iris hep

Institute for Research & Innovation
in Software for High Energy Physics

Analysis Grand Challenge

Kyle Cranmer (NYU)

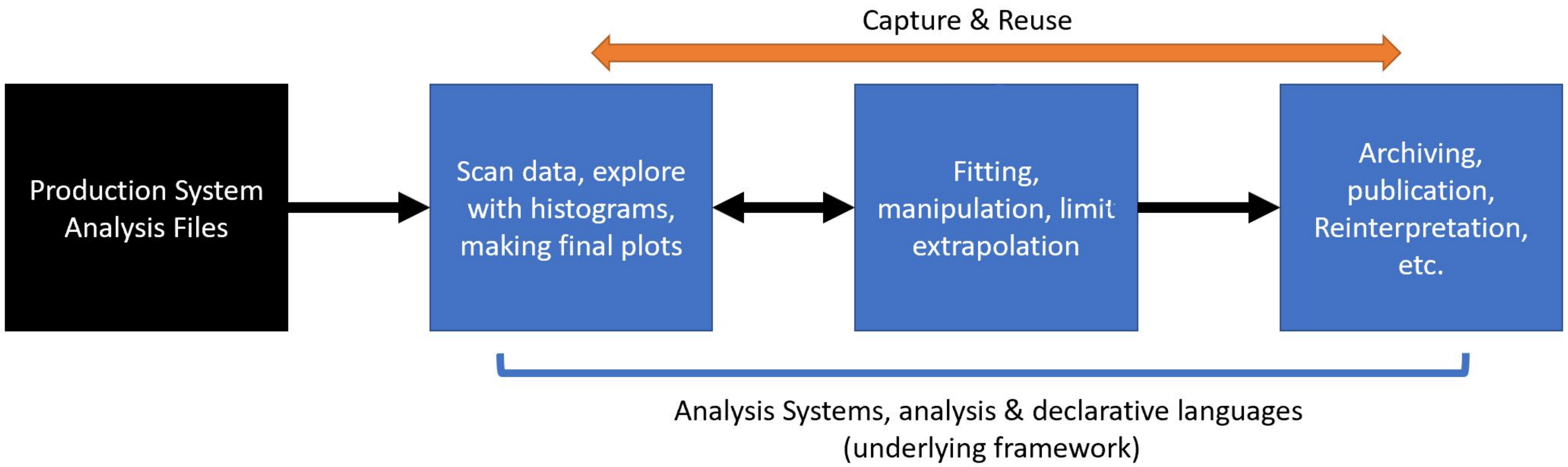
Some considerations

Looking for a challenge that:

- involves multiple IRIS-HEP products and can serve as an effort to help unifying and connect projects / efforts
- Needs to be clearly relevant to HL-LHC and IRIS-HEP goals
- Would like it to span scope of Analysis Systems
- Improve “light-house” and intellectual hub aspect of IRIS-HEP
- Align with other goals, like training and workforce development

Analysis Systems

- Develop sustainable analysis tools to extend the physics reach of the HL-LHC experiments
  - *create greater functionality to enable new techniques,*
  - *reducing time-to-insight and physics,*
  - *lowering the barriers for smaller teams, and*
  - *streamlining analysis preservation, reproducibility, and reuse.*


Capture & Reuse
Production System Analysis Files
Scan data, explore with histograms, making final plots
Fitting, manipulation, limit extrapolation
Archiving, publication, Reinterpretation, etc.
Analysis Systems, analysis & declarative languages (underlying framework)

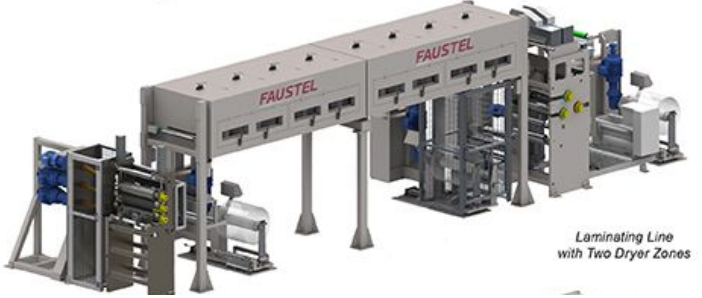
Analysis Systems projects span all stages of end-user analysis.

