DIANA HEP

AB meeting - NYU

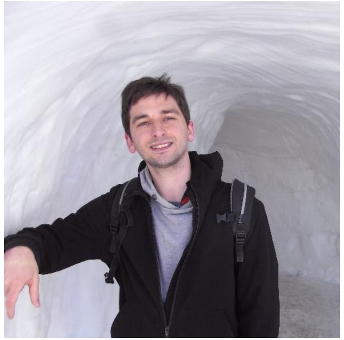
Gilles Louppe, Lukas Heinrich, Kyle Cranmer

January 26, 2017

NYU

Kyle Cranmer

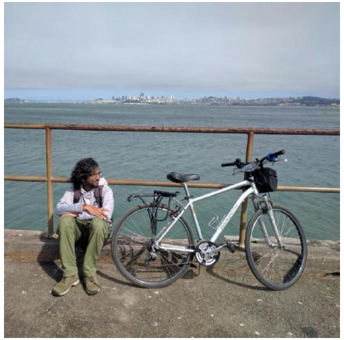
Gilles Louppe



Lukas Heinrich

Juan Pavez

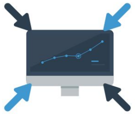
Cyril Becot



Manoj Kumar

Software for high-level analysis

- Integrating statistics and machine learning
- Reproducible workflows
- Scalability and modernization of statistical tools



Collaborative Analyses

Establish infrastructure for a higher-level of collaborative analysis, building on the successful patterns used for the Higgs boson discovery and enabling a deeper communication between the theoretical community and the experimental community

Reproducible Analyses

Streamline efforts associated to reproducibility, analysis preservation, and data preservation by making these native concepts in the tools

Interoperability

Improve the interoperability of HEP tools with the larger scientific software ecosystem, incorporating best practices and algorithms from other disciplines into HEP

Faster Processing

Increase the CPU and IO performance needed to reduce the iteration time so crucial to exploring new ideas

Better Software

Develop software to effectively exploit emerging many- and multi-core hardware.
Promote the concept of software as a research product.

Training

Provide training for students in all of our core research topics.

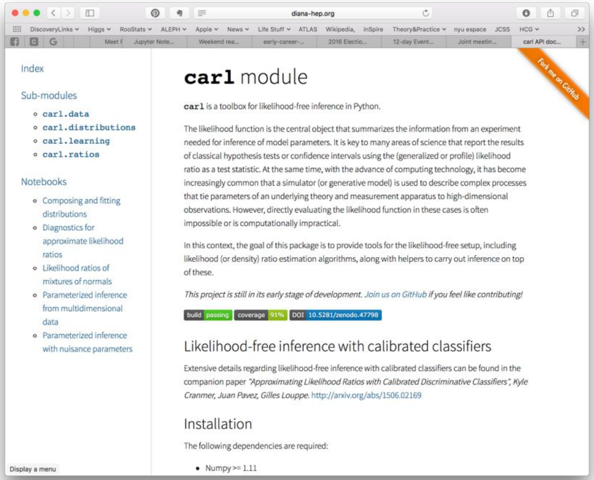
Likelihood-free inference

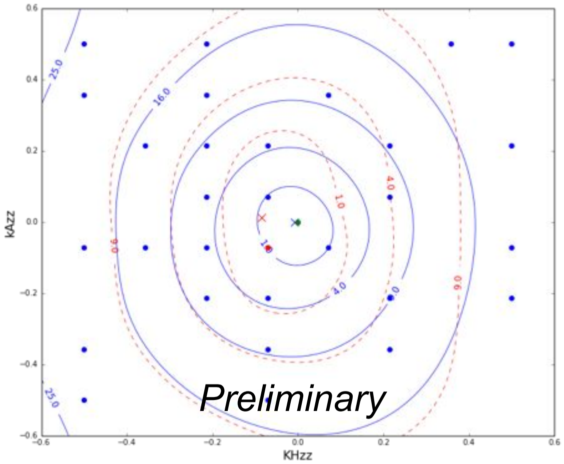
How to do inference when the model is written as a simulator? Can we use modern machine learning?

