Track 3 Summary

Computations in Theoretical Physics: Techniques and Methods

Anke Biekötter - Leonardo Cosmai -
Joshua Davies - Latifa Elouadrhiri

ACAT 2022 -
24-28 October 2022
Villa Romanazzi Carducci, Bari, Italy

ACAT 2022


ALTROCONSUMO
APPROVATO
Pubblicato il 09/2022
PRODOTTI PIÙ ECONOMICI
DISCOUNT SALVAPREZZO IN ITALIA
Eurospin approvato da Altroconsumo come discount salvaprezzo in Italia, per la categoria prodotti più economici, classifica unica iper, super e discount.
$E=mc^2$
Eurospin Massima Convenienza
Prodotti di qualità alla massima convenienza tutti i giorni: questa è la Spesa intelligente.
EURO Spin

ACAT 2022

Thank you for your contributions!

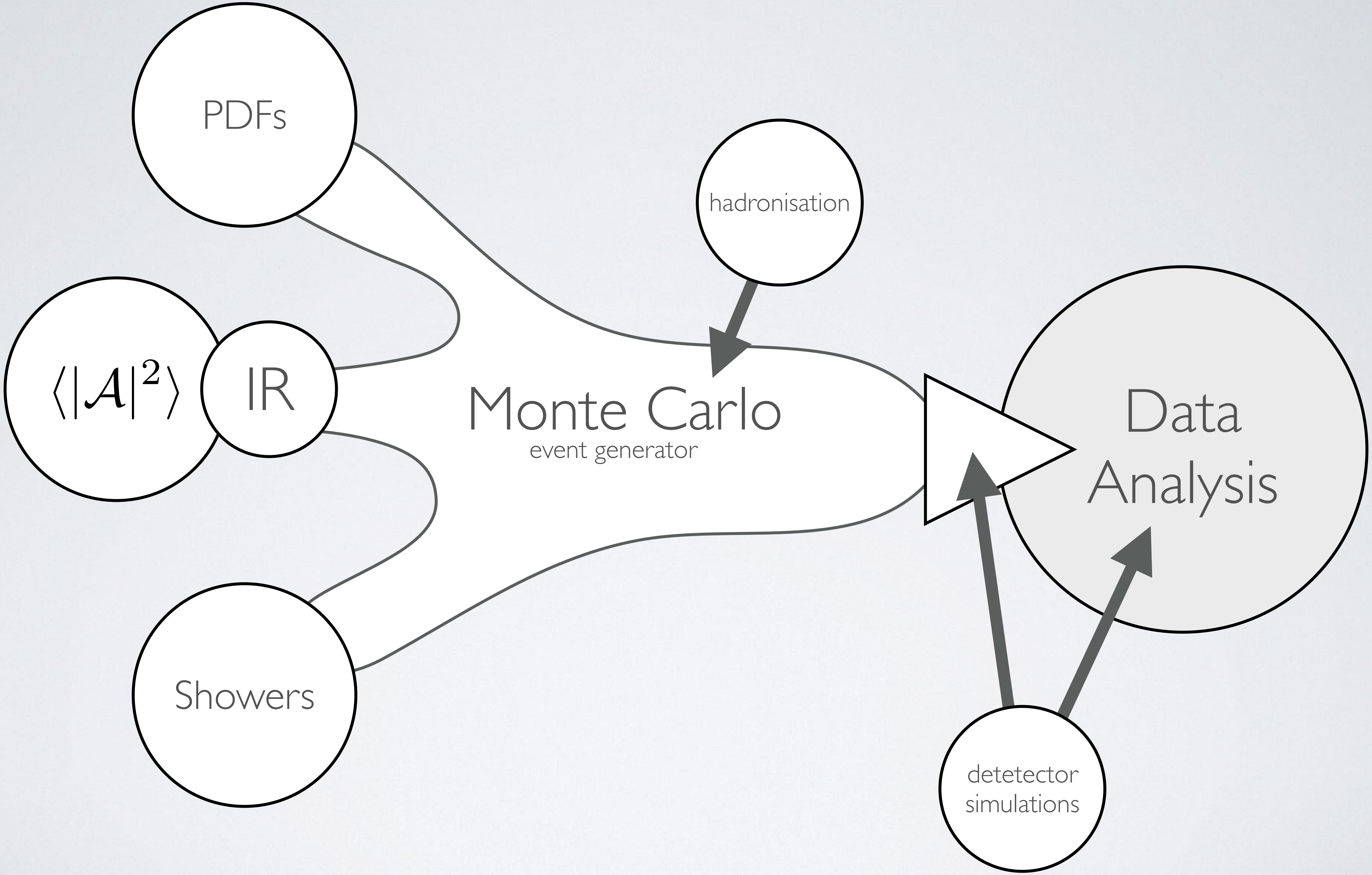
Track 3 Highlights

- Monte Carlo generation
- Precision frontier
- Beyond Standard Model physics
- Towards Quantum Computing