Value of IRIS-HEP as an Institute

IRIS-HEP as a lighthouse:

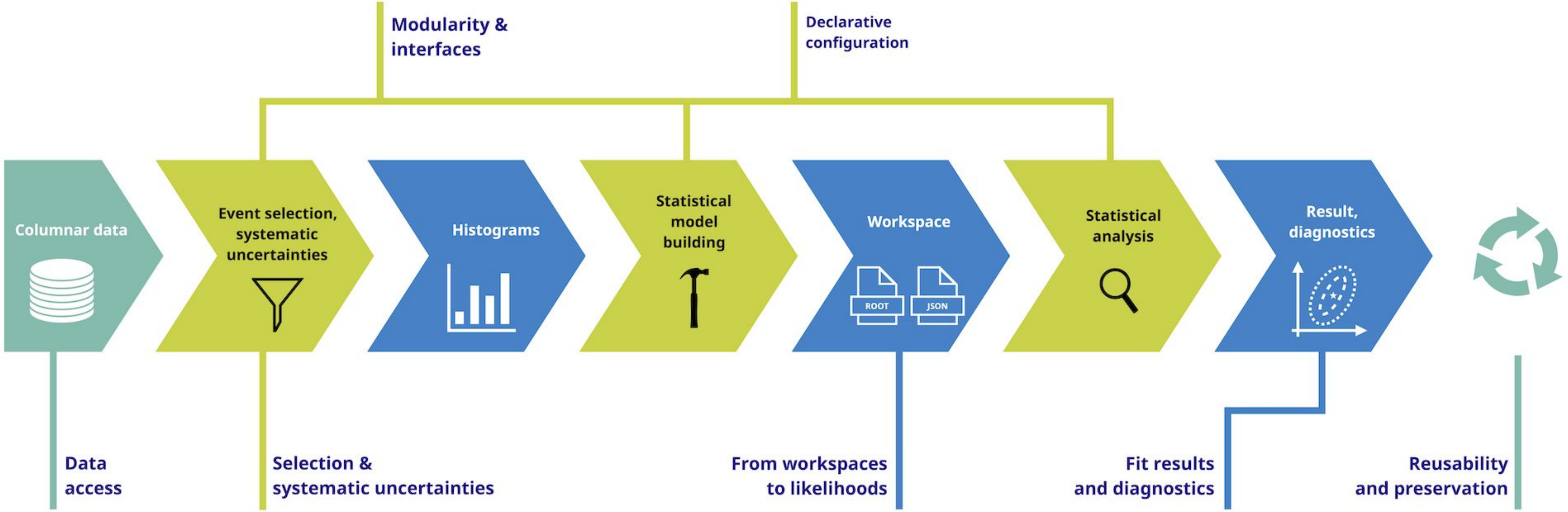
- provide cohesive, long-term vision for how software should evolve to meet needs of HL-LHC
- take advantage of holistic perspective of the institute
- Examples:
  - *columnar analysis*
  - *declarative programming*
  - *differentiable programming*
  - *preservation & reuse*

A coherent ecosystem



func_adl
formulate

cabinetry


Modularity & interfaces
Declarative configuration
Columnar data
Event selection, systematic uncertainties
Histograms
Statistical model building
Workspace
ROOT
JSON
Statistical analysis
Result, diagnostics
Data access
Selection & systematic uncertainties
From workspaces to likelihoods
Fit results and diagnostics
Reusability and preservation

ServiceX

uproot

Awkward Array

coffea

Boost histogram

pyhf
∂ifferentiable
ℒikelihoods

yadage

recast

Link to Google doc

User Story

As an analyzer, I want to optimize an analysis end-to-end for a targeted signal hypothesis (including systematics) on an HL-LHC sized dataset so that I can obtain sensitive observed results for that signal while still being able to reinterpret the analysis for various signal hypotheses.

