

Adoption of Automatic Differentiation in HEP

[some thoughts with focus on analysis]

Alexander Held¹

¹ University of Wisconsin–Madison

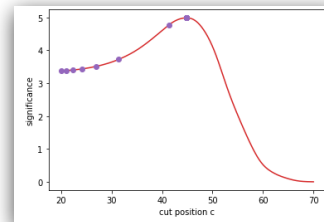
Differentiable Analysis Blueprint

<https://indico.cern.ch/event/1633539/>

March 5, 2026

“First wave” (?) of diffprog in HEP

[2020: I wrote a very simple example demonstration which felt late to the party at that point, different in retrospect]



- **2018+:** **first wave of interest** in the topic around the time of **INFERNO** [[arXiv:1806.04743](https://arxiv.org/abs/1806.04743)]

- gradient-based **analysis optimization**, later on also **neos** [[arXiv:2203.05570](https://arxiv.org/abs/2203.05570)]

- 2020: Snowmass whitepaper

- *“a long-term goal of this effort is to optimize real physics analyses”*

- *“achieving this goal is ambitious”*

- *“not clear that gradient-based optimization will always be superior”*

Differentiable Programming in High-Energy Physics

Atihm Güneş Baydin (Oxford), Kyle Cranmer (NYU), Matthew Feickert (UIUC), Lindsey Gray (FermiLab), Lukas Heinrich (CERN), Alexander Held (NYU), Andrew Melo (Vanderbilt), Mark Neubauer (UIUC), Jannicke Penke (Stanford), Nathan Simpson (Lund), Nick Smith (FermiLab), Gordon Stark (UCSC), Savannah Thais (Princeton), Vassil Vassilev (Princeton), Gordon Watts (U. Washington)

August 31, 2020

- **2020+:** **related activity in IRIS-HEP blueprints**

- 2020: “Future Analysis Systems and Facilities”: what is technically possible / feasible?

- 2021: “Differentiable Programming for the Analysis Grand Challenge”

- *“How far back will the backward pass extend?”*, lead eventually to [arXiv:2508.17802](https://arxiv.org/abs/2508.17802)

Prototype: differentiating through PyTorch + JAX + Tensorflow using functions as a service

- Also: HSF activity area / gradHEP group, MODE whitepaper, MIAPbP workshop, lots more papers

Differentiable Programming for the Analysis Grand Challenge

1 December 2021

Zoom

Enter your search text

An IRIS-HEP Blueprints Workshop

Objective: The purpose of this meeting is to finalize the planning document to complete the IRIS-HEP milestone (C) Differentiable programming teaching course services needed for analysis challenge.

Overview: This project is supported by National Science Foundation cooperative agreement OAC-1936602 (DUE19075). Any opinions, findings, conclusions or recommendations expressed in this material are those of the developers and do not necessarily reflect the views of the National Science Foundation.

Starts: Dec 2021, 16:00

Ends: Dec 2021, 18:30

Zoom

Virtual

Organizer: Alexander Held

App Store: Contact

Google Play: Contact

Google Drive: Contact

There are no materials yet.

Which gradients do we want?

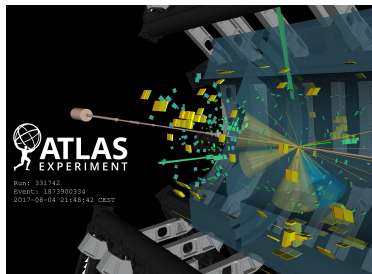
observables x

detector interaction z_D

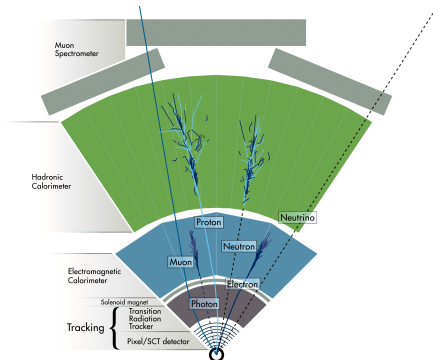
parton shower z_S

parton level z_P

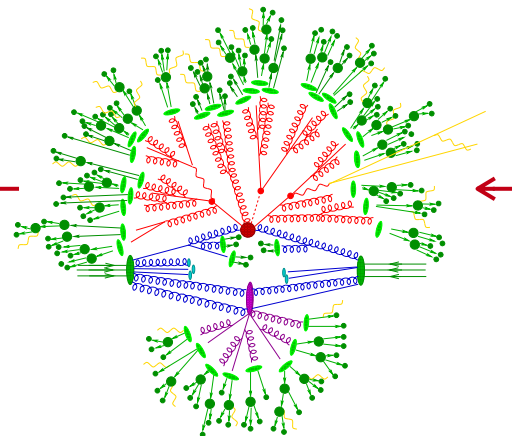
[physics objects $f(z_D)$]



