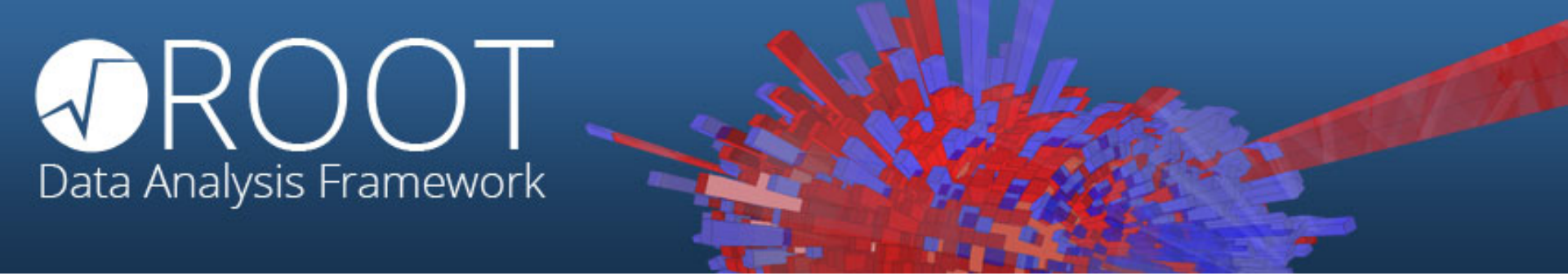


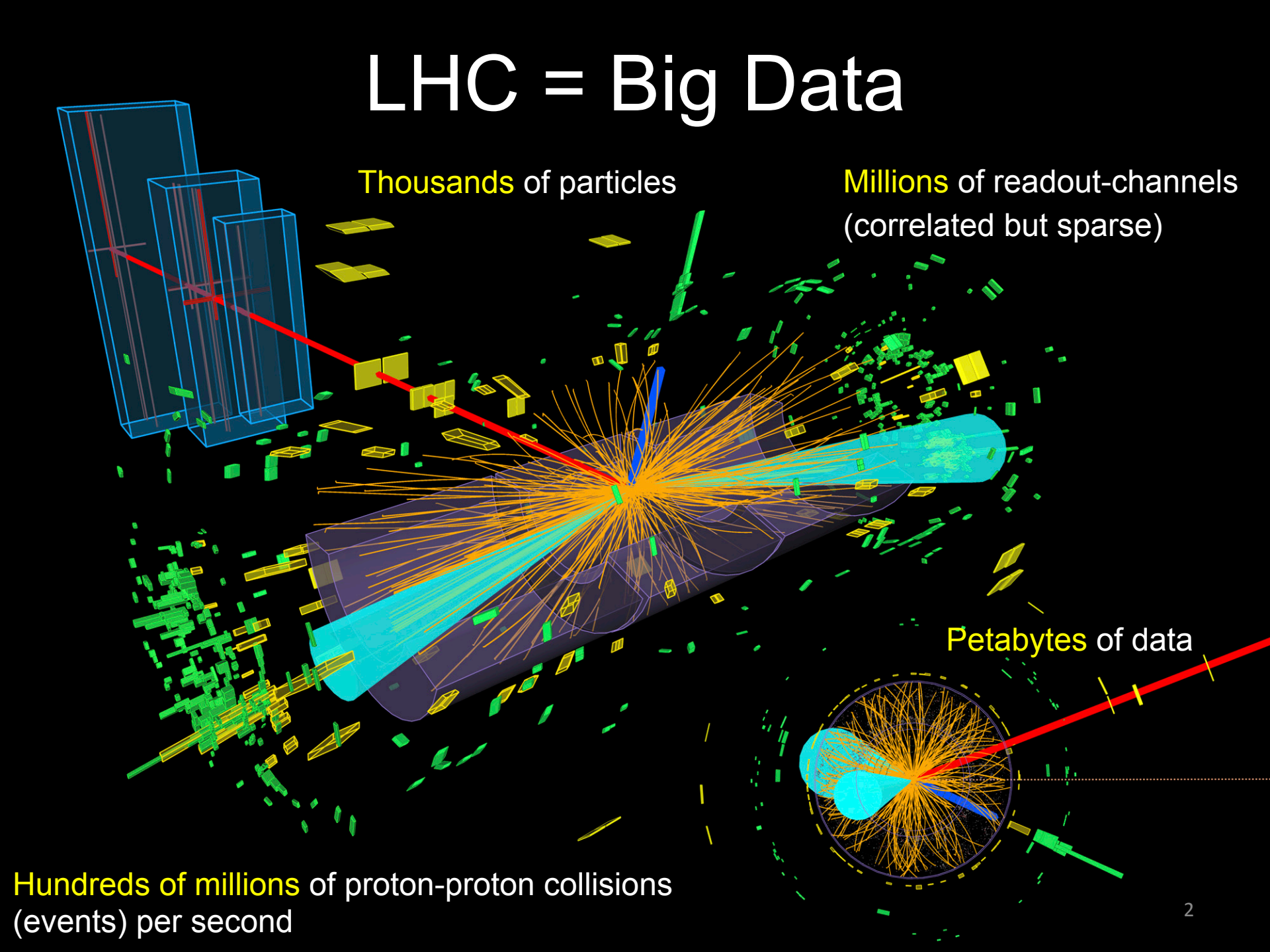
How to bring Modern Machine Learning to HEP