- Theoretical results arXiv:1506.02169 [Kyle Cranmer, Juan Pavez, Gilles Louppe]

- Carl: a likelihood-free inference toolbox [Kyle Cranmer, Juan Pavez, Gilles Louppe]

- (Ongoing work) Application to the Higgs to approximate likelihoods using kinematic information parameterized in coefficients of quantum field theory. *Promising data efficiency!* [Kyle Cranmer, Juan Pavez, Gilles Louppe, Cyril Becot, Lukas Heinrich]





A testbed for likelihood-free inference

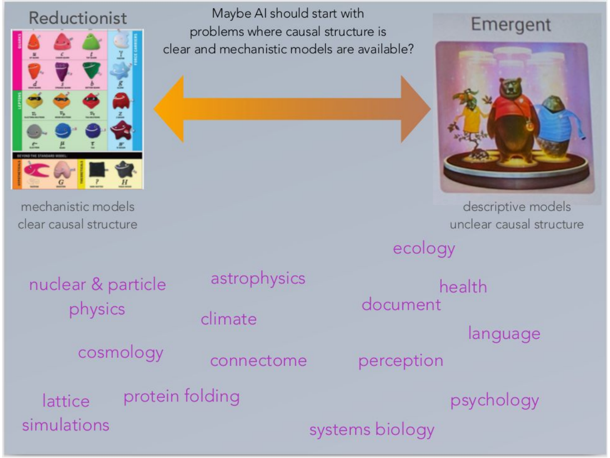
Particle physics is one of the few fields of Science where causal structures are clear and mechanistic models are available.

- This is an ideal testbed for the development of simulator-based likelihood-free inference.

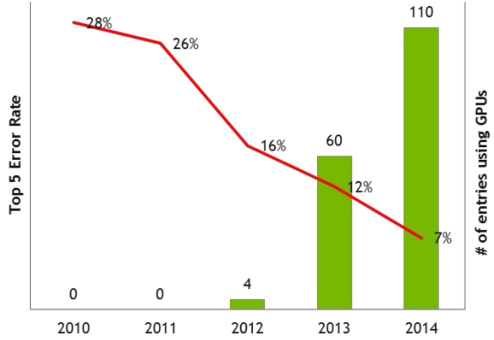
Artificial intelligence has the **potential to revolutionize** the physics analysis pipeline, provided we allow outsiders to run our tools easily.

- Simulator-based likelihood-free inference is a research problem the AI community is increasingly interested in (see NIPS 2016).

Project. Join forces with AI: Package the simulation software and design benchmark problems to trigger the development of new likelihood-free inference methods *with* the AI/ML community. [Connection to Lukas Heinrich



IMAGENET



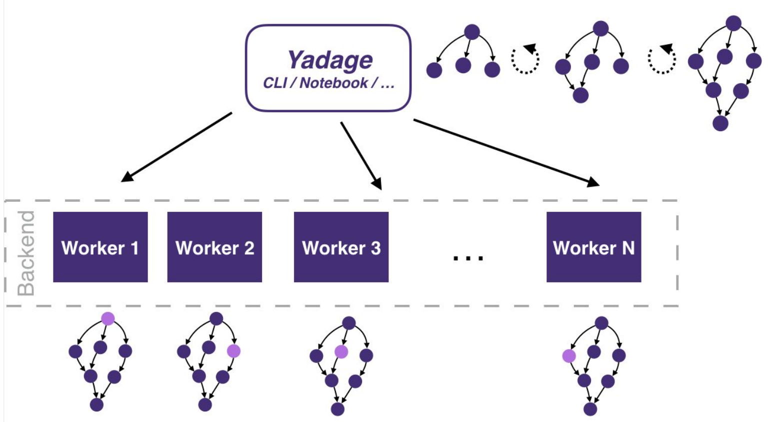
An ImageNet for particle physics?