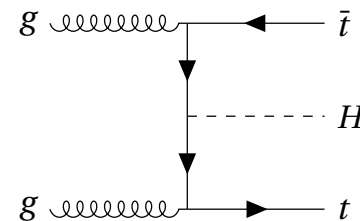
Phys. Lett. B 784 (2018) 173



CERN-EX-1301009



JHEP 0902 (2009) 007



$\partial/\partial x$

“normal” analysis-level ML

∂/∂ (energy deposit in cell)

“Hits to Higgs”

∂/∂ (PS parameter)

“let’s tune our PS!”

$\partial/\partial \kappa_\lambda$

full end-to-end

∂/∂ (object definition)

“what is a jet / b-tag?”

∂/∂ (GEANT parameter)

“let’s tune Geant4”

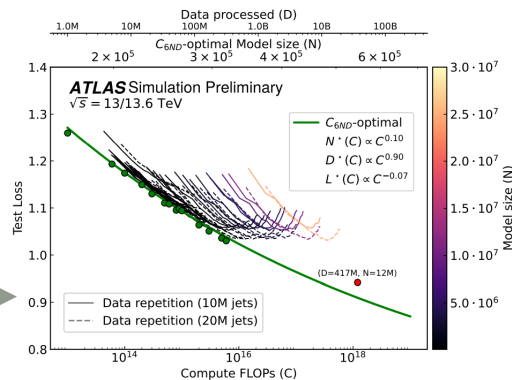
[take advantage of (approximate) factorization -> SBI / mining gold]

Status quo

- **Autodiff in HEP = “standard ML” = optimize NN** wrt. proxy metric, not the final number in the paper
 - established, accepted, and **works very well**
- The **“differential programming” perspective** has not (yet?) significantly changed the way we work
 - will / should it?

Challenges and key questions

- **Not everything can be easily differentiated through**: sometimes technically difficult, sometimes conceptually
 - see e.g. [Annalena's talk](#) and [Frederic's talk](#)
 - can we **replace (all) non-differentiable pieces** with suitable alternatives (*without* a lot of manual intervention?)
- How **computationally feasible** is this for an HL-LHC scale analysis?
 - pass gradients through large, complicated, **distributed** differentiable programs
 - can we store all the **gradient tape** we need (in memory)?
 - parts of pipeline are increasingly expensive -> unlikely to re-optimize per analysis →
 - what **software stack** do we need, where can we harmonize?



- How **computationally efficient** is this compared to non-gradient **blackbox optimization**?
 - can we tell **without needing to implement things fully**?

[arXiv:2203.13818]

Depending on the optimization task and its software computations packages different hardware resources should be provided. Thus, the system should run on relatively capable virtual nodes (24 CPUs, 128GB of RAM). However, some of the computational tasks might require additional hardware resources like GPU/TPU cards or **extremely large RAM volumes**. For such dedicated tasks the system should be able to instantiate a separate computational Kubernetes clusters.

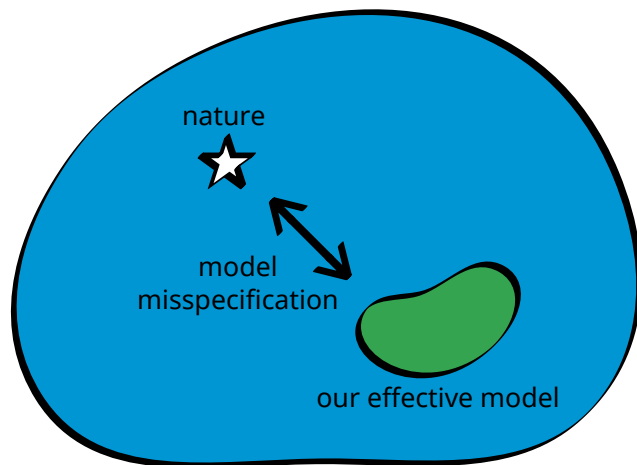
Model misspecification

- Our **model of nature is wrong** in ~all interesting use cases
 - how can gradient-based optimization **avoid optimizing the wrong thing**
 - what is the **right loss** to optimize (“avoid overconstraints” etc.)?
 - how to fold in the the **“misspecification loop”**?

