

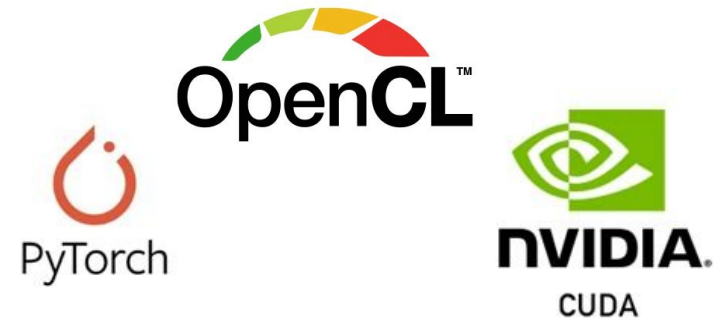
GPUs and AI in Particle Physics

Mark Slater

UNIVERSITY OF
BIRMINGHAM

Overview

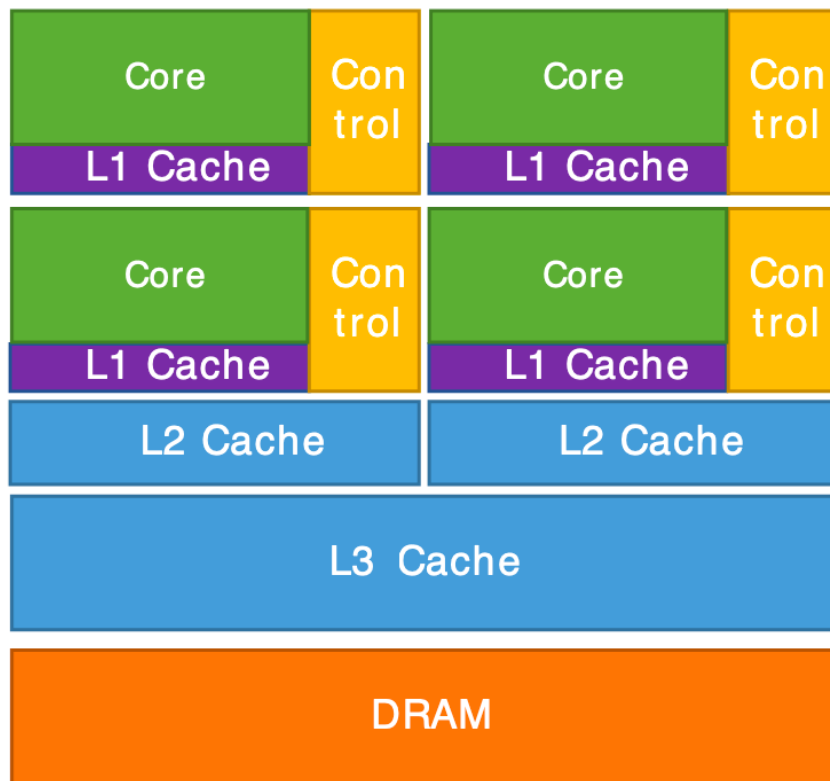
1. What are GPUs
2. Common Applications
3. Coding and Tools
4. AI/Large Language Models
5. Current GPU/AI use in PP
6. Resources available
7. Leveraging Options



What are GPUs?