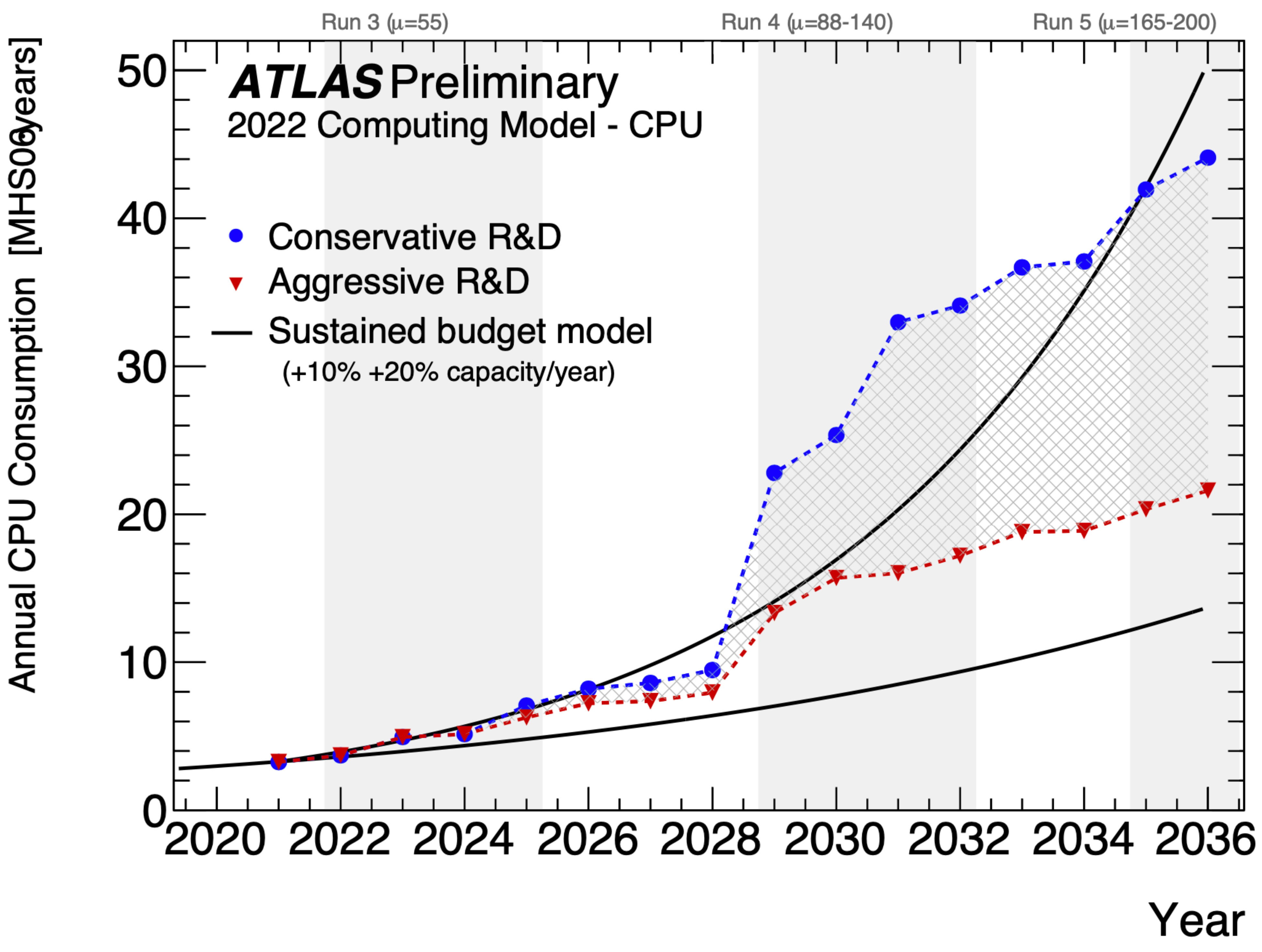
This is a biased selection

Simon Badger

from theory to experiment



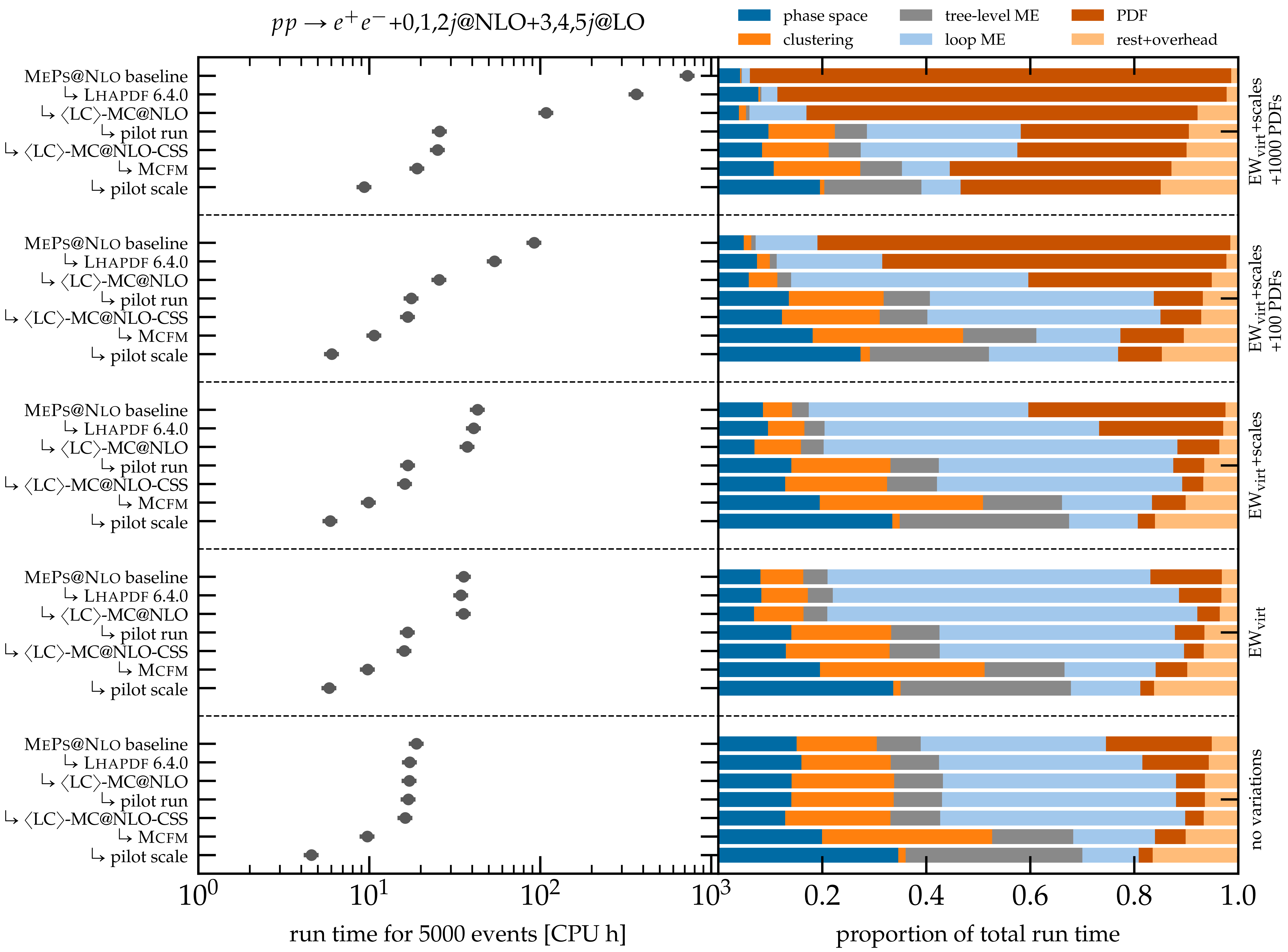
Speeding up Monte Carlo event generators

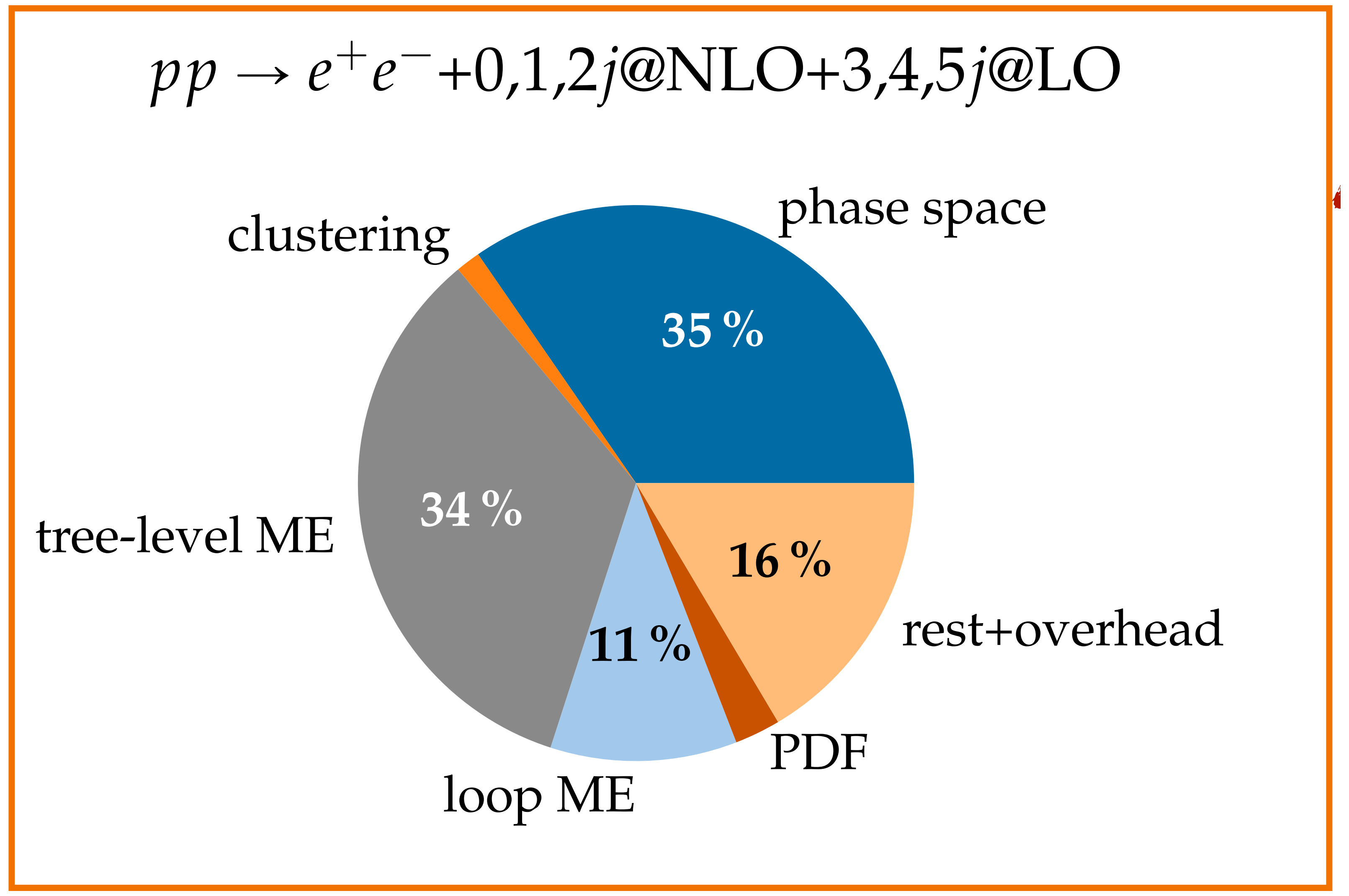


- Performance analysis
  - Pilot runs (what do we need when?)
- New architectures - GPUs, vector CPUs
  - Portability (Kokkos, Alpaka, ...)
- Physics ideas and analytic results