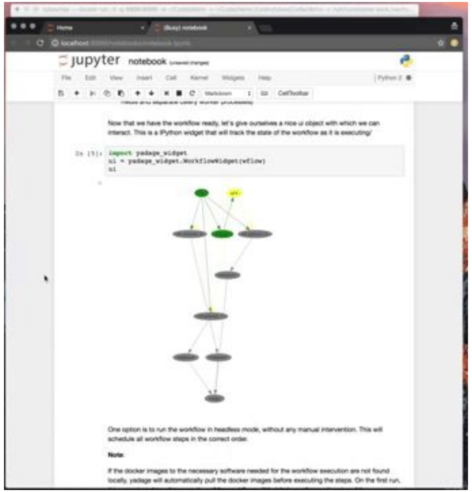
Reproducible workflows

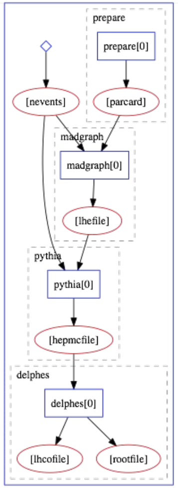
[Lukas Heinrich, Kyle Cranmer]

Collection of schemas and tools for declarative parametrized workflows, where each step describes its own s/w environment (complete capture with e.g. containers) [repo]

- Graph-based distributed computing on e.g. Kubernetes/Swarm/… clusters.
- Declarative definitions via pure YAML / JSON
- Funded by NSF via extension to DASPOS, supported by NYU MS-DSE
- Being deployed at CERN and integrated into CERN Analysis Preservation - CAP (ATLAS, CMS, LHCb)
- Interest from Project Jupyter and C. Titus Brown (Genomics)







yadage - yaml based adage

pypi package 0.10.1 | build passing | health 91% | docs latest | 403.2MB 8 layers

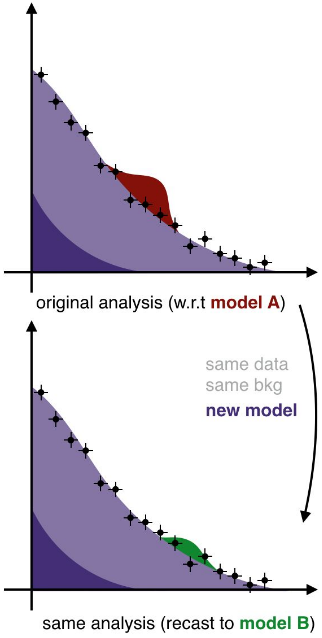
A declarative way to define adage workflows using a JSON schema (but we'll always write it as YAML)