Link to Google doc

Assumptions

1. We have ~200TB for background MC samples, a specific signal hypothesis to target for optimization, and a placeholder for “observed data”
2. We have a typical SUSY search with multiple selection regions, cuts, and variables to be histogrammed for input to a template analysis like that of pyhf. The analysis strategy has some sensitivity to the target signal hypothesis.
3. We have an analysis facility with ServiceX and SkyHook (hand off from DOMA to AS) and 1500 cores. If each core can process @ 50 kHz each this gives 75MHz which would process 100B events @ 2kB/event (=200TB) in 25 min. Eg. 25 min per optimization iteration.
4. We have necessary ingredients to compute systematic variations.
   Either:
   a. Pre-computed event weights, scale factors, varied kinematics, etc. that needs to be processed for input to statistical model, or
   b. code to compute those ingredients on the fly
   c. Could also have (b) triggered in first pass, and then use those cached values for those ingredients cached for the later optimization passes.
5. “End-to-end” starts with necessary ingredients described above and ends with limits on signal strength and background-only p-values as objective of optimization
   a. Assuming here the beginning of the analysis chain is already in a format compatible with columnar analysis tools (eg. Arrow, Awkward, Coffea) and no conversion from xAOD etc. is needed -- this has already been demonstrated and such a conversion shouldn’t be part of our eventual analysis model)
6. We have multiple signal scenarios suitable for reinterpretation.

Link to Google doc

Acceptance Criteria

1. End-to-end analysis optimization including systematics on a realistically sized HL-LHC (~200TB) end-user analysis dataset + observed limit & reinterpretation afterburner
2. Specifications of regions, variables, and systematic variations declared using cabinetry and func_adl
3. Corresponding input datasets should be identified using in a way that is, abstracted from traditional file-based interface
4. Use of ServiceX, SkyHook, Coffea to perform event selection and deliver histograms for pyhf model
5. Optimize analysis by using automatic differentiation to compute d(Expected limit)/d(analysis parameters), which are back-propagated from from output of stats tool, through pyhf running in fitting service, back to ServiceX running at analysis facility, and through the event selection & histogramming code.
    - Notes:
        - we have prototypes for this running on a single machine [see grad-hep, neos]
        - We could make some aspects of this challenge be stretch goals, but let's don't water it down so that this goal is not taken seriously.
        - We will need to do some more quick investigation before we are are able to
        - it is also possible to use non-gradient based approaches for optimization (eg. Bayesian Optimization) that wouldn't require passing gradients back to ServiceX. And some of these approaches can use partial gradient information.
6. Once optimized: apply optimized analysis to "observed data" (may also be synthetic in reality) to obtain "observed limits".
7. Analysis Preservation & RECASTing:
    - Given a record pointing to preserved analysis (may be more than a git repository, could also include docker images, workflow components, etc.) and that I have access to compatible compute resources, I can reproduce results and reinterpret the analysis
8. Stretch: using active learning [excursion] to reinterpret efficiently

Why Differentiable?