Breakdown of CPU budget in V+jets

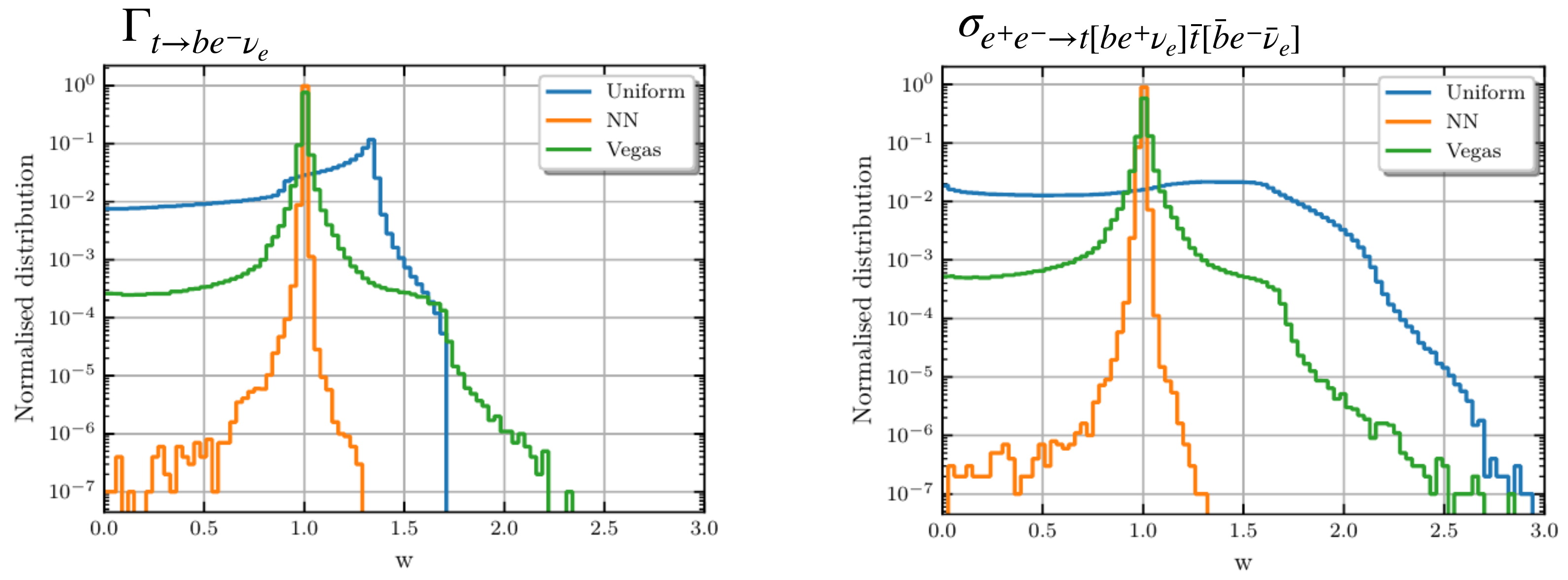
Christian Gütschow




$pp \to e^+e^-$+0,1,2j@NLO+3,4,5j@LO
phase space
35 %
clustering
tree-level ME
34 %
16 %
rest+overhead
11 %
PDF
loop ME

Neural Importance Sampling – Results

remember: aim for $w = f/g \approx 1$, i.e. peaked distribution of w

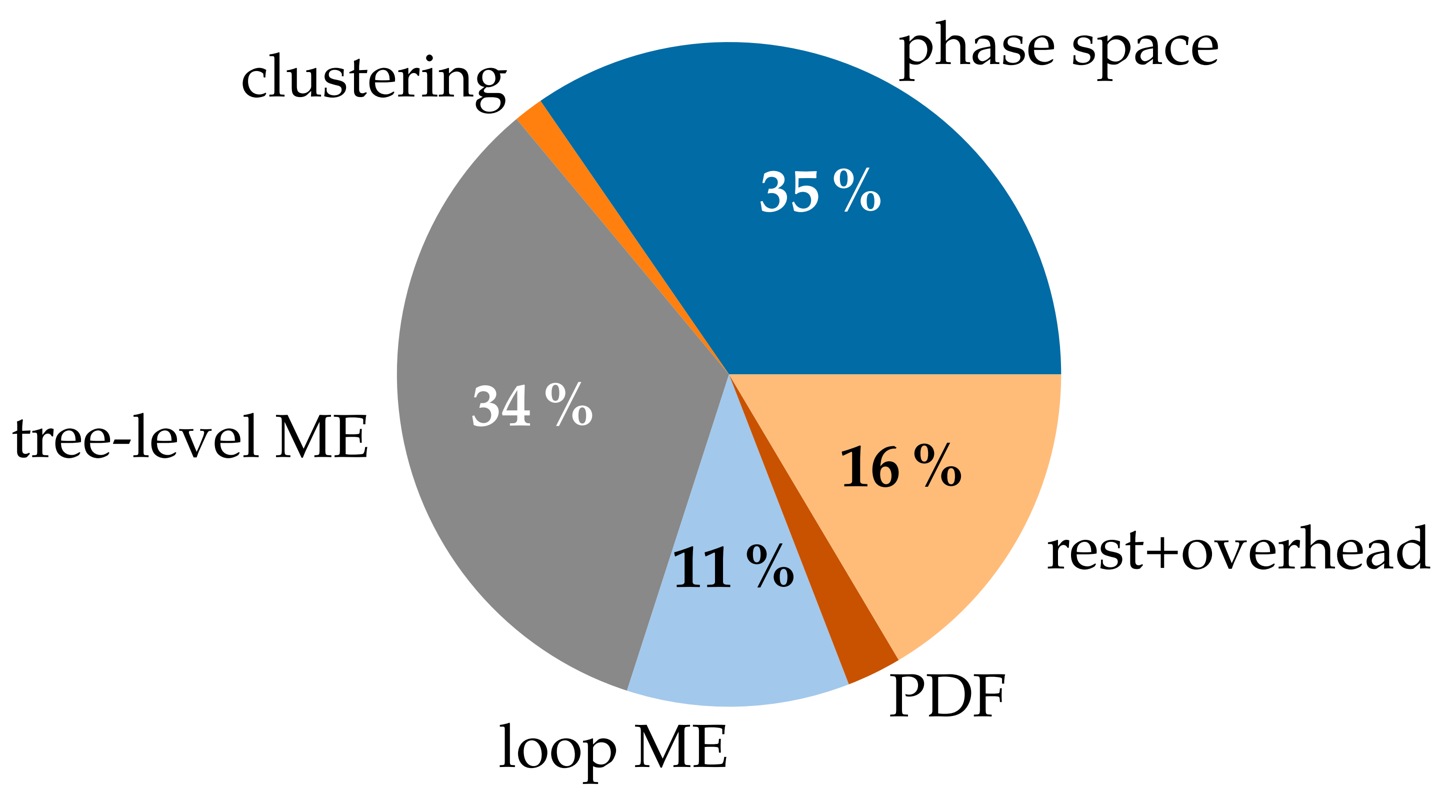


	top decays		top-pair production		$gg \to 3g$		$gg \to 4g$	
Sample	ϵ_{uw}	E_N [GeV]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]
Uniform	59 %	0.1679(2)	35 %	1.5254(8)	3.0 %	24806(55)	2.7 %	9869(20)
Vegas	50 %	0.16782(4)	40 %	1.5251(1)	27.7 %	24813(23)	31.8 %	9868(10)
NN	84 %	0.167865(5)	78 %	1.52531(2)	64.3 %	24847(21)	48.9 %	9859(10)