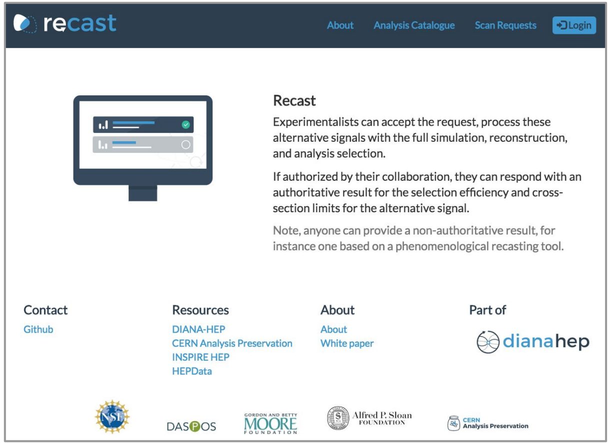
RECAST

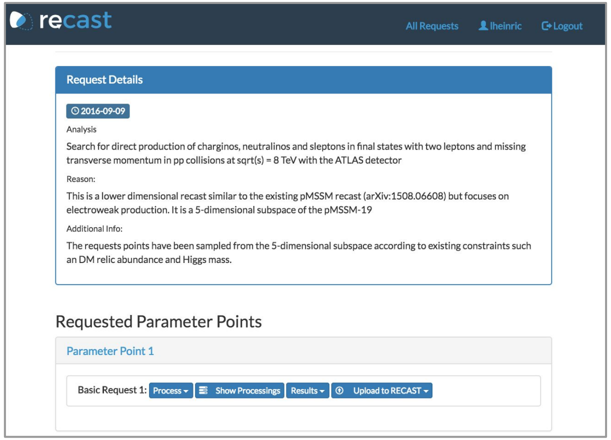
[Lukas Heinrich, Kyle Cranmer]

Effort to systematize reinterpretations of LHC analyses.

- Frontend allows 3rd party to submit proposals for reinterpretation. Experiments can accept or reject.
- Backend uses yadage to run full-fidelity analysis code developed in original publication (not a re-Implementation)
- Integration with CERN Analysis Preservation to access workflow definitions / archived software (docker images)
- Deployment at scale on CERN OpenStack Kubernetes Pilot (Magnum)
- Used internally by ATLAS for pilot recastings



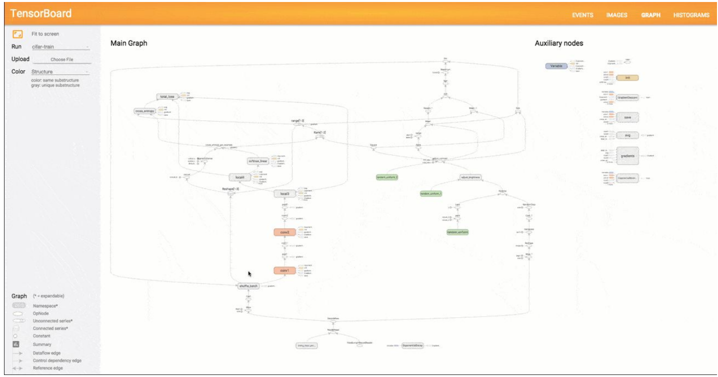




The modern AI/ML software stack

Recent switch to

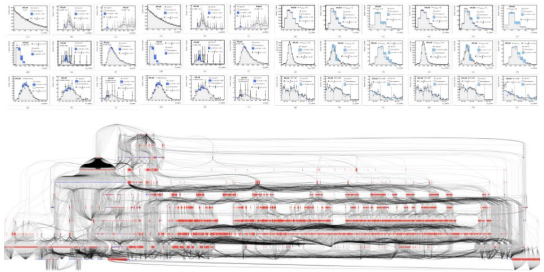
- Numerical computations with data flow graphs
  - TensoFlow, Theano, MXNet, etc
  - Support for CPUs and GPUs out of the box.
  - Automatic differentiation
  - Enable new ways of thinking (model composition, learning to learn, etc)
- Probabilistic programming languages
  - Stan, Anglican, Edward, etc



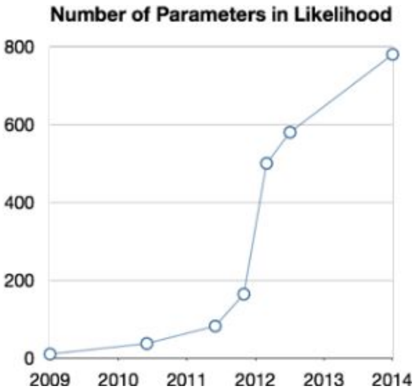
Recommendation. The next generation of physics software for high-level analysis should take notice and inspiration from the AI/ML community.