Tobias Golling, University of Geneva

"ROOT turns 20" Users' Workshop, Saas-Fee,
September 15-18 2015

UNIVERSITÉ DE GENÈVE

LHC = Big Data


Thousands of particles
Millions of readout-channels
(correlated but sparse)
Petabytes of data
Hundreds of millions of proton-proton collisions
(events) per second

Modern Machine Learning showcase

- **Modern Machine Learning (MML) is in all big-data fields**

- **MML is a fast-moving field**:
  - Finds cats, drives cars, answers questions (her), knows what we like (recommendations) – who knows what is possible in 1-2 decades…
  - Constantly improving **performance, automation, speed, robustness, applicability**
  - "in one decade MML will be more mainstream than C++"

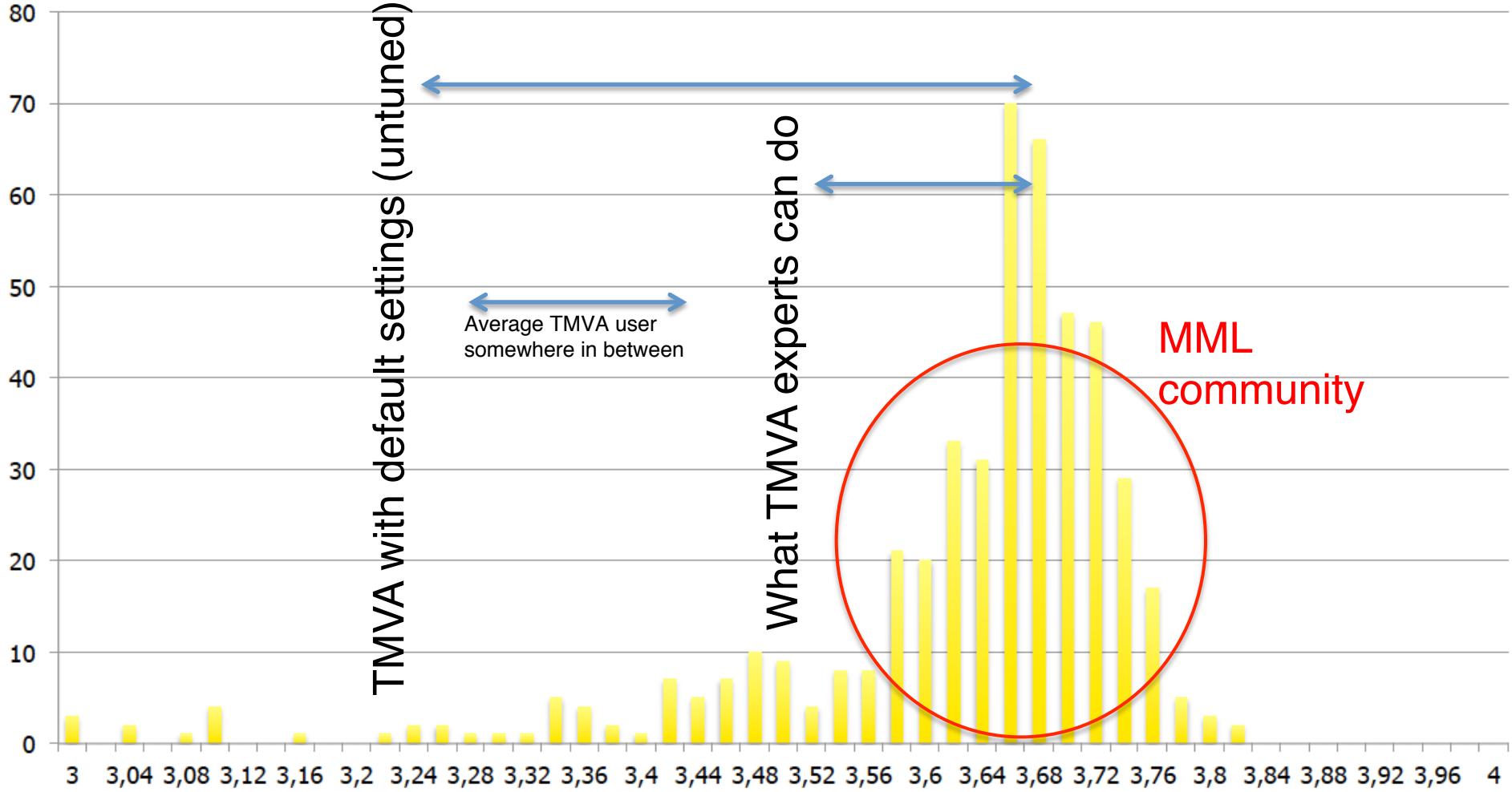
HEP is not capitalizing on this

Machine Learning (ML) usage in HEP

- Many HEP problems can be posed in form of a classification or regression problem
- Best signal-background discrimination, both high and low level
  - Kinematic selection for physics analysis (used in many Run 1 results, HiggsML Challenge, pheno papers, e.g. 1402.4735 ,…)
  - Object identification: b-tagging (e.g. MV1 in ATLAS, ATLAS-CONF-2014-046), boosted objects…
  - Track reconstruction (NN clustering for ATLAS pixel: 1406.7690, connecting the dots 2015 WS: https://indico.physics.lbl.gov/indico/conferenceDisplay.py?confId=149)
  - Trigger level (LHCb example: http://cds.cern.ch/record/2019813?ln=ru)
  - Idea to use ML in FPGA's for phase 2 upgrade
  - Most conventional algorithms are already black boxes to most users
  - Many of these applications are in production software
  - Many other ideas & plans…
  - I am sure there are similar use cases at the level of the LHC machine