- Smaller impact for more complicated (multi-channel) processes, similar in [Gao et al., Phys. Rev. D 101 (2020) no.7, 076002]
- GPU evaluation of MEs desirable for efficient training cf. talks by M. Knobbe, R. Wang and A. Valassi
- Alternative to ML-*assisted* phase space sampling: directly learn target distribution using autoregressive flows, GANs, VAEs [Stienen and Verheyen SciPost Phys. **10**, 038 (2021)], [Butter, Plehn and Winterhalder, SciPost Phys. **7** (6), 075 (2019)], [Sipio et al. JHEP **08**, 110 (2019)], [Otten et al. Nature Commun. **12** (1), 2985 (2021)], [Choi and Lim, J. Korean Phys. Soc. **78** (6), 482 (2021)]
    - if no surjectivity guarantee → might miss tails of distributions and get small bias in overall integration result

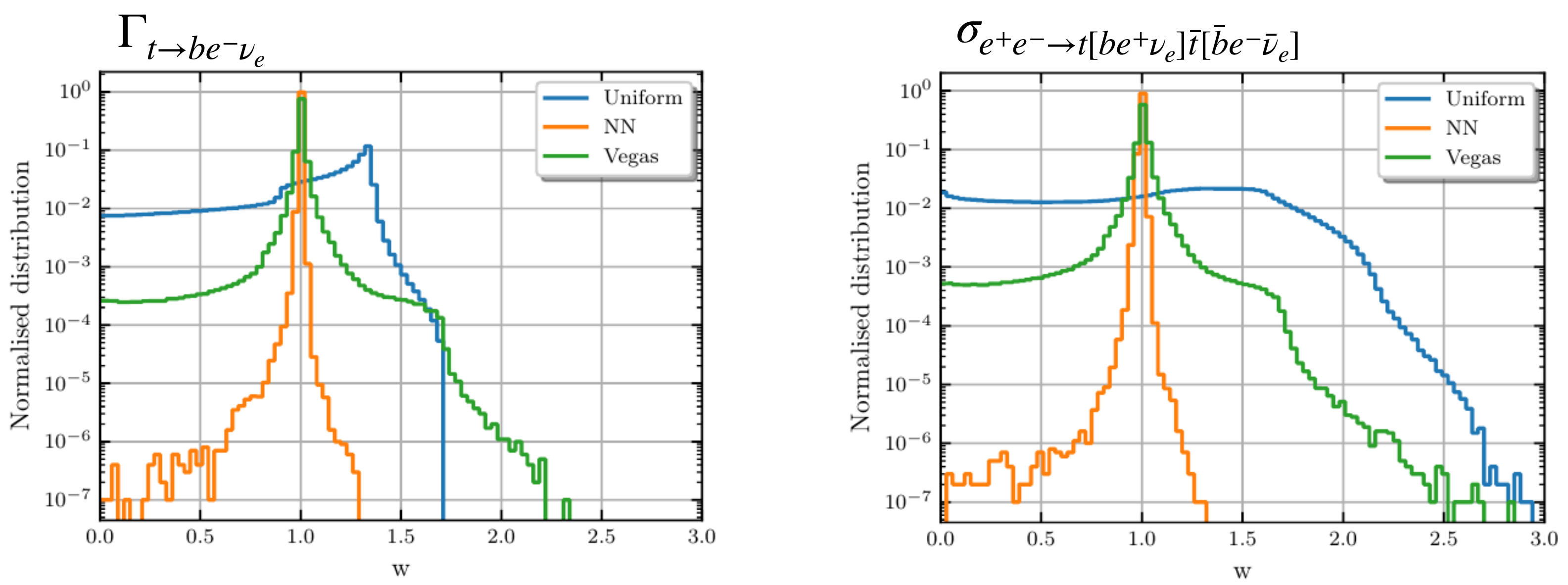
Enrico Bothmann

$pp \to e^+e^- + 0,1,2j@\mathrm{NLO} + 3,4,5j@\mathrm{LO}$



Neural Importance Sampling – Results

remember: aim for $w = f/g \approx 1$, i.e. peaked distribution of w

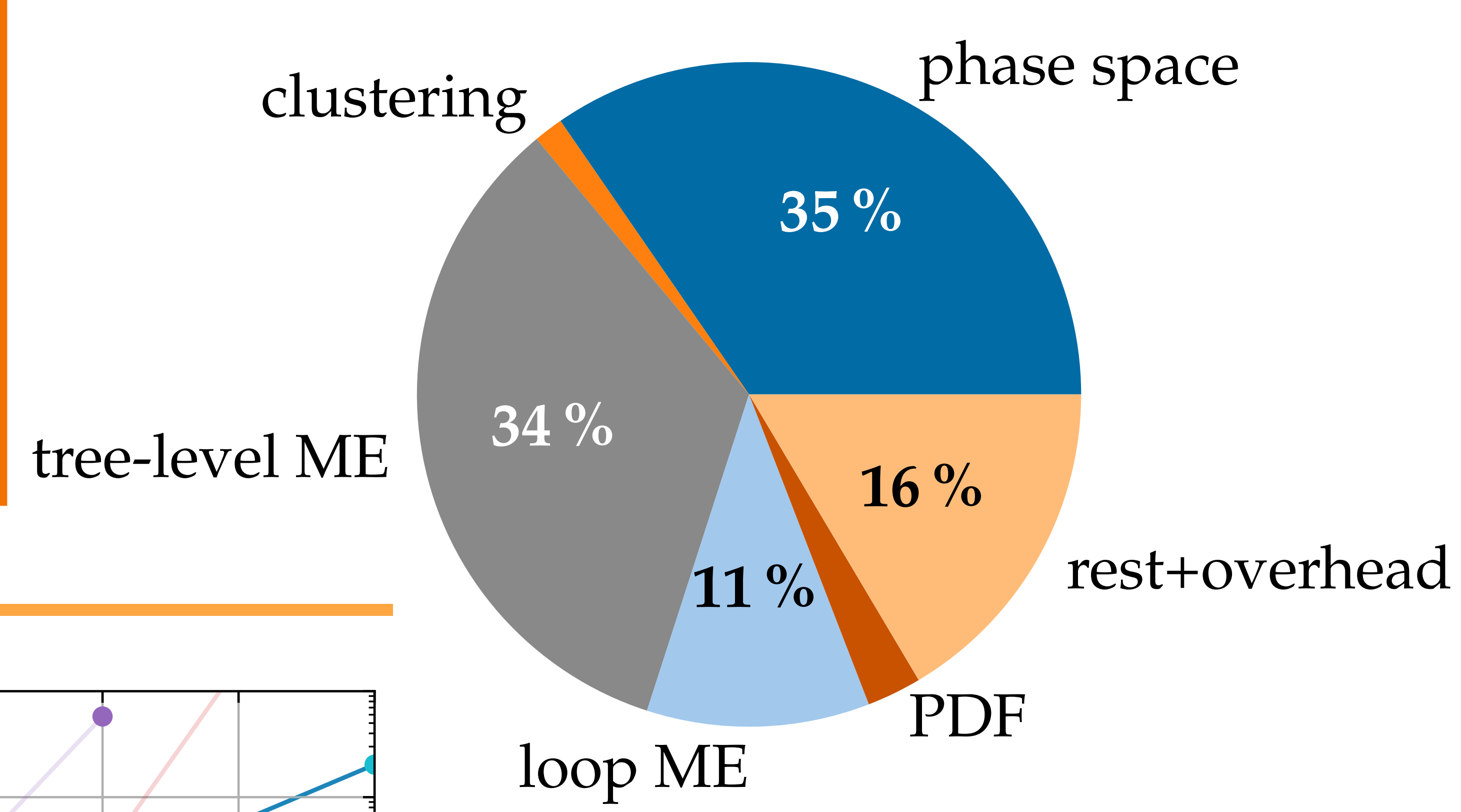


	top decays		top-pair production		$gg \to 3g$		$gg \to 4g$	
Sample	ϵ_{uw}	E_N [GeV]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]
Uniform	59 %	0.1679(2)	35 %	1.5254(8)	3.0 %	24806(55)	2.7 %	9869(20)
Vegas	50 %	0.16782(4)	40 %	1.5251(1)	27.7 %	24813(23)	31.8 %	9868(10)
NN	84 %	0.167865(5)	78 %	1.52531(2)	64.3 %	24847(21)	48.9 %	9859(10)