TensorFlow

Probabilistic programming frameworks



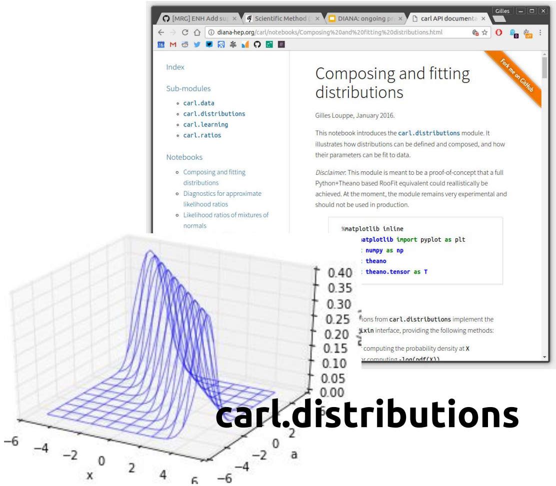
$$f_{\rm tot}(\mathcal{D}_{\rm sim},\mathcal{G}|\boldsymbol{\alpha}) = \prod_{c\in\text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha}))\prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha})\right]\cdot\prod_{p\in\mathbb{S}} f_p(a_p|\alpha_p)$$



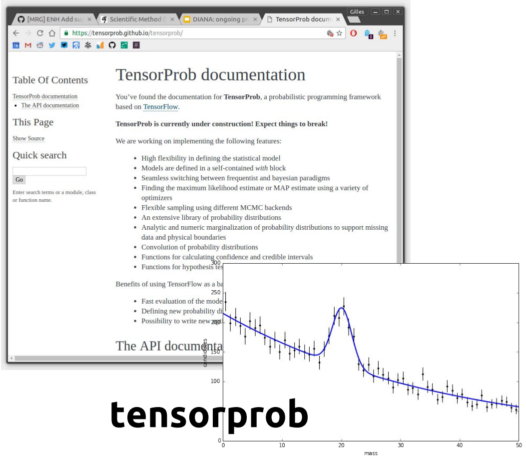
RooFit serves us well, but shows limits in terms of **scalability**.

Using a data flow graph framework, RooFit would be **distributed**, **GPU-enabled** and automatically **differentiable**.

Feasibility? Certainly **within reach!** As illustrated by our tentative proof-of-concepts carl.distributions [Gilles Louppe] and tensorprob [Igor Babuschkin, now at DeepMind]. See also Edward.



carl.distributions



tensorprob

Study of data flow graphs for statistical models in HEP

[Chien-Chin Huang, Matthew Feickert, Gilles Louppe, Kyle Cranmer]

- Ongoing study on the suitability/scalability of data flow graphs for statistical models in particle physics (vs. the typical ML use case)
    - Goal is to make an educated recommendation for/against these technologies in HEP.
- We found out that HEP likelihoods show unusual patterns in terms of model and data parallelism.
    - Also related to the fact that HEP likelihoods are optimized with full batches and Minuit (vs. mini-batches and SGD in ML)
- We identified that one of the bottlenecks is to have a declarative language for creating likelihoods of given shape. Opportunity for project merging Histogrammar and HistFactory.

Tensorflow benchmark

Execution time per loop for 10 levels FF (weights are all the size of 1024 x 1024) with 1 softmax network. The result is the average of 200 loops.

Batch SIZE	geeker-3:GPU0	geeker-3:GPU0 + ge geeker-3:CPU0 (sha Data parallelism	geeker-3:GPU0 + geeker-3:GPU1 geeker-4:GPU0 (shared paramete Data parallelism	geeker-3:GPU0 + geeker-3:GPU1 Model parallelism	geeker-3:GPU0 + geeker-4:GPU0 Model parallelism	
2	0.005631	0.037925	0.851353	0.016255	0.537392	
4	0.005754	0.037886	0.855416	0.016246	0.535423	
8	0.005815	0.037808	0.850857	0.016425	0.542009	
16	0.005842	0.038033	0.858842	0.016459	0.545612	
32	0.006151	0.038238	0.843241	0.016766	0.565767	
64	0.006184	0.03792	0.91737	0.01791	0.601712	
128	0.006472	0.038894	0.868997	0.018744	0.676443	
256	0.007427	0.03978	0.864928	0.021036	0.781197	
512	0.009843	0.039739	0.911763	0.027205	1.007766	Distributed model parallelism(125MB/loop) i
1024	0.017004	0.040637	0.873001	0.040332	1.462531	Distributed model parallelism(191MB/loop) i
2048	0.029238	0.044518	0.959429	0.070745	2.423491	Distributed model parallelism(322MB/loop) i
4096	0.053899	0.057538		0.130785		
8192	0.101063	0.095903		0.252418		
16384	0.20444	0.171501		0.490306		

