage cross-collaboration
  - *e.g. FACET-II / LCLS Cu very similar designs / needs with respect to ML-based tuning*

All applications will be tied together in the end for operations
→ need resources and robust approaches to do this

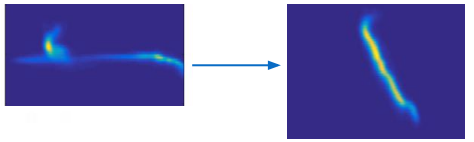
automated control
+ optimization

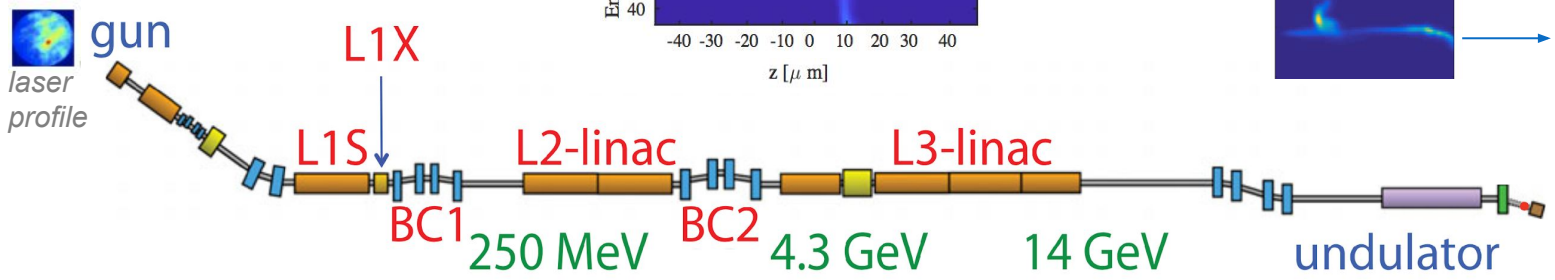


advanced diagnostics
(reconstruct / analyze beam)

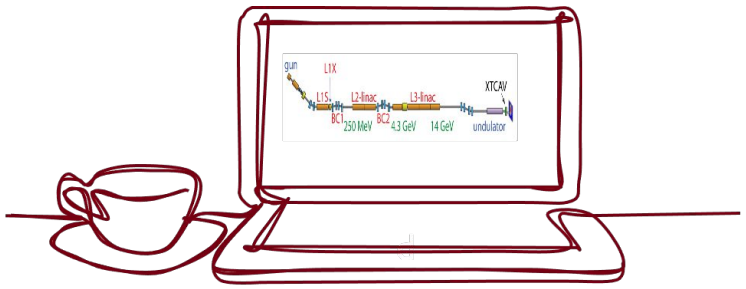


anomaly detection
failure prediction





incorporate
physics
information



extract unexpected
relationships
(feed into control / design)

digital twins + online modeling
(fast sims, autodiff sims, model calibration)

+ need UQ for all