- Smaller impact for more complicated (multi-channel) processes, similar in [Gao et al., Phys. Rev. D 101 (2020) no.7, 076002]
- GPU evaluation of MEs desirable for efficient training cf. talks by M. Knobbe, R. Wang and A. Valassi
- Alternative to ML-*assisted* phase space sampling: directly learn target distribution using autoregressive flows, GANs, VAEs [Stienen and Verheyen SciPost Phys. **10**, 038 (2021)], [Butter, Plehn and Winterhalder, SciPost Phys. **7** (6), 075 (2019)], [Sipio et al. JHEP **08**, 110 (2019)], [Otten et al. Nature Commun. **12** (1), 2985 (2021)], [Choi and Lim, J. Korean Phys. Soc. **78** (6), 482 (2021)]
  - if no surjectivity guarantee → might miss tails of distributions and get small bias in overall integration result

Enrico Bothmann

$pp \to e^+e^- + 0{,}1{,}2j$@NLO$+3{,}4{,}5j$@LO

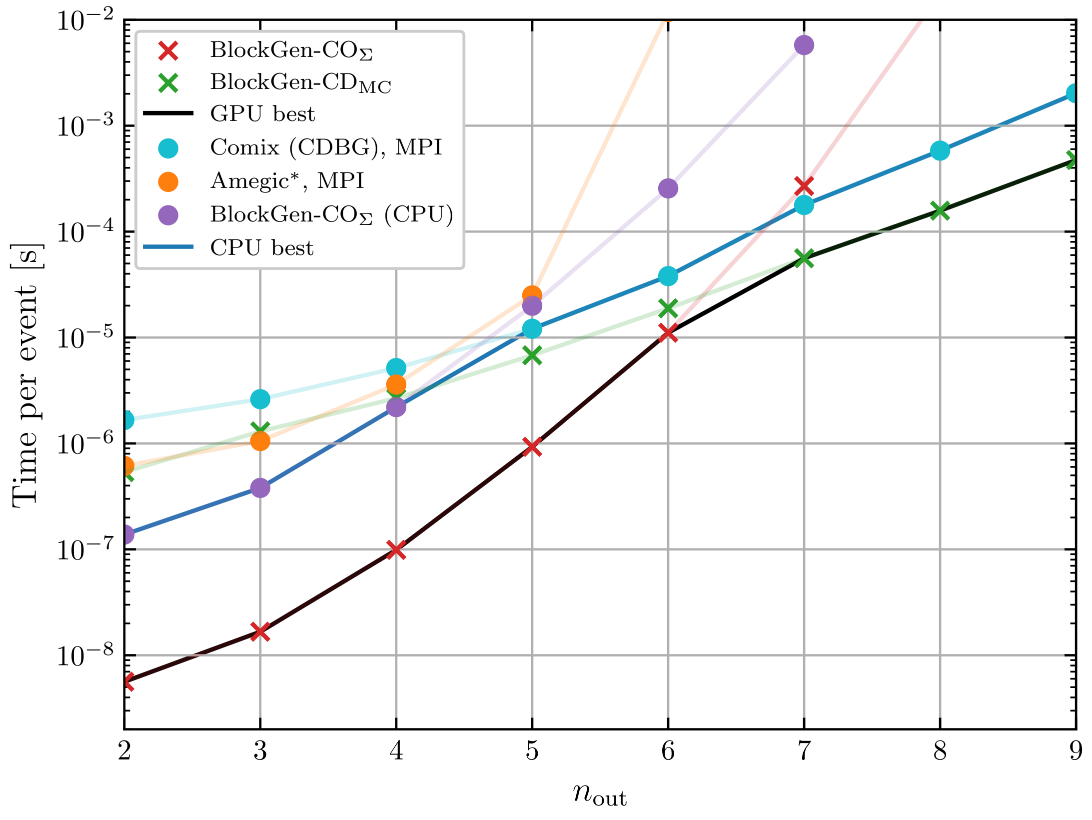


Max Knobbe

3rd Component: The Color Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

- Benchmark performance for gluon-only process
- Relevant test, since as-many-gluon-as-possible amplitudes make up largest portion of computing time for jet-processes
- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors**



ME on a GPU — Andrea Valassi

MadEvent/CUDA for $gg \to t\bar{t}gg$ (improved at ACAT2022)

ICHEP2022

		madevent				standalone
CUDA grid size		8192				524288
$gg \to t\bar{t}gg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]		
Fortran	double	58.3 = 5.2 + 53.1	1.55E3 (=1.0)	1.70E3 (=1.0)	—	—
CUDA	double	6.1 = 5.7 + 0.36	1.49E4 (x9.6)	2.54E5 (x149)	2.51E5	4.20E5 (x247)
CUDA	float	5.7 = 5.4 + 0.24	1.59E4 (x10.3)	3.82E5 (x224)	3.98E5	8.75E5 (x515)

Reduced the overhead from scalar Fortran MadEvent overhead from 10% to 5% of initial Fortran (improved handling of MLM merging)
Maximum allowed overall speedup from Amdahl's law is now increased from x10 to x20 - which we do achieve

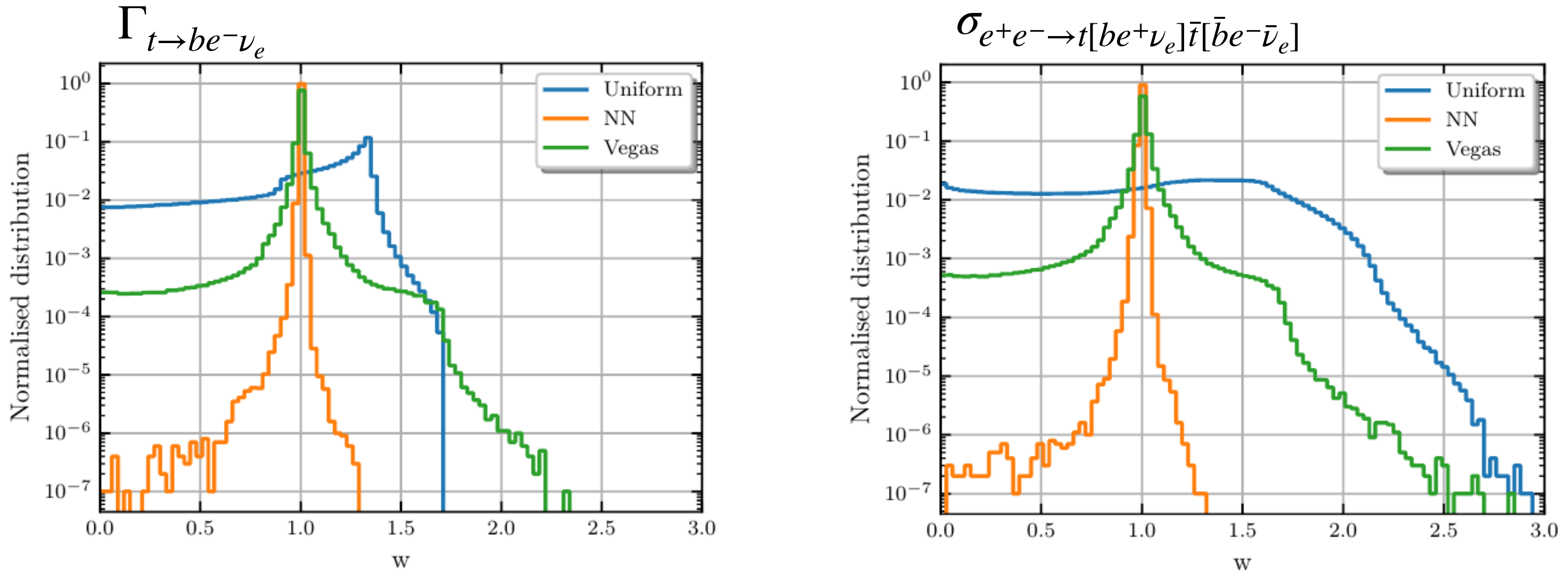
ACAT2022

		madevent				standalone
CUDA grid size		8192				524288
$gg \to t\bar{t}gg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]		
Fortran	double	55.4 = 2.4 + 53.0	1.63E3 (=1.0)	1.70E3 (=1.0)	—	—
CUDA	double	2.9 = 2.6 + 0.35	3.06E4 (x18.8)	2.60E5 (x152)	2.62E5	4.21E5 (x
CUDA	float	2.8 = 2.6 + 0.24	3.24E4 (x19.9)	3.83E5 (x225)	3.96E5	8.77E5 (x