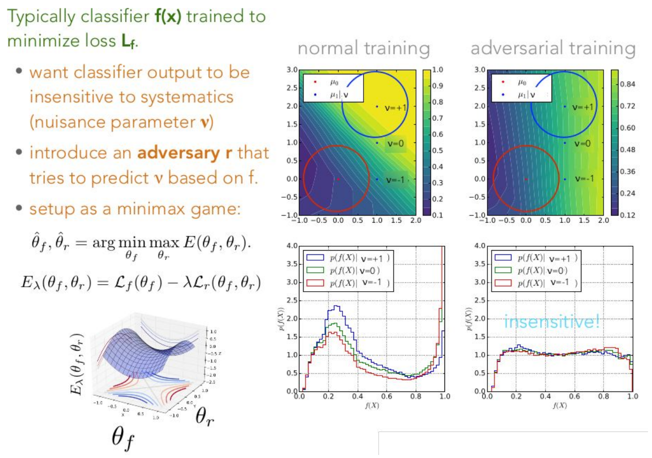
2048 x 2048 weights (16MB), 10 levels

Batch SIZE	Single GPU	Data parallelism	Model parallelism	Data parallelism	Model parallelism
2	0.006798	0.136345	0.032043	0.04985881404	0.2121524202
4	0.006885	0.13652	0.031905	0.05043217111	0.215796897
8	0.00703	0.136551	0.032043	0.05148259625	0.2193926911
16	0.008957	0.138055	0.032111	0.06487993915	0.2789386814
32	0.009023	0.140331	0.032336	0.0642979812	0.2790388422
64	0.011492	0.144435	0.032811	0.07956520234	0.3502483923
128	0.012741	0.142379	0.036927	0.08948651135	0.3450320903
256	0.017199	0.142795	0.045445	0.1204453937	0.3784574761
512	0.029618	0.165492	0.065492	0.1789693762	0.4522384413
1024	0.052463	0.149424	0.111745	0.3511015633	0.4694885677
2048	0.099097	0.157857	0.203965	0.627764369	0.485852965
4096	0.195194	0.201433	0.386243	0.9690269221	0.505365793
8192	0.409409	0.356251	0.756597	1.14921502	0.5411189841
16384	0.861228	0.724885	1.60836	1.188089145	0.535469671

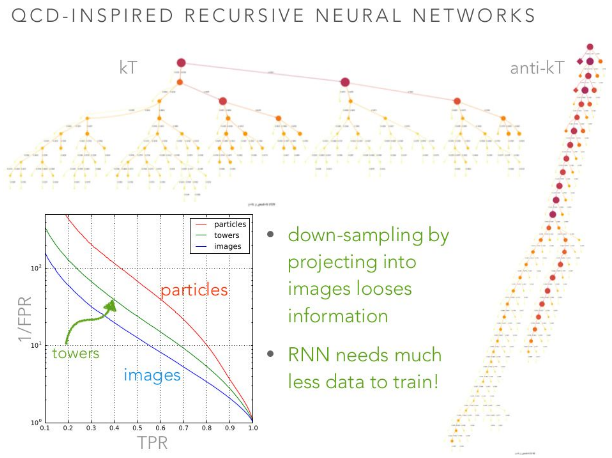
Enabling new solutions for particle physics problems

In addition to scalability, the modern ML/AI software stack enables to **rethink particle physics problems in new and effective ways**, that would otherwise be very complex to implement with the standard software stack of particle physics.