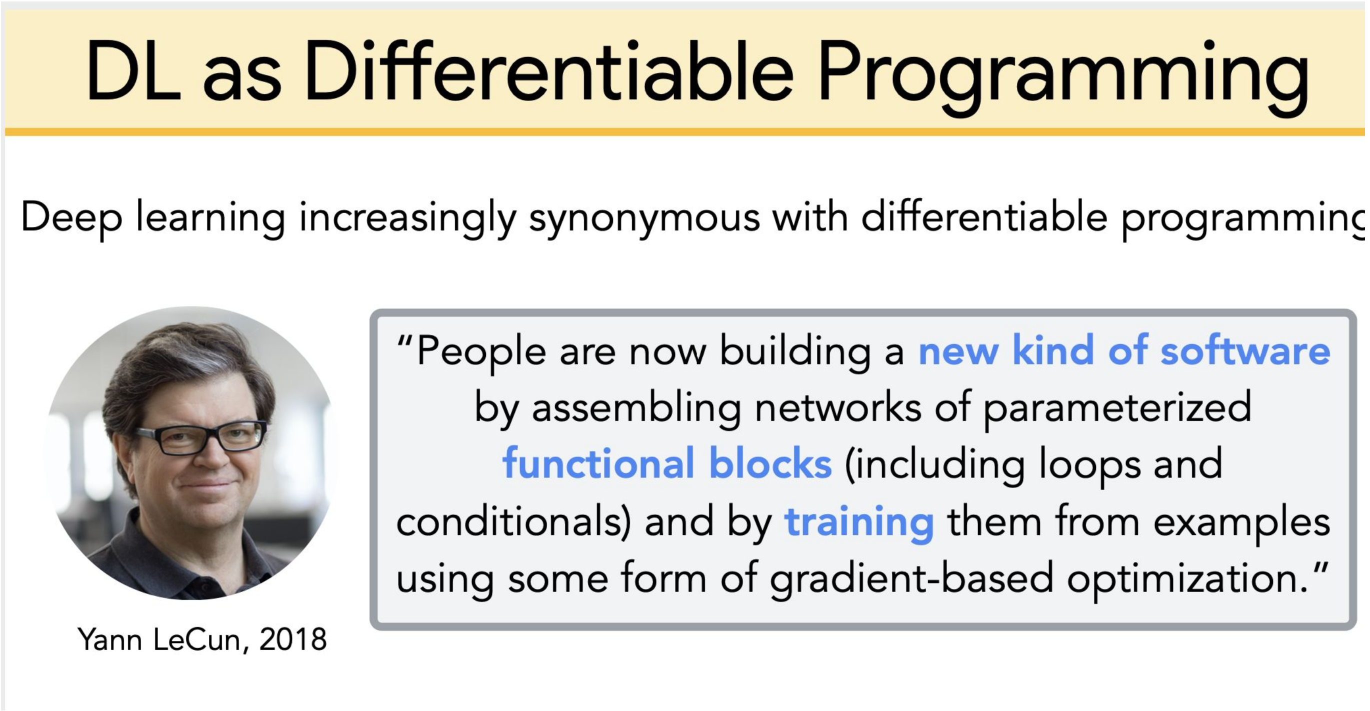
Forward looking, game changing functionality

Fast differentiable sorting and ranking

M.Blondel O. Teboul Q. Berthet J. Djolonga

March 12th, 2020

Google Research



DL as Differentiable Programming

Deep learning increasingly synonymous with differentiable programmin

"People are now building a **new kind of software** by assembling networks of parameterized **functional blocks** (including loops and conditionals) and by **training** them from examples using some form of gradient-based optimization."