Solidifying Case for MML for HEP

example from Higgs Machine Learning Challenge



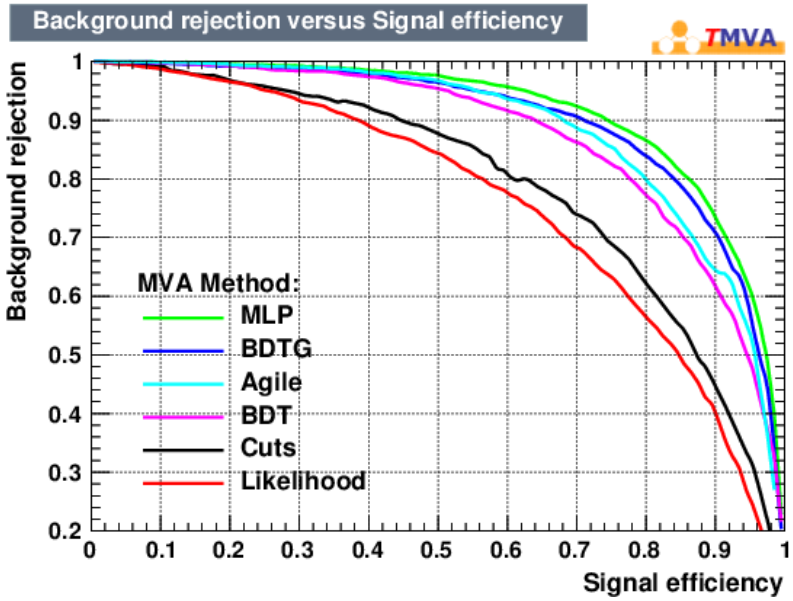
20-40% more data needed to get the same improvement

We have TMVA

- TMVA is the ROOT-integrated package for ML
- Provides first point of contact for people in HEP trying to use ML
- Has basic neural networks, boosted decision trees, etc
- Provides a common interface and associated support - very useful to HEP
- Written about 10 years ago, and ML has evolved significantly in that time
  - 10 years ago Deep Learning became mainstream, but no Deep Learning in TMVA
  - Originally written to introduce ML techniques to the HEP community
  - It has now fulfilled that purpose - time to move to the next stage

A few examples of TMVA extensions

- AGILEPack – a small C++ Deep Learning framework being applied within ATLAS (https://github.com/lukedeo/AGILEPack)
  - Following the NeuroBayes example we have a written a TMVA plugin class for AGILEPack
  - Also tried to train AGILEPack standalone, and use TMVA for evaluation
    - Problems: no complete separation of training and evaluation in TMVA, non-standard activation functions, conversion of NN configuration needed ⇒ not feasible
- xgBoost – a python wrapped C++ library for highly optimized, distributed gradient boosted decision trees (also Java/R/etc. wrapped)
  - Did extremely well in HiggsML Challenge
  - TMVA plugin for xgBoost?
- Ongoing collaboration HEP-ML
  - Luke de Oliveira, Pierre Baldi, Balazs Kegl, Yandex, kaggle, ChaLearn,…



(Demonstrator using only 2k events – by far not enough for Deep Learning)

Standing on the Shoulder of Giants

- Take advantage of ML community
- Capitalize on the insights of ML
- Avoid reinventing the wheel, so we can focus on doing physics
- Foster more communication between HEP and ML communities

Wish list for TMVA

- "Grassroots" effort (ATLAS, CMS, LHCb, ML experts,…), "future of TMVA" kick-off https://indico.cern.ch/event/441952/ , incomplete list of contributors & attendees

- Set up egroup for efficient communication across experiments: lhc-machinelearning-wg@cern.ch
  - Plan monthly meetings to define way forward
  - See initial feeback from this group on next slides

- Ece Akilli
- Andrea Coccaro
- Johannes Erdmann
- Sergei Gleyzer
- Tobias Golling
- Andreas Hoecker
- Marie Lanfermann
- Gilles Louppe
- Lorenzo Moneta
- Olaf Nackenhorst
- Luke de Oliveira
- Michela Paganini
- Maurizio Pierini
- David Rousseau
- Steven Schramm
- Peter Speckmayer
- Andrey Ustyuzhanin
- Pietro Vischia
- Helge Voss
- Simone Amoroso
- Dan Guest
- Maria Spiropulu
- Jean-Roch Vlimant
- Tommaso Dorigo
- Enrico Giraud
- O. Zapata