Learning to Pivot with Adversarial Networks
arXiv:1611.01046 [Gilles Louppe, Michael Kagan, Kyle Cranmer]



QCD-aware recursive deep learning for jet physics
[Gilles Louppe, Kyunghyun Cho, Cyril Becot, Kyle Cranmer]

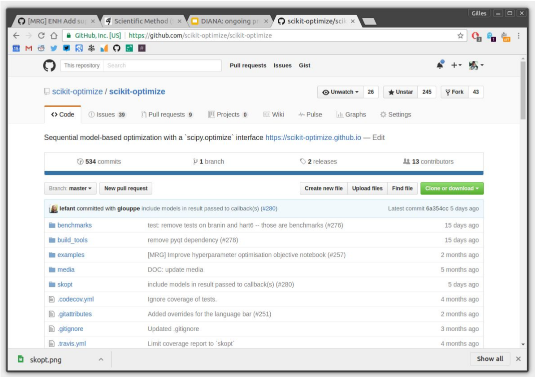


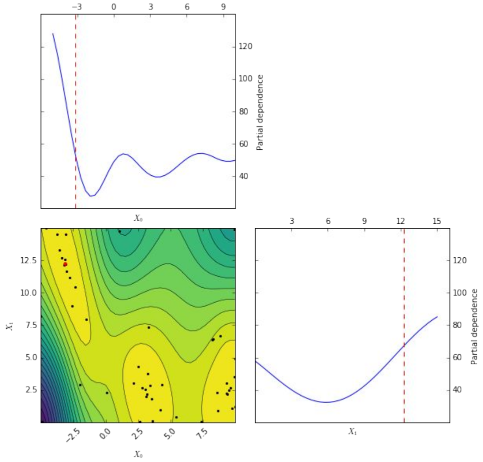
These two examples should help convincing physicists of the benefits of adopting these new tools.

Scikit-Optimize

[Manoj Kumar, Noel Dawe,
Tim Head, Gilles Louppe]

A simple and efficient library for sequential model-based optimization, accessible to everybody and reusable in various contexts.





The project is already quite advanced and popular.

Use cases in particle physics:

- Automated tuning of simulators
- Automated search for cuts
- Automated tuning of ML models

Training and software guidance

- Lectures and tutorials on machine learning **tailored for physicists**.
    - Focus on recent methodological developments and on modern software (scikit-learn, etc)
    - Opportunity to grow our user base

- Strong presence and participation at the IML, ATLAS ML and meetings at CERN.

- Internal guidance within ATLAS to setup an Anaconda-based machine learning software stack. [with Michael Kagan]

Roadmap

Short-term (next 6 months):

- Technical report on opportunities offered by data flow graph frameworks and probabilistic programming for high-level analysis in particle physics (as part of the CWP effort) [Gilles Louppe, Kyle Cranmer]
- Feasibility/Scalability study of graph-based models for statistical models in particle physics [Chien-Chin Huang, Matthew Feickert, Gilles Louppe, Kyle Cranmer]
- Sequential model-based optimization for HEP (software, physics use cases, tutorials) [Manoj Kumar, Tim Head, Noel Dawe, Gilles Louppe]
- Continue promoting modern machine learning software with tutorials and lectures.

Mid-term (next 12 months):

- Likelihood-free inference for particle physics (research, benchmarks, software, tutorials) [Juan Pavez, Cyril Becot, Lukas Heinrich, Gilles Louppe, Kyle Cranmer]
- Continue discussion on RooFit with auto-diff and probabilistic programming. One interesting path is auto-diff with clang [by Vassil Vassilev].
- Initiate discussion with the ROOT team on the challenges of implementing, in a sustainable way, algorithms like adversarial networks or jet embeddings in ROOT.
- Packaging, validation and release of statistical software for HEP ("fasimov", Fisher Interpolation, Look-elsewhere effect corrections, etc)