Yann LeCun, 2018

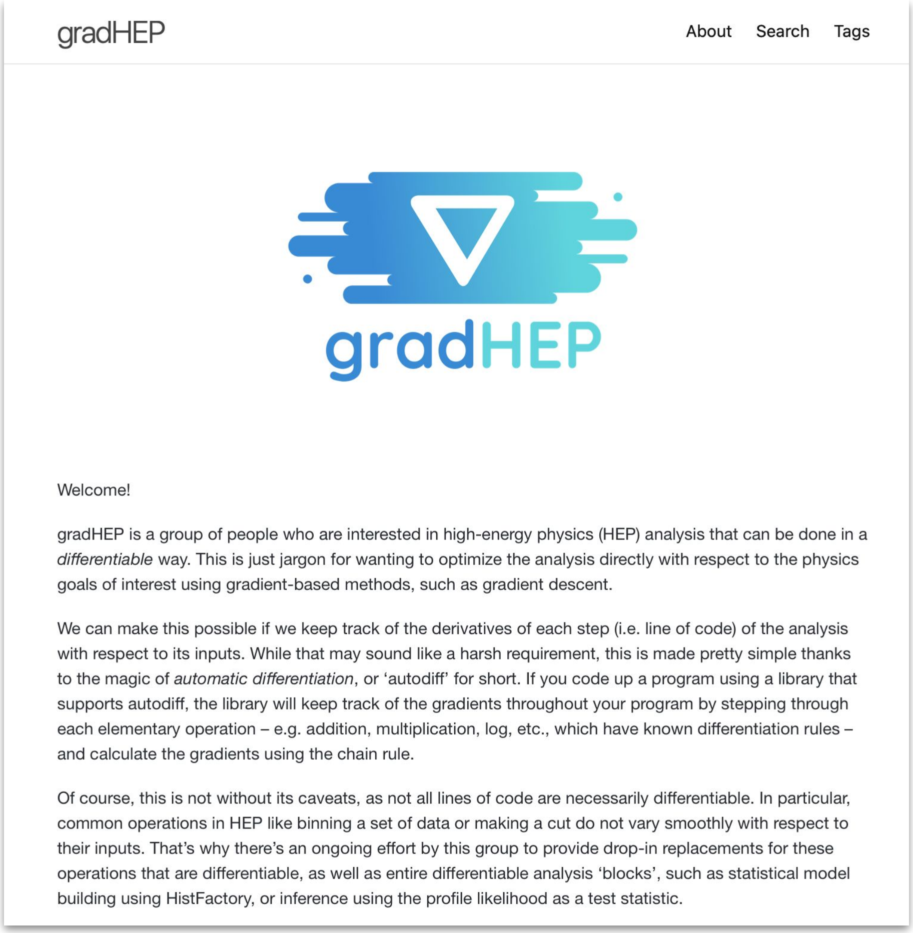
Wikipedia on Differentiable programming

Why focus on differentiable programming?

- **Intellectual Leadership:** It is a modern paradigm growing and abstracting from success of deep learning, and a more natural fit to HEP than replacing everything with machine learning.
- **Increased Functionality:** We will have more sensitive analyses. Differentiable analysis systems would dramatically accelerate and improve essentially all fitting / tuning / optimization tasks. It also facilitates propagation of uncertainty in a more powerful way. Paves way to hybrid systems that fuse traditional approaches and machine learning more seamlessly.
- **Connection with Industry:** This has been an effective conduit to connections with Google (Jax and Tensorflow teams) and pytorch community.
- **Foster Innovation:** there are a ton of ideas around use of differentiable programming
- **Training & Workforce development:** These are very valuable skills, young people will do much better on job market if they are familiar with diff prob.

In our community

http://gradhep.github.io

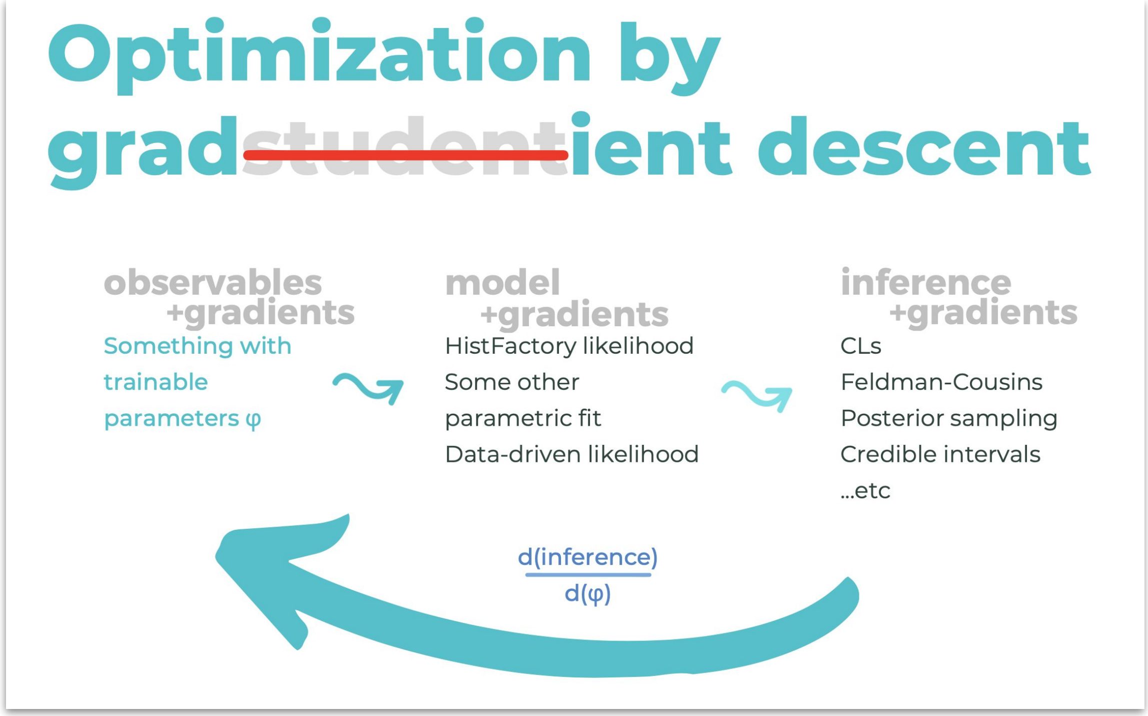


Welcome!

gradHEP is a group of people who are interested in high-energy physics (HEP) analysis that can be done in a *differentiable* way. This is just jargon for wanting to optimize the analysis directly with respect to the physics goals of interest using gradient-based methods, such as gradient descent.