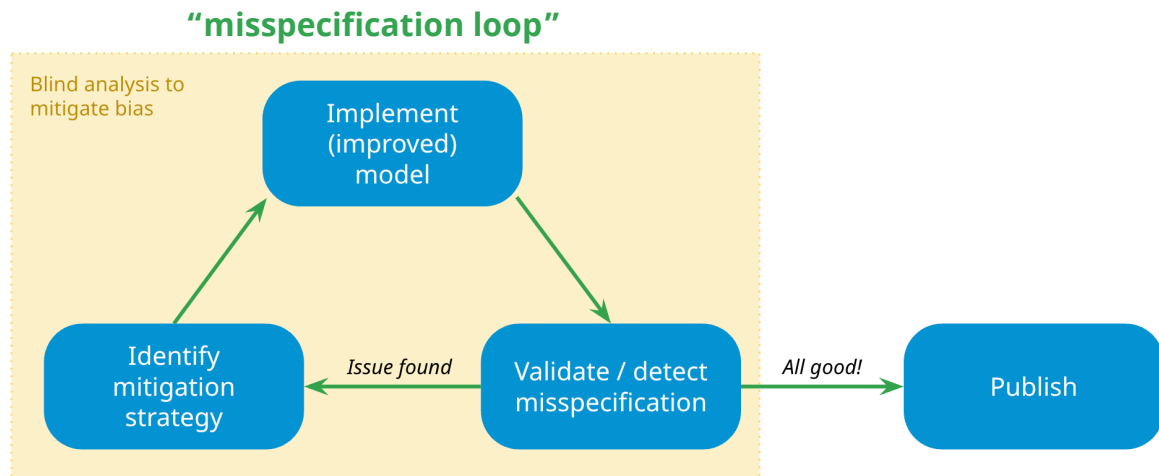
Four main categories of **mitigation strategies**

- **Cover with uncertainties**
 - Convert unknowns unknowns to known unknowns
- **Calibrate away**
 - Reweighting / flows / optimal transport, simulator tuning
- **Avoid misspecified features**
 - Feature selection, adversarial approaches, built-in symmetries
- **Data-driven approaches**
 - Empirical models, hybrid approaches

see [[VERaIPHY talk](#)] for more

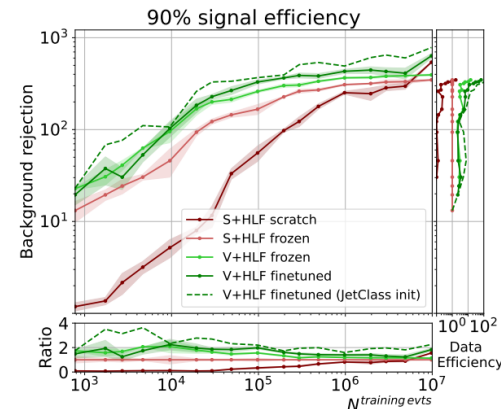
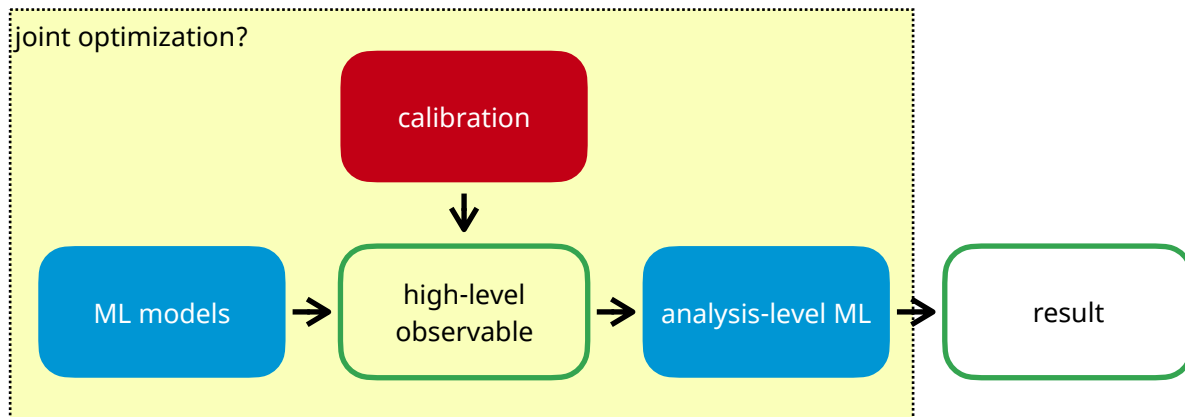