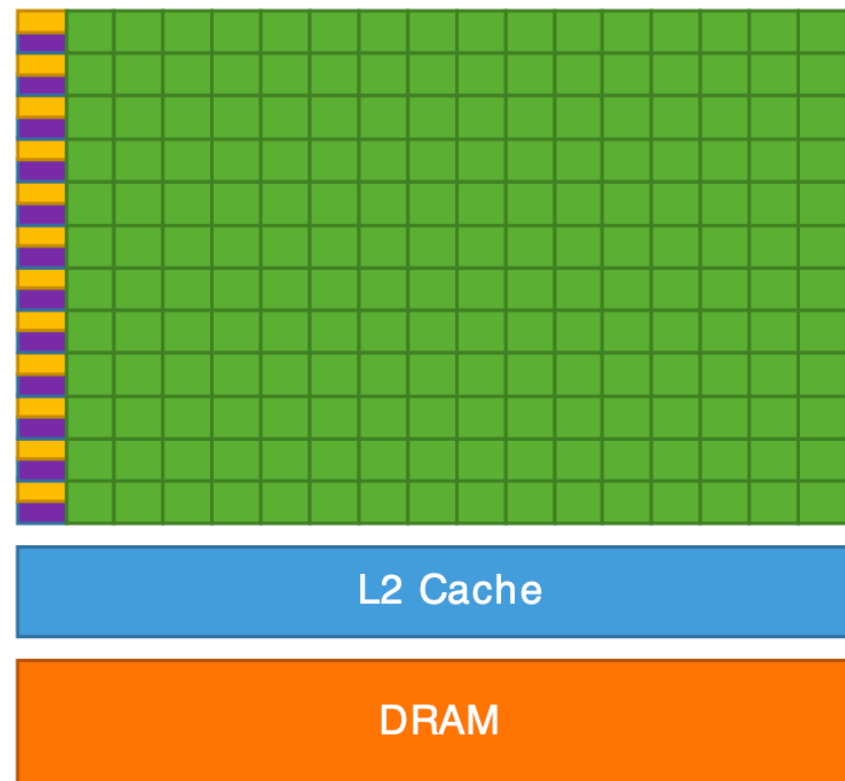
- The need for **dedicated graphics hardware** dates back to the 1960s but increased significantly with the **rise of video gaming** in the 1980s
- During the 1990s, **separate 3D hardware accelerator cards/co-processors** became essential to support the requirements of gaming at the time
- As 3D and graphics acceleration is an embarrassingly parallel problem, these cards moved towards a common architecture in the late 1990s:
 - **Many parallel cores** that run a reduced/specialised instruction set
 - Designed to run the **same code over many 'fragments'** of data
 - **Dedicated memory** with fast links to the cores
- From early 2000s onwards, these started to be used for similarly parallel computing problems and the **first dedicated 'General Purpose' GPUs** started to be manufactured

Differences in Architectures



CPU

- Few but powerful cores
- Single cores operate (mostly) independently
- High clock speed for faster sequential processing
- Significant Cache memory
- Broad instruction set to cater for more tasks