Enrico Bothmann

Neural Importance Sampling – Results

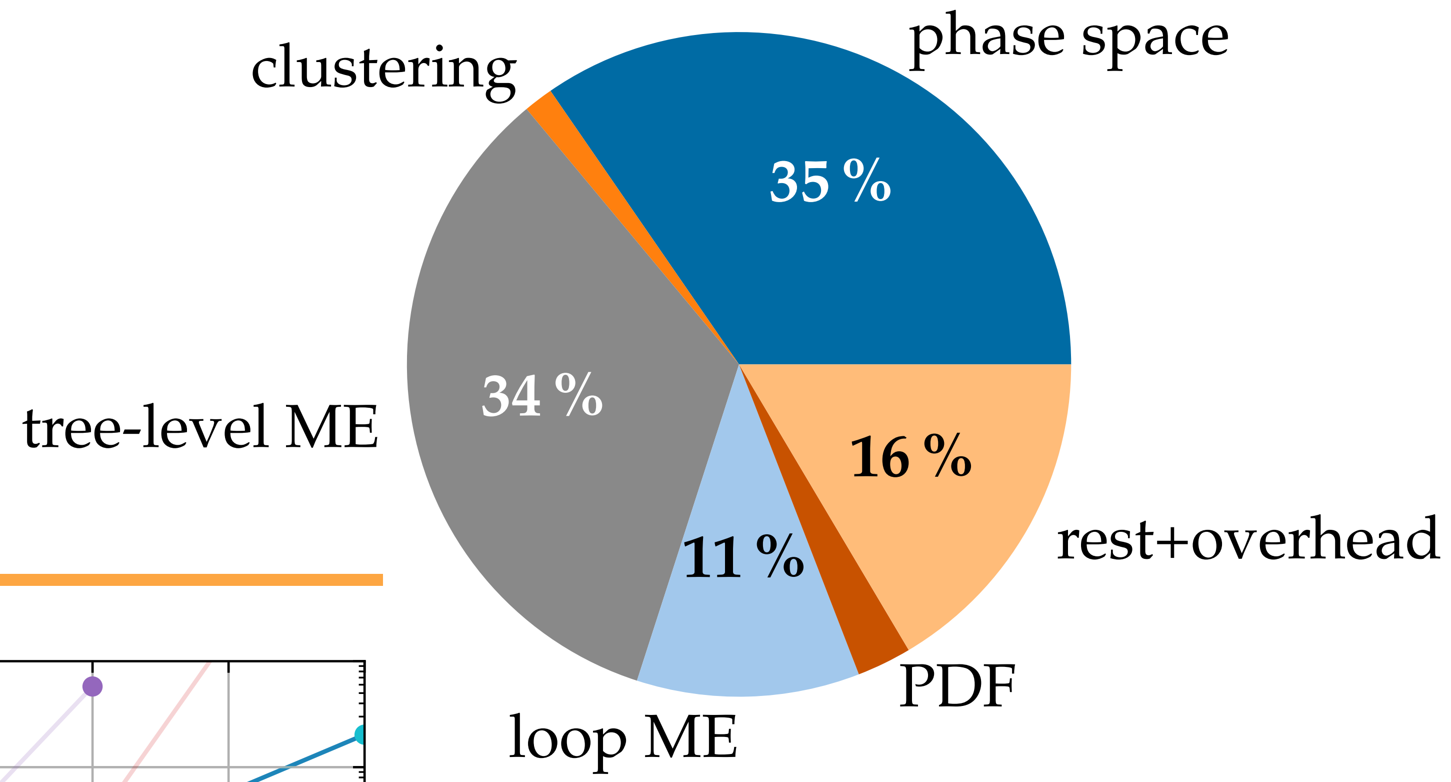
remember: aim for $w = f/g \approx 1$, i.e. peaked distribution of w



	top decays		top-pair production		$gg \to 3g$		$gg \to 4g$	
Sample	ϵ_{uw}	E_N [GeV]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]
Uniform	59 %	0.1679(2)	35 %	1.5254(8)	3.0 %	24806(55)	2.7 %	9869(20)
Vegas	50 %	0.16782(4)	40 %	1.5251(1)	27.7 %	24813(23)	31.8 %	9868(10)
NN	84 %	0.167865(5)	78 %	1.52531(2)	64.3 %	24847(21)	48.9 %	9859(10)

- Smaller impact for more complicated (multi-channel) processes, similar in [Gao et al., Phys. Rev. D 101 (2020) no.7, 076002]
- GPU evaluation of MEs desirable for efficient training cf. talks by M. Knobbe, R. Wang and A. Valassi
- Alternative to ML-*assisted* phase space sampling: directly learn target distribution using autoregressive flows, GANs, VAEs [Stienen and Verheyen SciPost Phys. **10**, 038 (2021)], [Butter, Plehn and Winterhalder, SciPost Phys. **7** (6), 075 (2019)], [Sipio et al. JHEP **08**, 110 (2019)], [Otten et al. Nature Commun. **12** (1), 2985 (2021)], [Choi and Lim, J. Korean Phys. Soc. **78** (6), 482 (2021)]
  - if no surjectivity guarantee → might miss tails of distributions and get small bias in overall integration result

$$pp \to e^+e^- + 0,1,2j@NLO + 3,4,5j@LO$$

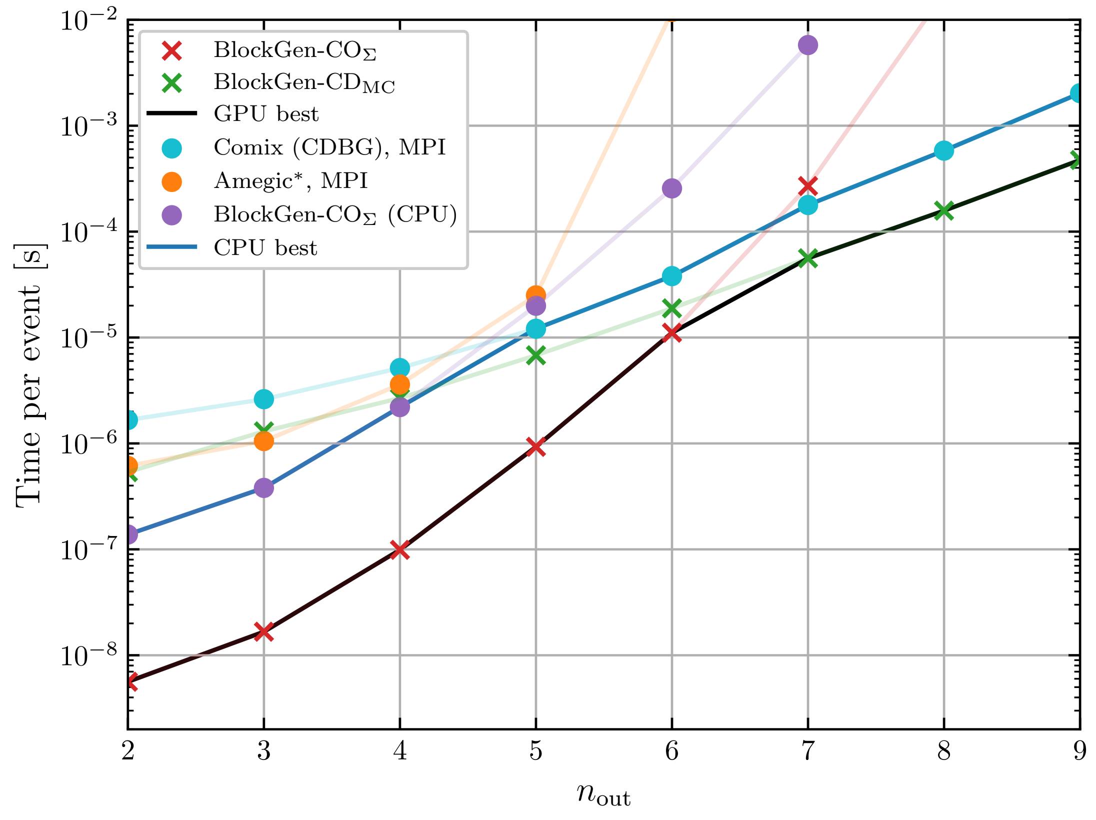


Max Knobbe

3rd Component: The Color Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

- Benchmark performance for gluon-only process
- Relevant test, since as-many-gluon-as-possible amplitudes make up largest portion of computing time for jet-processes
- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors**