all possible data-generating processes in some extremely high-dimensional space

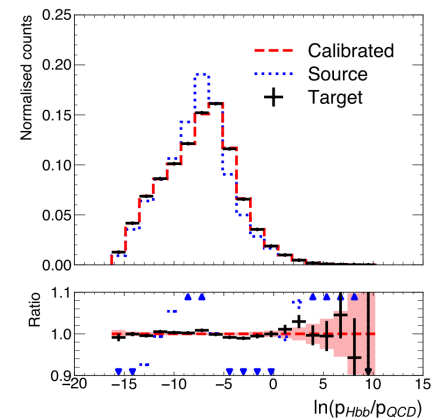


Sensitivity gains and calibration

- Our pipeline **factorizes into established steps**
 - avoid **duplication of effort**, benefit from single **central calibration**
 - **object reconstruction** as “foundation model” (fine-tune by picking working point)
- How does **calibration** work when we **optimize jointly**?
 - need to **calibrate high-dim** latent space [[arXiv:2507.08867](https://arxiv.org/abs/2507.08867), [ATLAS FTAG example](#)]

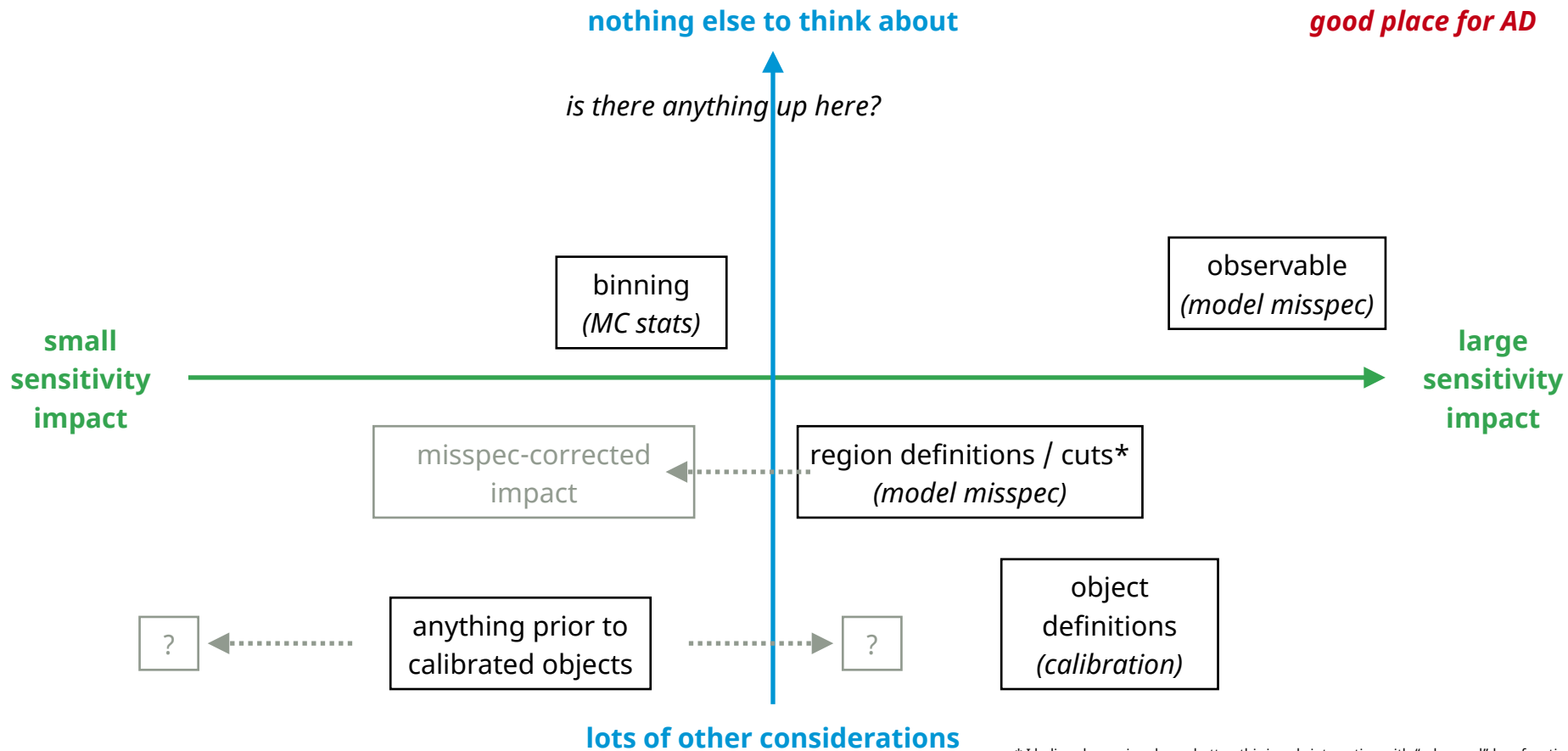


sensitivity gains with fine-tuning
[[arXiv:2401.13536](https://arxiv.org/abs/2401.13536)]



OT-based latent space calibration
[[arXiv:2507.08867](https://arxiv.org/abs/2507.08867)]

Analysis choices



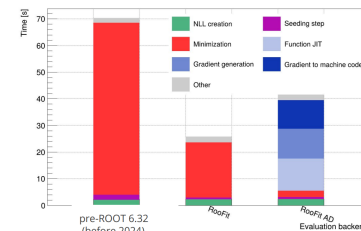
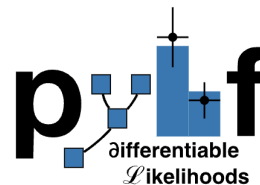
* I believe looser is ~always better, this is only interesting with "advanced" loss functions

What are the most promising use cases?

- Some applications are **very well motivated**, feasible and / or already in use for a while
 - they are typically **localized** to a specific step in an analysis chain
- **Full spectrum of “standard” ML** S-vs-B discrimination, background estimation techniques, SBI, ...
 - we use AD-optimized pieces all throughout the pipeline!

• Statistical inference

- **likelihood minimization** with exact gradients
- Hessian calculations without numerical problems
- sensitivity analysis / uncertainty decomposition [[arXiv:2307.04007](https://arxiv.org/abs/2307.04007)]
- and “next order” corrections like binning optimization [[arXiv:2601.07756](https://arxiv.org/abs/2601.07756), [Nitish's talk](#)]