We can make this possible if we keep track of the derivatives of each step (i.e. line of code) of the analysis with respect to its inputs. While that may sound like a harsh requirement, this is made pretty simple thanks to the magic of *automatic differentiation*, or 'autodiff' for short. If you code up a program using a library that supports autodiff, the library will keep track of the gradients throughout your program by stepping through each elementary operation – e.g. addition, multiplication, log, etc., which have known differentiation rules – and calculate the gradients using the chain rule.

Of course, this is not without its caveats, as not all lines of code are necessarily differentiable. In particular, common operations in HEP like binning a set of data or making a cut do not vary smoothly with respect to their inputs. That's why there's an ongoing effort by this group to provide drop-in replacements for these operations that are differentiable, as well as entire differentiable analysis 'blocks', such as statistical model building using HistFactory, or inference using the profile likelihood as a test statistic.



Optimization by grad~~student~~ient descent

observables +gradients

Something with trainable parameters φ

model +gradients

HistFactory likelihood
Some other parametric fit
Data-driven likelihood

inference +gradients

CLs
Feldman-Cousins
Posterior sampling
Credible intervals
...etc

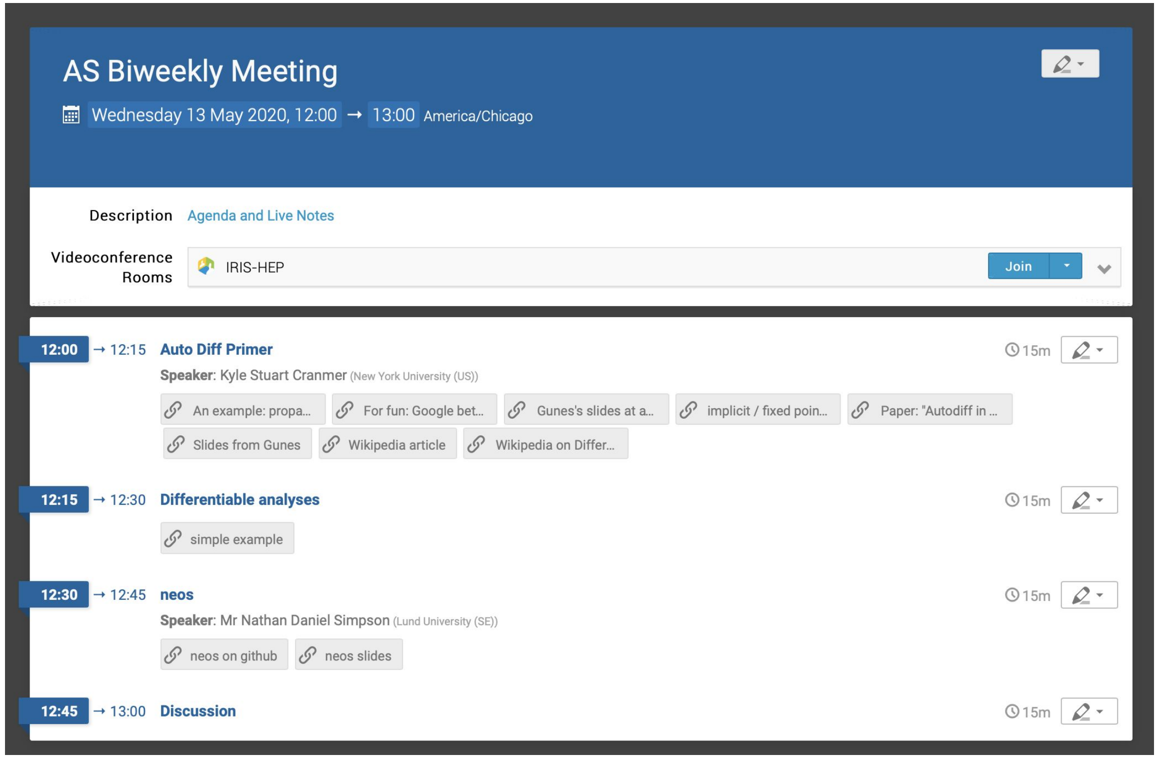
$$\frac{d(\text{inference})}{d(\varphi)}$$

slide from Nathan Simpson: [link to talk]