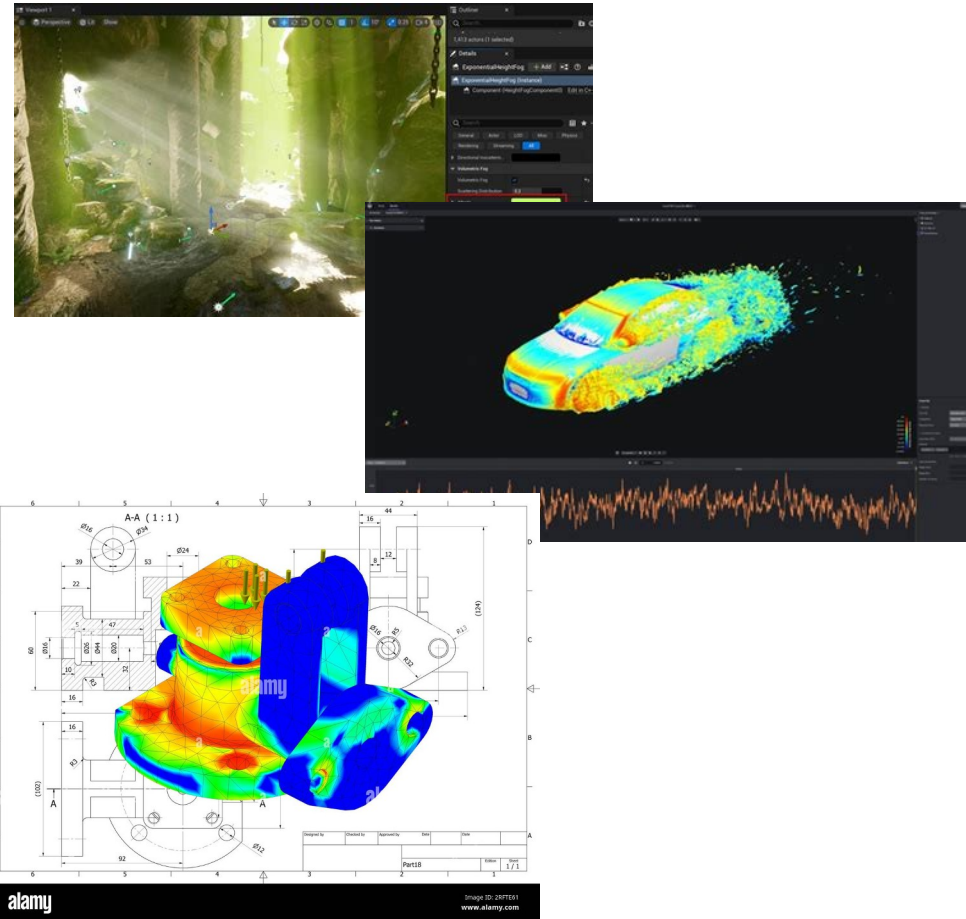
GPU

- Large (>1K) of smaller, less powerful cores
- Multiple cores grouped together in 'warps'
- Lower clock speed
- High memory bandwidth to/from on-board memory
- Specialised instruction set for parallelisation

Common Applications

- Due to their **highly parallelised architecture**, GPUs are best suited to problems with **many independent (but similar) calculations** over lots of small pieces of data such as:

- Graphics processing
- Climate simulation
- Fluid Dynamics
- Simulations of Physics Processes



- Any complex problem that can be broken down into small independent parts **has the potential** to benefit from GPU use

Low Level Coding for GPUs

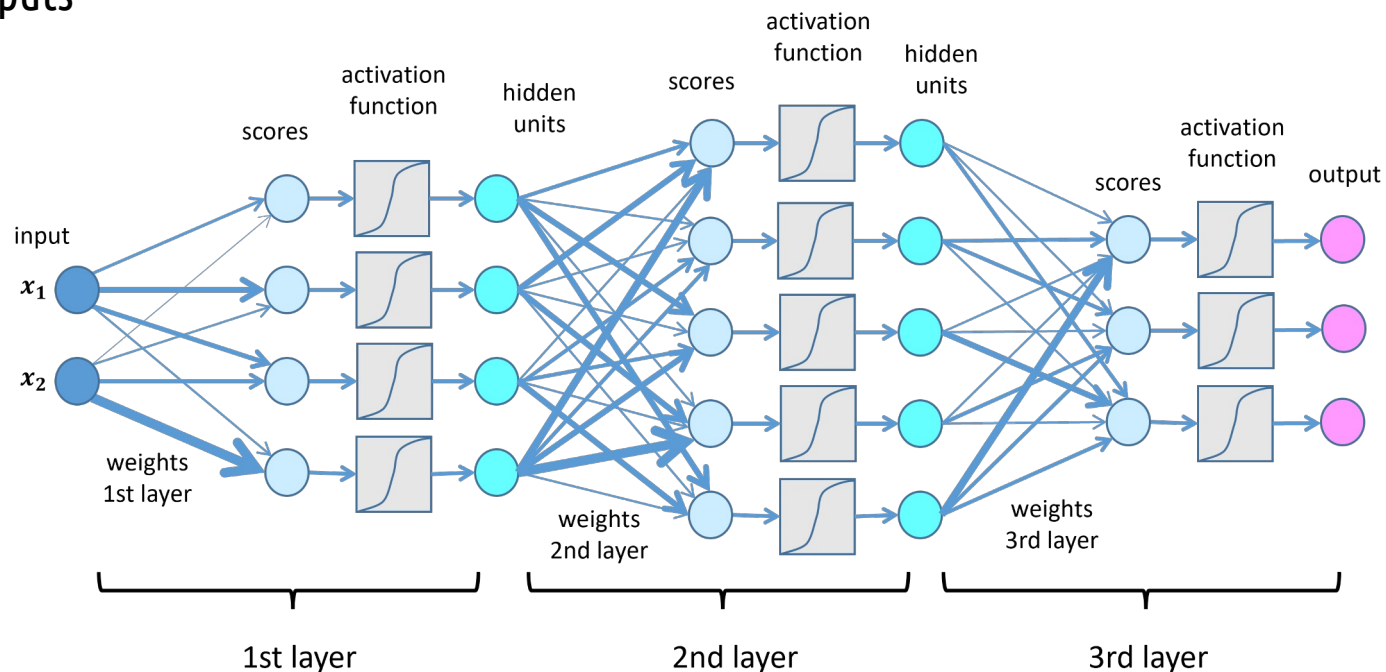
- At present, the most popular tool for low level control of GPUs is **nVidia's CUDA framework** though other options are available (OpenCL, AMD ROCm, etc.)
- When directly coding for a GPU, you need to consider:
 - **Memory management** to minimise transfer between host and GPU
 - Optimising **GPU memory access** (e.g. structs-of-arrays rather than arrays-of-structs)
 - Very **restricted coding patterns** (basically 'C' rather than 'C++')
 - Minimising **branching code**
- These restrictions mean some (most?) problems **aren't well suited to GPUs**
- These cases may be better off becoming **traditionally multi-threaded** to take advantage of high CPU core count
- Note: Even if the whole algorithm can't be ported to GPUs, **specific parts may still benefit!**
- A good overview:



https://indico.cern.ch/event/1366450/attachments/2832474/4952176/ATLAS_Lecture_Series_GPU_Beomki_Yeo.pdf

Machine Learning and Neural Networks

- One of the main uses of GPUs in modern times is with **neural networks and machine learning**. These are critical for:
 - Image recognition and other Classifiers
 - Natural language processing
 - Large Language Models and Generative AI
- These all work using similar interconnected **'layers' of neurons** in various configurations
- Weights within the layers are **altered during 'training' of the network** to give the desired outputs from a set of given inputs



ML Frameworks and Tools

- The main big tech firms offer various **machine learning tools**, quite often including a **platform/cloud component** as well
- **Python is the most popular** (though not exclusive) language for working neural networks with several modules available:

→ TensorFlow

→ PyTorch

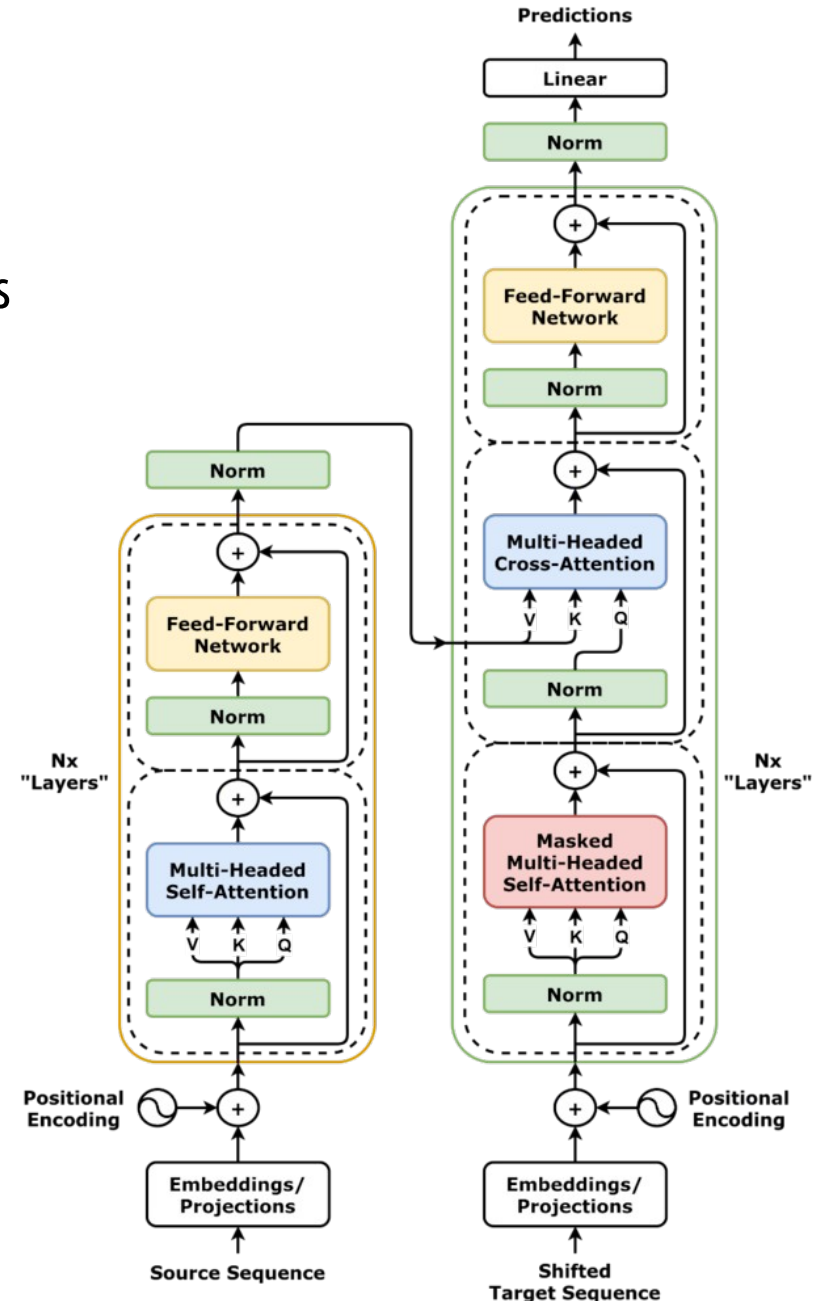
→ scikit_learn



- Root also contains an ML module – **TMVA** – which can be used in isolation or with the other tools
- With these tools, you can control the layers, activation functions, optimisers, etc. to **build, train and evaluate your NN**
- You will still need to carefully select a **large amount of appropriate training data** though!

Large Language Models and AI

- These days, due to the rise of ChatGPT, CoPilot, etc., when people talk about 'AI' they usually mean **generative Large Language Model systems (LLMs)**
- These are a specific type of deep learning NN that has **billions of neurons and many (96+) 'Transformer' style layers**
- They rely on training:
 - **a representative vector space** of the type of input data
 - **The transformer layers** within the model itself
- Each Transformer layer consists of:
 - An Attention step that **links words in the input together**
 - A feed forward layer that **applies relational information** encoded in the vector representation



Current Applications for AI/LLMs

- Due to the (very) large investment in LLM-based AI, **it has been pushed heavily** into both people's private and professional lives
- The **generalised LLM approach** has been applied to many applications:
 - Image Classification (e.g. Looking at scans for cancer or eye problems)
 - Scribe tools for automatic transcription of meetings
 - Chatbots for websites to deal with queries
 - Code generation, either from scratch or as bug testing/validation
 - Image, video and audio generation from prompts
 - Agents to control related aspects of a system autonomously
- Generally speaking, any problem that has a large dataset enumerating acceptable 'solutions' to it (in text, video and/or audio form) **MIGHT benefit from LLM development**
- However, there are several **potential drawbacks** such as cost of training, availability of good data, stochastic nature of the models, 'hallucinations', validation of outputs, etc.
- This can lead to **minimal (if any) improvement** over traditional methods for some applications