ME on a GPU — Andrea Valassi

MadEvent/CUDA for $gg\to t\bar{t}gg$ (improved at ACAT2022)

ICHEP2022

CUDA grid size		madevent 8192			standalone 524288	
$gg\to t\bar{t}gg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]		
Fortran	double	58.3 = 5.2 + 53.1	1.55E3 (=1.0)	1.70E3 (=1.0)	—	—
CUDA	double	6.1 = 5.7 + 0.36	1.49E4 (x9.6)	2.54E5 (x149)	2.51E5	4.20E5 (x247)
CUDA	float	5.7 = 5.4 + 0.24	1.59E4 (x10.3)	3.82E5 (x224)	3.98E5	8.75E5 (x515)

Reduced the overhead from scalar Fortran MadEvent overhead from 10% to 5% of initial Fortran (improved handling of MLM merging)

Maximum allowed overall speedup from Amdahl's law is now increased from x10 to x20 - which we do achieve

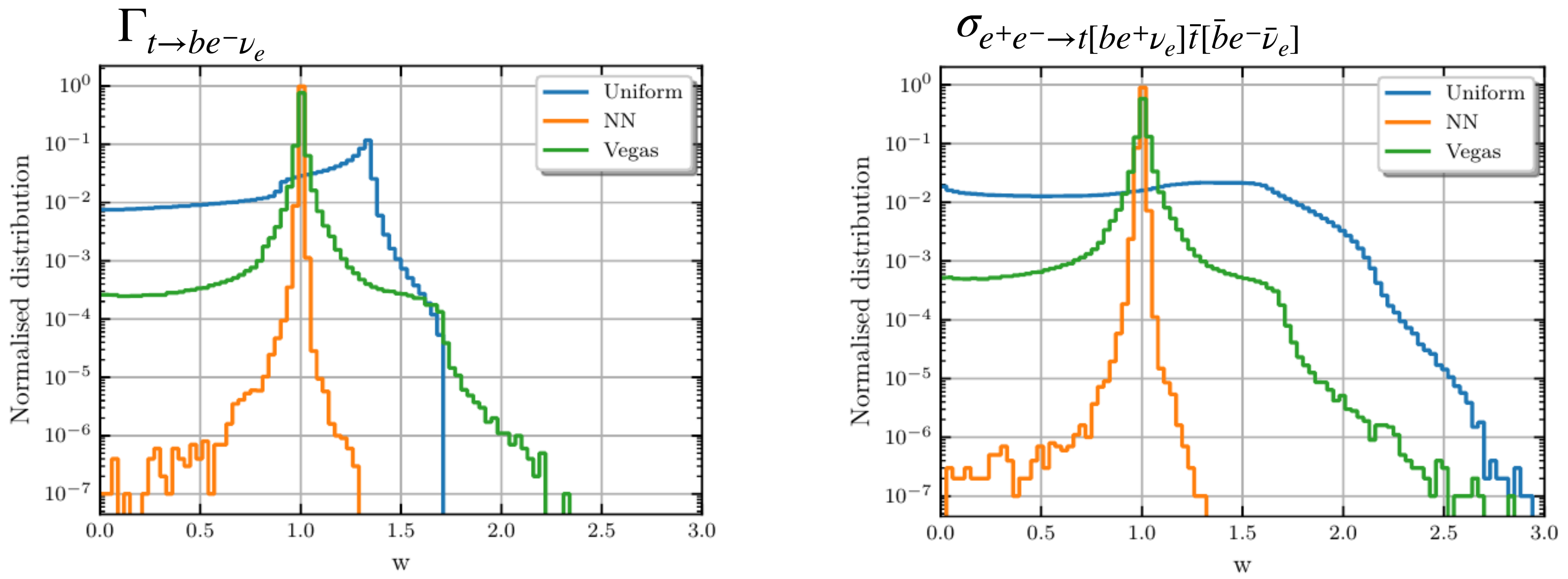
ACAT2022

CUDA grid size		madevent 8192			standalone 524288	
$gg\to t\bar{t}gg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]		
Fortran	double	55.4 = 2.4 + 53.0	1.63E3 (=1.0)	1.70E3 (=1.0)	—	—
CUDA	double	2.9 = 2.6 + 0.35	3.06E4 (x18.8)	2.60E5 (x152)	2.62E5	4.21E5 (x
CUDA	float	2.8 = 2.6 + 0.24	3.24E4 (x19.9)	3.83E5 (x225)	3.96E5	8.77E5 (x.

Enrico Bothmann

Neural Importance Sampling – Results

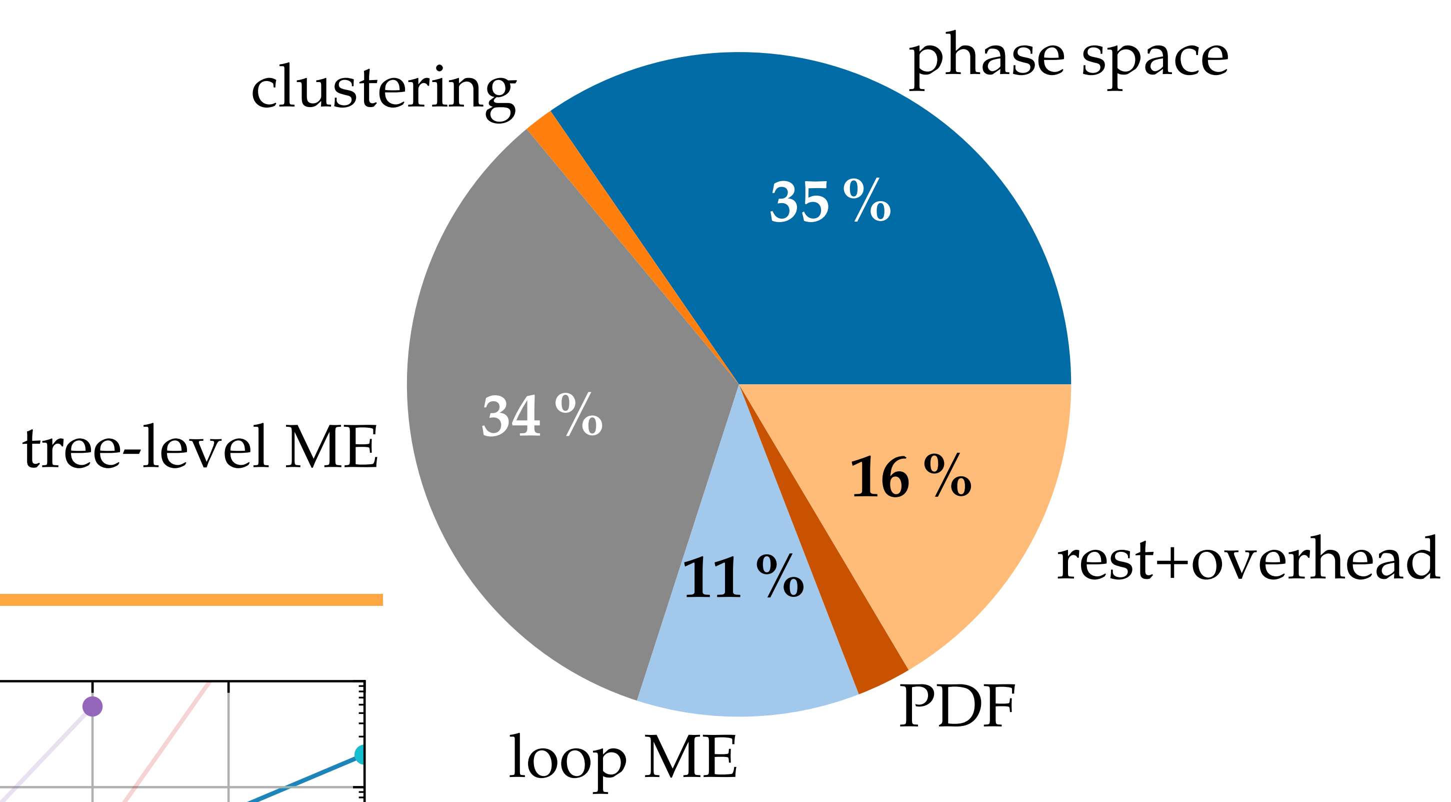
remember: aim for $w = f/g \approx 1$, i.e. peaked distribution of w



	top decays		top-pair production		$gg\to 3g$		$gg\to 4g$	
Sample	ϵ_{uw}	E_N [GeV]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]	ϵ_{uw}	E_N [fb]
Uniform	59 %	0.1679(2)	35 %	1.5254(8)	3.0 %	24806(55)	2.7 %	9869(20)
Vegas	50 %	0.16782(4)	40 %	1.5251(1)	27.7 %	24813(23)	31.8 %	9868(10)
NN	84 %	0.167865(5)	78 %	1.52531(2)	64.3 %	24847(21)	48.9 %	9859(10)