Wikipedia on Differentiable programming

Prototypes

https://indico.cern.ch/event/915053/



AS Biweekly Meeting

Wednesday 13 May 2020, 12:00 → 13:00 America/Chicago

Description Agenda and Live Notes

Videoconference Rooms IRIS-HEP Join

12:00 → 12:15 **Auto Diff Primer** 15m

Speaker: Kyle Stuart Cranmer (New York University (US))

An example: propa... | For fun: Google bet... | Gunes's slides at a... | implicit / fixed poin... | Paper: "Autodiff in ... | Slides from Gunes | Wikipedia article | Wikipedia on Differ...

12:15 → 12:30 **Differentiable analyses** 15m

simple example

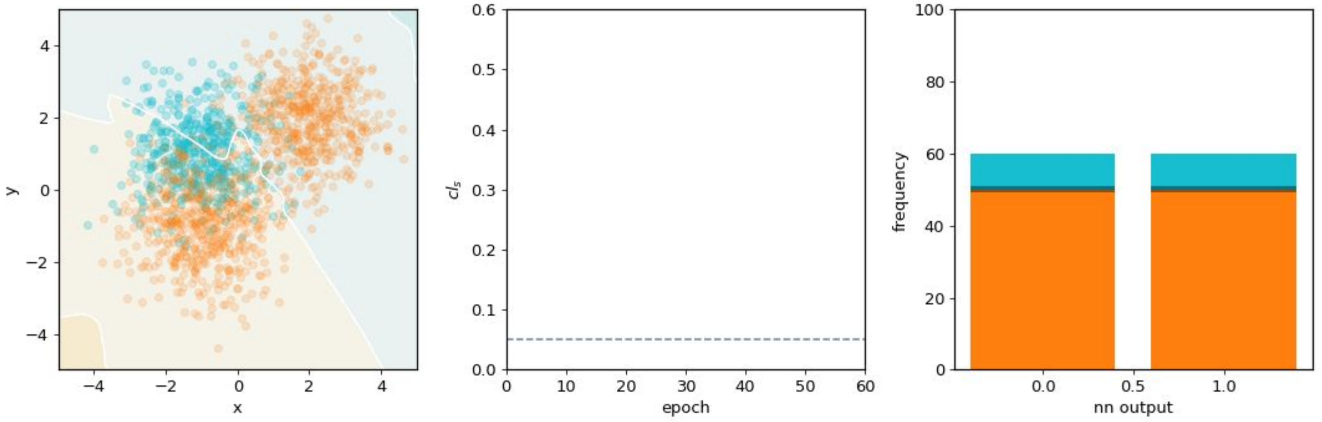
12:30 → 12:45 **neos** 15m

Speaker: Mr Nathan Daniel Simpson (Lund University (SE))