PP Applications Involving GPUs and AI

- Neural Networks have been used in PP for decades now for **Particle ID and event classification**
- GPUs have also been increasingly used in **high level triggering** and **online reconstruction** across the LHC experiments
- **ATLAS** seems to have made most progress on using GPUs outside of HLT:
 - Athena on GPUs
 - Celeritas – GPU simulation of EM interactions
 - Tracc - GPU Track reconstruction based off ACTS (<https://acts.readthedocs.io/en/latest/>)
 - Clustering algorithms
- CMS is also moving forward with studies on doing **event reconstruction using machine learning**
- Closer to home, I extended **Garfield++ to use GPUs** providing in ~50-100x speed up
- For LLMs, there is **not much dedicated use** at present. There are two studies I found:
 - **Knowledge retrieval** from CERN pages/docs using NLP
 - Trialing **Paper reviews** in CMS– do first pass with LLM
- The majority of use seems to be for **'private' code generation/review**

Group Facilities Available

- In the group, we have 3 options for GPU access:
 - epldt001 – dual P100 cards (quite old now)
 - epldt003 – single A100 card (pretty good)
 - epldt004 – single NVIDIA L40S (powerful GPU. To be installed after Easter)
- These all have CUDA installed and are setup as a normal desktop/login node
- They should provide a good testbed for small to medium GPU applications
- Used by myslef and Tom Neep to develop Garfield++ GPU extension for shower generation

University Facilities Available

- The university provides a number of services under the 'Birmingham Environment for Academic Research' (BEAR) banner
- Primarily, there is the 'Bluebear' computing cluster featuring a total of 20000 cores and 40 GPUs
- This is relatively easy to start using by signing up for access through the IT Service Desk
- Additional BEAR services include:
 - Research Data Store (5TB per project, more possible on request)
 - Gitlab instance (used for our website – see me to get group projects on there)
 - Cloud VMs instances available
- Direct dev support is available through the Advanced Research Computing (ARC) team
- Training is also available: <https://preview-uob.cloud.contensis.com/research/arc/bear/training>



National Facilities Available

- There are a number of 'Tier 2' facilities in the UK that are (in theory) available for use:

- Archer2



- Isambard

- Sulis

- Dirac

GW4

DiRAC



- For a full list, see <https://www.hpc-uk.ac.uk/facilities/>
- Generally, the bar to get access is higher requiring detailed proposals, etc. but the number of resources is high
- Locally, UoB has just been awarded £18M to host a new centre (Baskerville) that should be online next year with a focus on GPUs:
 - <https://www.birmingham.ac.uk/news/2026/university-of-birmingham-to-host-national-computing-centre>
- In addition, there are 3 other NCRs planned in Cambridge, Edinburgh and UCL:
 - <https://www.ukri.org/news/new-national-computing-resources-to-open-doors-for-researchers/>

Thoughts and Speculation

- Overall, **PP has leveraged these technologies well** where they have a clear advantage (PID, Analysis and Online Triggering) over traditional methods
- For general GPU use, I suspect there are **more opportunities available** that are yet to be exploited
- For LLMs, I think **the benefits are less clear** due to their non-deterministic nature
- Even being used as **tools for coding requires thought and care** rather than blind application
- Having said that, this shouldn't stop us from using the 'AI bubble' to **seek out additional funding streams**
- You can easily (and genuinely!) badge anything involving **Neural Networks as AI** to help this process!

Shameless Plug

- There is a new Linux Community on MS Engage (I don't know what that is) aimed at:
 - Bringing Linux users in the Uni together
 - Sharing of ideas
 - Promoting Linux in general
 - Pushing for better Linux support by the Uni
- If you're interested, please join!
- Go to <https://engage.cloud.microsoft/main/feed>
 - Login with Bham credentials
 - Search → Linux Community
 - Join