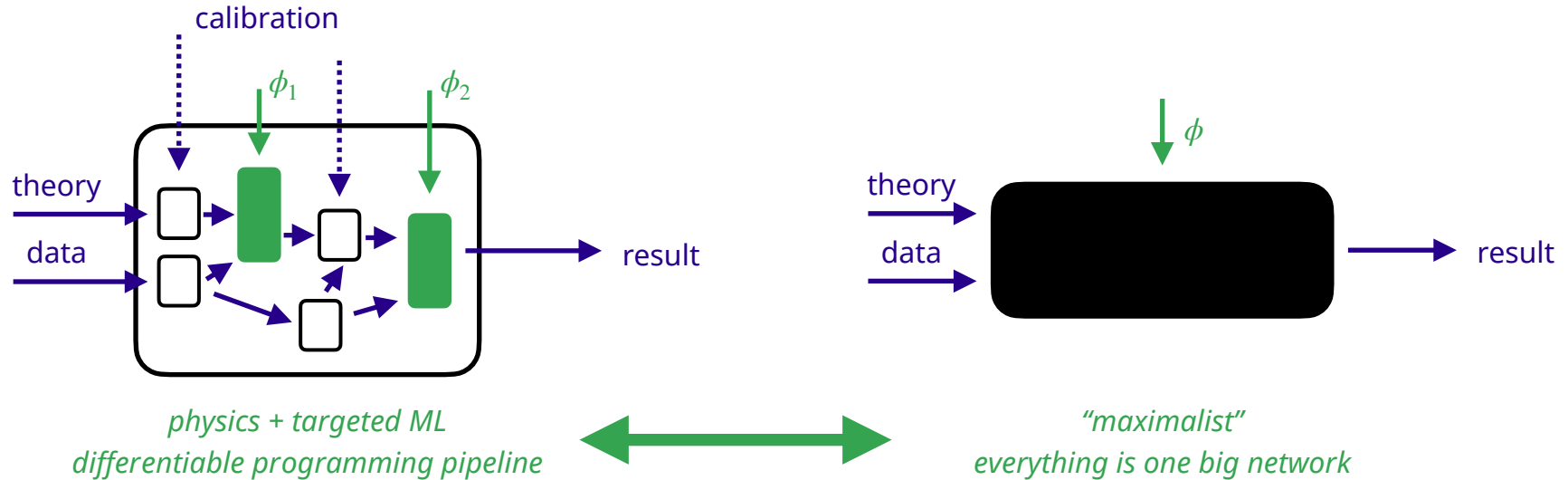


[Jonas Rembser]

- And maybe **LLMs-based agents**?

Possible future scenarios

- How much **inductive bias / physics-informed** design vs **blackbox optimization** do we want? [see also [Lukas' talk](#)]
 - we have decades of experience **factorizing** the problem
 - factorization also helps with **interpretability** and talking to **theory colleagues**



How will the future look?

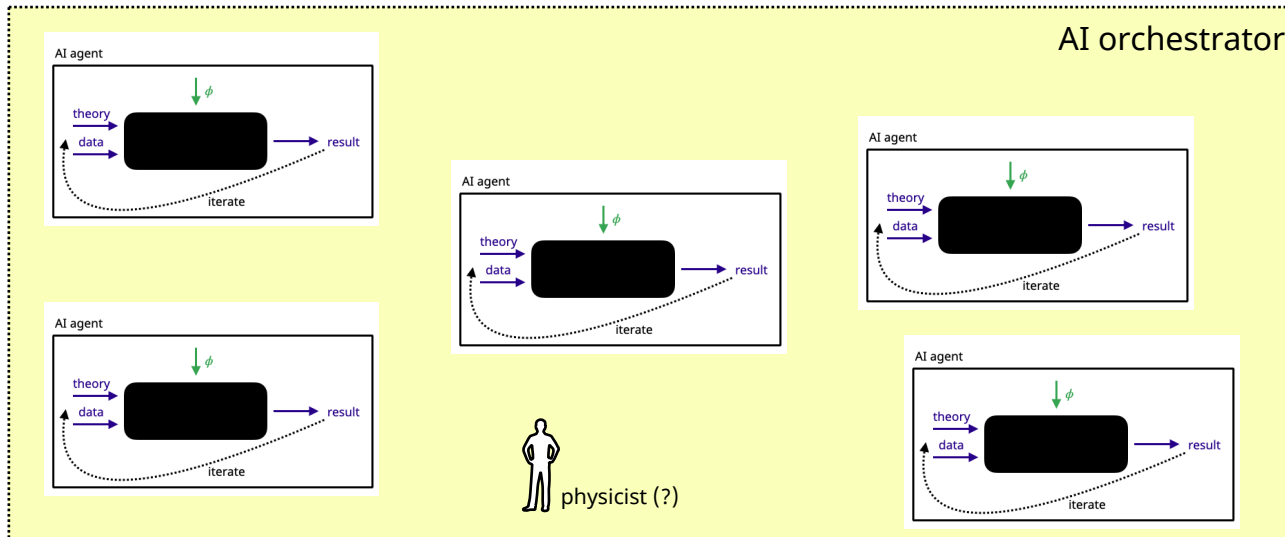
- Lots of interest in an “AI-native” future (e.g. [[arXiv:2602.17582](https://arxiv.org/abs/2602.17582)])

- Think about this as **physicists = agents** -> collaboration of agents?

- this is how we work as humans, should agents do too?
- formalized exchange protocol (papers, algorithms, ...), meetings as synchronization points, high-level planning

Building an AI-native Research Ecosystem for Experimental Particle Physics: A Community Vision