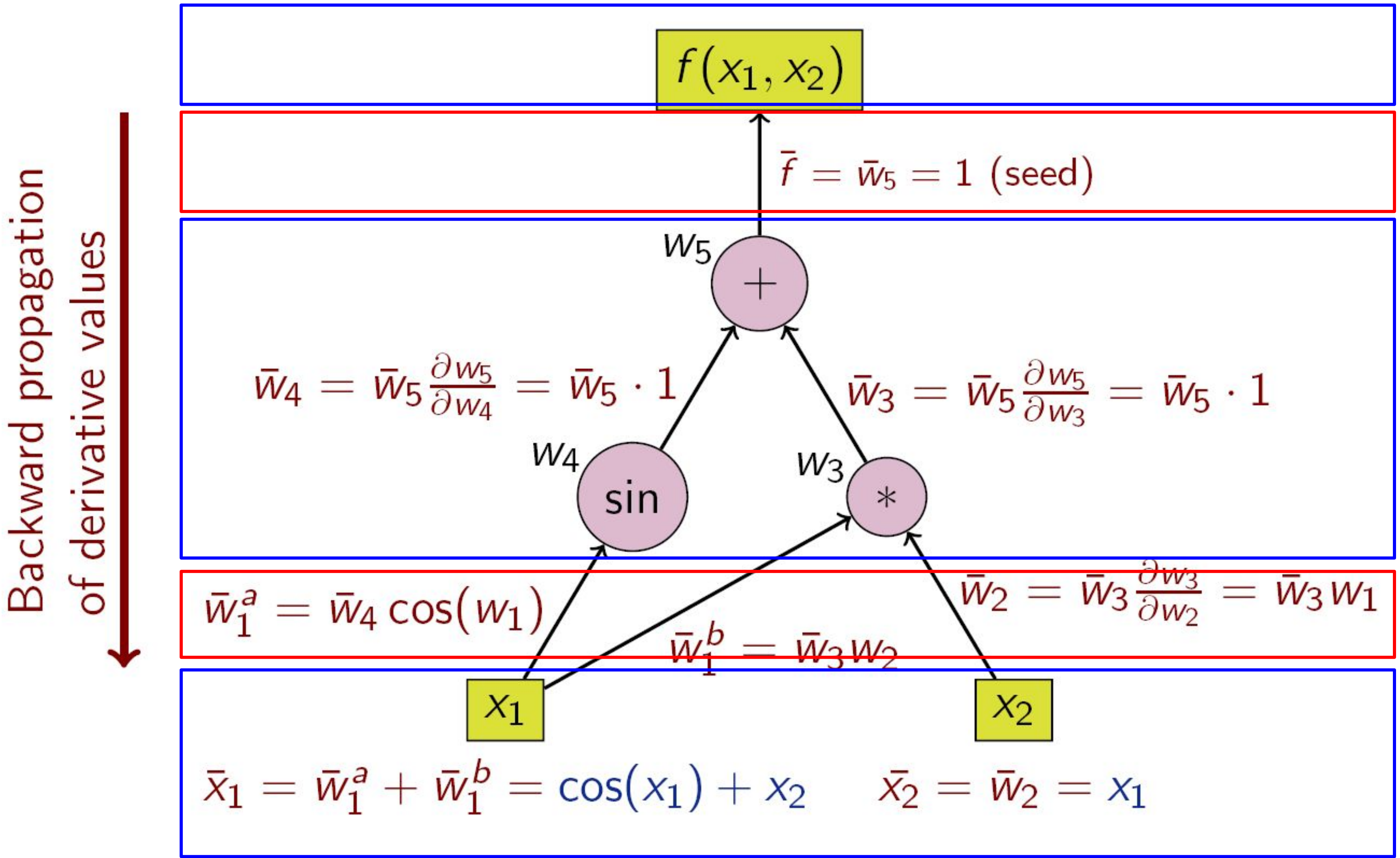
neos on github | neos slides

12:45 → 13:00 **Discussion** 15m

neos



Challenge: Auto-diff across systems


Backward propagation of derivative values
$f(x_1, x_2)$
$\bar{f} = \bar{w}_5 = 1$ (seed)
w_5
$+$
$\bar{w}_4 = \bar{w}_5 \frac{\partial w_5}{\partial w_4} = \bar{w}_5 \cdot 1$
$\bar{w}_3 = \bar{w}_5 \frac{\partial w_5}{\partial w_3} = \bar{w}_5 \cdot 1$
w_4
sin
w_3
$*$
$\bar{w}_1^a = \bar{w}_4 \cos(w_1)$
$\bar{w}_2 = \bar{w}_3 \frac{\partial w_3}{\partial w_2} = \bar{w}_3 w_1$
$\bar{w}_1^b = \bar{w}_3 w_2$
x_1
x_2
$\bar{x}_1 = \bar{w}_1^a + \bar{w}_1^b = \cos(x_1) + x_2$
$\bar{x}_2 = \bar{w}_2 = x_1$

fitting service calculates expected significance or limit

pass gradients back

Final event selection, filling of histograms and building of statistical model

pass gradients back ?

Initial selection of events and columns needed