Core Requirements for TMVA

- Core TMVA package to provide a set of competitive and simple algorithms for standard HEP analysis usage
  - E.g. abovementioned xgBoost
  - Other core algorithms should also be updated
- TMVA interfaces for R and python (with support libraries) for high-performance use
  - Allows usage of MML packages
- Provide full and straightforward separation of training and testing
  - Allows one to train externally and to apply results through TMVA (for packages which are not simple to integrate with TMVA)

Modernising TMVA

- Flexibility
  - more modular code ⇒ straightforward to add interfaces (see progress by the RMVA group)
  - Additional support for external input file formats (such as those used in ML, e.g. HDF5)
  - Decoupling datasets/methods/variables in contrast to the current approach (RMVA progress)
- Computational Performance
  - The core code should be redesigned for improved computational performance
  - Use latest C++ features, vectorization and optimized Math libraries
  - Dataset I/O should be revisited, e.g. only relevant parts of the dataset are held in memory
- Latest ML improvements
  - Avoid re-inventing the wheel
  - Easy interfaces to the most powerful ML methods (see RMVA and PyMVA)
  - Promising: R and python interfaces, with additional support libraries (scikit-learn, pandas etc.)
  - A fully flexible interface for arbitrary language wrappers would be very useful, and should be easier after the R and python interfaces

Desired TMVA Features

- Cross-validation
  - Standard in ML
  - New redesign by the RMVA team allows easy implementation due to feature/method/dataset decoupling
- Additional information for analyzer
  - Variable importance, accurate feature ranking
  - FAST algorithm for feature importance currently being integrated by the RMVA team with the new redesign
- Parallelization
  - Many places where it applies, the RMVA team is currently working on a general prototype
- GPU support (important for the most computationally intensive algorithms)
- Define/provide a high-statistics sample for testing purposes: the current sample within TMVA is not adequate for studying modern algorithm performance
- Expert users should be able to pause and resume training after tweaking hyperparameters as is done in the ML community
- Make it easier for the ML community to contribute directly to TMVA such as through a GitHub repository which is open to pull requests

Impact of TMVA Redesign

- All users: improved computational performance and dataset flexibility
- Standard users: provides access to modern ML algorithms for additional power
- Performance users: provides access to cutting-edge ML algorithms through interfaces
- Potential developers: improved modularity makes it easier to contribute, interfaces make it easier to try new things, lots of areas for interested parties to contribute
- ML experts working with HEP: facilitates interactions between HEP and ML
  - Easier to re-import ML results into HEP, increasing the benefit of ML challenges and reducing the overhead of exploiting new ML techniques
  - Easier for ML community to work with the software they are familiar with and which may be better optimized for a given problem (in a way we did not consider); if we place restrictions on how they approach problems, this may become a limitation

Conclusion

- Interest from HEP & ML community to bring MML to HEP
  - Very promising initial results / work done
  - Still a lot to do: data-MC comparison, systematics, etc.
- We believe in TMVA – need dedicated, long-term support for TMVA
- Need coordination & "forum" to discuss, work & bring together people with MML@HEP interests
  - Develop & sustain MML4HEP expertise – it is not just about software, but need ML know-how & insights, e.g.
    - which algorithms to use for which problem
    - how to tune hyperparameters
    - how to deal with non-continuous or missing variables
    - Troubleshooting, novel applications, data vs. MC,…
  - Series of dedicated LHC ML challenges to further strengthen & grow MML-HEP interaction, so we can more effectively collaborate
  - For now started with egroup: lhc-machinelearning-wg@cern.ch
  - Decide on form of coordination/support/forum – input welcome
    - To be discussed further at Data Science workshop in November

Backup

References to work done by the RMVA group

- TMVA restructuring for modularity: http://oproject.org/TMVA
- RMVA interface: http://oproject.org/RMVA
- PyMVA (scikit-learn) interface: http://oproject.org/PyMVA

- See talk by Lorenzo Moneta