Tha Kjaebee Aarrestad, Alsa Abdelhamid, Haider Abidi, Jahred Adelman, Jennifer Adelman-McCarthy, Shuchin Aeron, Gurvita Agarwal, Usman Ali, Cristiano Alpigiani, Omar Alterkait, Mohamed Aly, Oz Amram, Saeed Ansari Farid, Aram Apyan, John Arrington, Marvin Ascencio-Sosa, Mohammad Atif, Aneesa Avasthi, Muhammad Bilal Azam, Bhim Bam, Joshua Barrow, Rainer Bartoldus, Amit Bashyal, Aashwin Basnet, Ayse Bat, Lothar A. T. Bauerdick, John Beacom, Chris Bee, Michael Beigel, Matthew Bellis, Rene Bellwied, Rakitha Beminawattha, Gabriele Benelli, Douglas Benjamin, Catrin Bernius, Binod Bhandari, Avinay Bhat, Meghna Bhattacharya, Saaptarna Bhattacharya, Prajita Bhattarai, Sudip Bhattarai, Wahid Bhimji, Jianming Bian, Burak Bilki, Mary Bishai, Kevin Black, Kenneth Bloom, Brian Bockelman, Johan Sebastian Bonilla Castro, Tulika Bose, Nilay Bostan, Othmane Bouhali, Dimitri Bourlikov, Dominic Brailsford, Gusaaf Brooijmans, Elizabeth Brost, Maria Brigidia Brunetti, Quentin Buat, Brendon Bullard, Jackson Burzynski, Paolo Calafura, Rodolfo Capdevilla, Fabian Andres Castaño Usuga, Raquel Castillo Fernandez, Fabio Catalano, Viviana Cavaliere, Flavio Cavanna, Giuseppe Cerati, Aidan Chambers, Maria Chamizo-Llata, Phillip Chang, Andrew Chappell, Arghya Chattopadhyay, Sergel Chekanov, Jian-ping Chen, Yi Chen, Zhengyang Chen, J. Taylor Childers, Hector Chinchay, Yuan-Tang Chou, Tasnuva Chowdhury, Neil Christensen, Wonyong Chung, Rafael Coelho Lopes de Sa, Simon Corradi, Kyle Cramer, Matteo Cremonesi, Roy Cruz, Mate Csanád, Mariarosaria D'Alfonso, Carlo Dallapiccola, Daine Danielson, Sridhara Dasu, Gavin Davies, Kaushik De, Patrick de Perio, Klaus Dehmel, Marco Del Tutto, Carlos Ruben Dell'Aquila, Sarah Demers et al. (359 additional authors not shown)



Conclusion / what to focus on?

- **“End-to-end” optimization** through a large part of the full pipeline is probably not the right target
- **Good calibration strategies** are key and beneficial not just for any new differentiable programming approaches!
 - this seems **increasingly important**
- **Consensus** for some areas
 - **“traditional” ML** approaches are integral part of HEP
 - autodiff gradient-based **likelihood fitting** makes a lot of sense
- It remains interesting to think about **non-gradient-based optimization** approaches, including the role of LLMs
 - clever injection of **domain knowledge** remains very powerful, e.g. factorization in SBI

Backup

Systematics + ML: wrong vs suboptimal

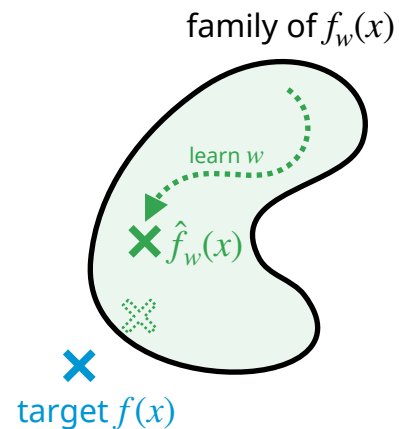
- **Model misspecification** and (lack of) **systematic uncertainties** can make our results **wrong** and / or **suboptimal**

- **Avoiding wrong results**

- incorporate and **propagate** all relevant sources of **systematic uncertainty** through chain
 - requires understanding which sources are relevant

- **Striving towards optimal results**

- possible limitations due to **training dataset size**, **model capacity**, **domain shift**
- e.g. “are we using a good summary statistic?”
- often **ML training + systematic uncertainties are factorized**, generally non-optimal
 - instead: e.g. data augmentation, parameterized models, ...



Systematic variations

- Need to model $\nu(\vec{k}, \vec{\theta})$ for any value of nuisance parameters $\vec{\theta}$ encoding systematic uncertainties

- **Ideal case:** just run simulator for any value of $\vec{\theta}$

- not computationally feasible in practice

- **Instead:** pick some values & **interpolate**

- in practice we use on-axis variations

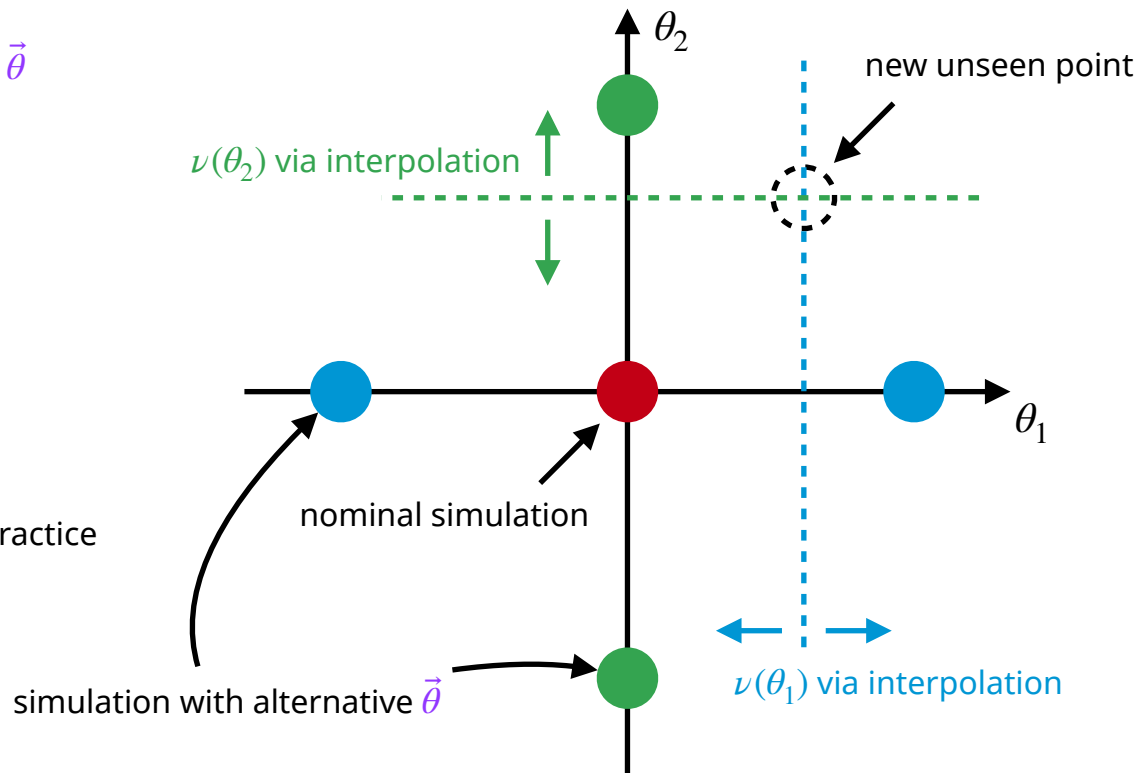
- variations typically are “one at a time”

- Lots of **assumptions** here that we rely on in practice

- where to simulate

- interpolation choice

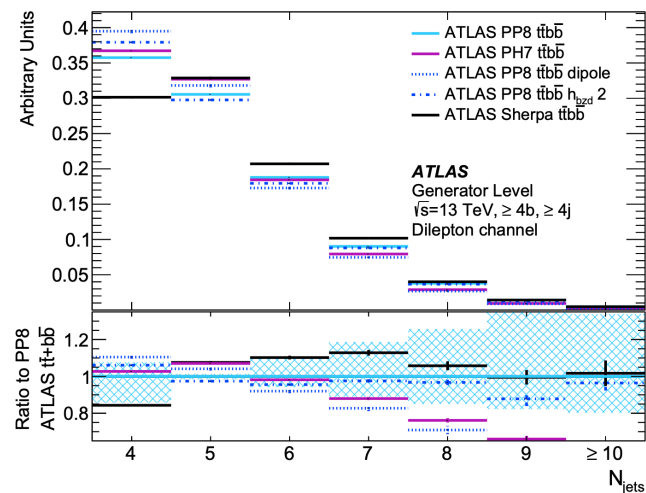
- effects factorize



Complication: two-point systematics

- Sometimes have cases where **variations in simulator chain are discrete**
 - e.g. **choice of one simulator vs alternative**
- Typical treatment: **interpolate to treat as continuous, symmetrize**
 - **lots of assumptions** here, but need to make a choice to profile
- Especially **tricky to deal with** when these play a large role
 - concerns about **overly constraining** uncertainty of nuisance parameter
 - best-fit model prediction may lie away from both choices

modeling choices for main background of $t\bar{t}(b\bar{b})$



LHCWGW-2022-003