- Smaller impact for more complicated (multi-channel) processes, similar in [Gao et al., Phys. Rev. D 101 (2020) no.7, 076002]
- GPU evaluation of MEs desirable for efficient training cf. talks by M. Knobbe, R. Wang and A. Valassi
- Alternative to ML-*assisted* phase space sampling: directly learn target distribution using autoregressive flows, GANs, VAEs [Stienen and Verheyen SciPost Phys. **10**, 038 (2021)], [Butter, Plehn and Winterhalder, SciPost Phys. **7** (6), 075 (2019)], [Sipio et al. JHEP **08**, 110 (2019)], [Otten et al. Nature Commun. **12** (1), 2985 (2021)], [Choi and Lim, J. Korean Phys. Soc. **78** (6), 482 (2021)]
  - if no surjectivity guarantee → might miss tails of distributions and get small bias in overall integration result

$pp\to e^+e^-$+0,1,2j@NLO+3,4,5j@LO

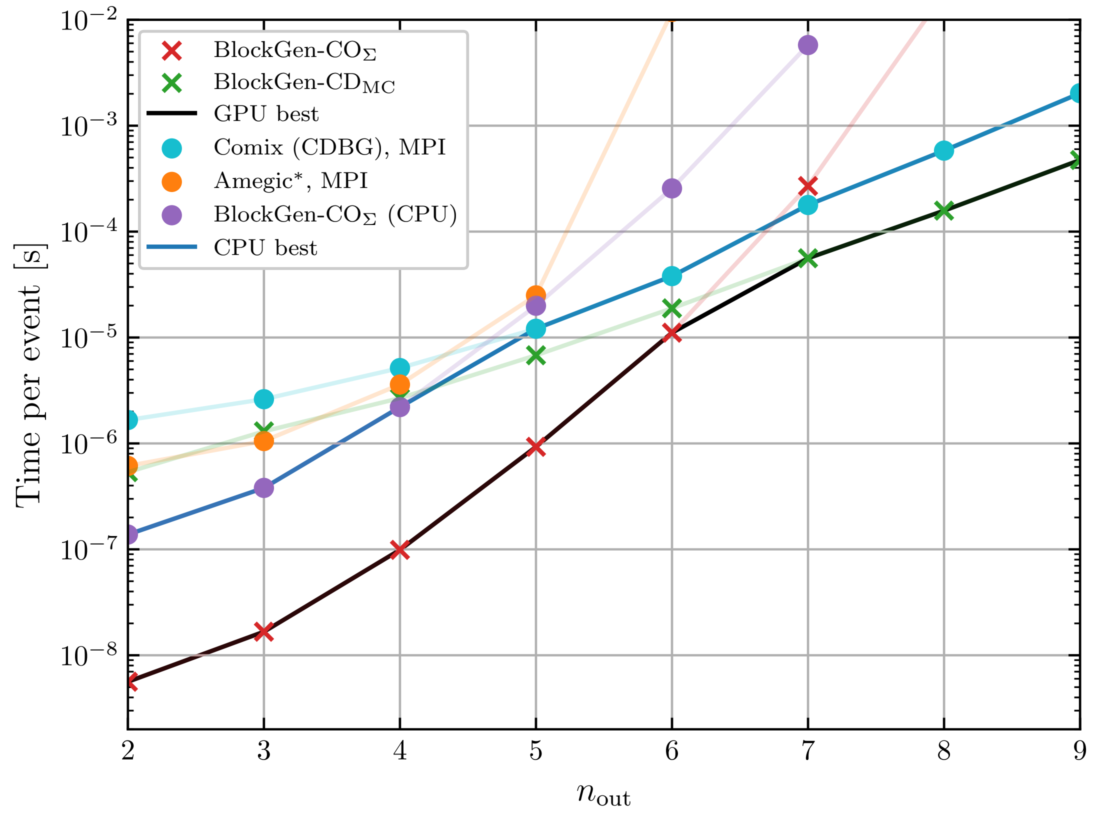


Max Knobbe

3rd Component: The Color Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

- Benchmark performance for gluon-only process
- Relevant test, since as-many-gluon-as-possible amplitudes make up largest portion of computing time for jet-processes
- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

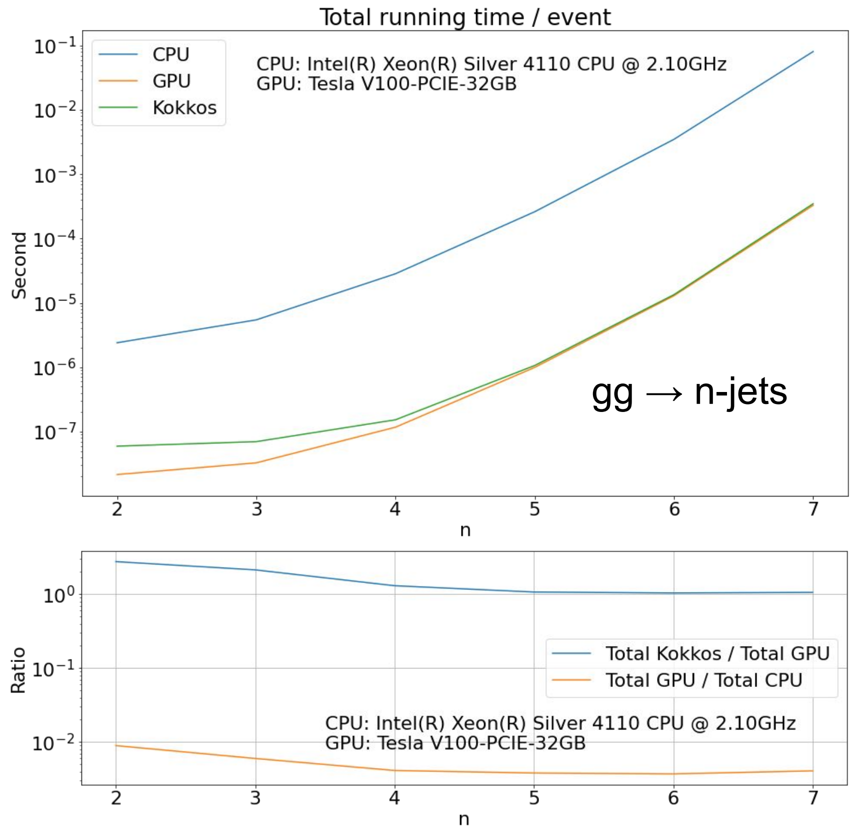
⇒ **Algorithmic choice: Sum colors**



Taylor Childers

Performance of Kokkos

- So, does Kokkos provide equivalent performance?
- Plot shows early versions of BlockGen calculating the process: gg→njets
- Time per Event on y-axis, number of outgoing partons on x-axis
- Compare CPU with C++, GPU with CUDA, and GPU with Kokkos
- Can see the CUDA is 100x faster than the CPU for this example
- Kokkos is slightly less performant than CUDA at low multiplicity (low computational complexity), but reaches comparable performance as multiplicity increases.



Precision Frontier

Johann Usovitsch

Precision test of the Standard Model | Future prospects

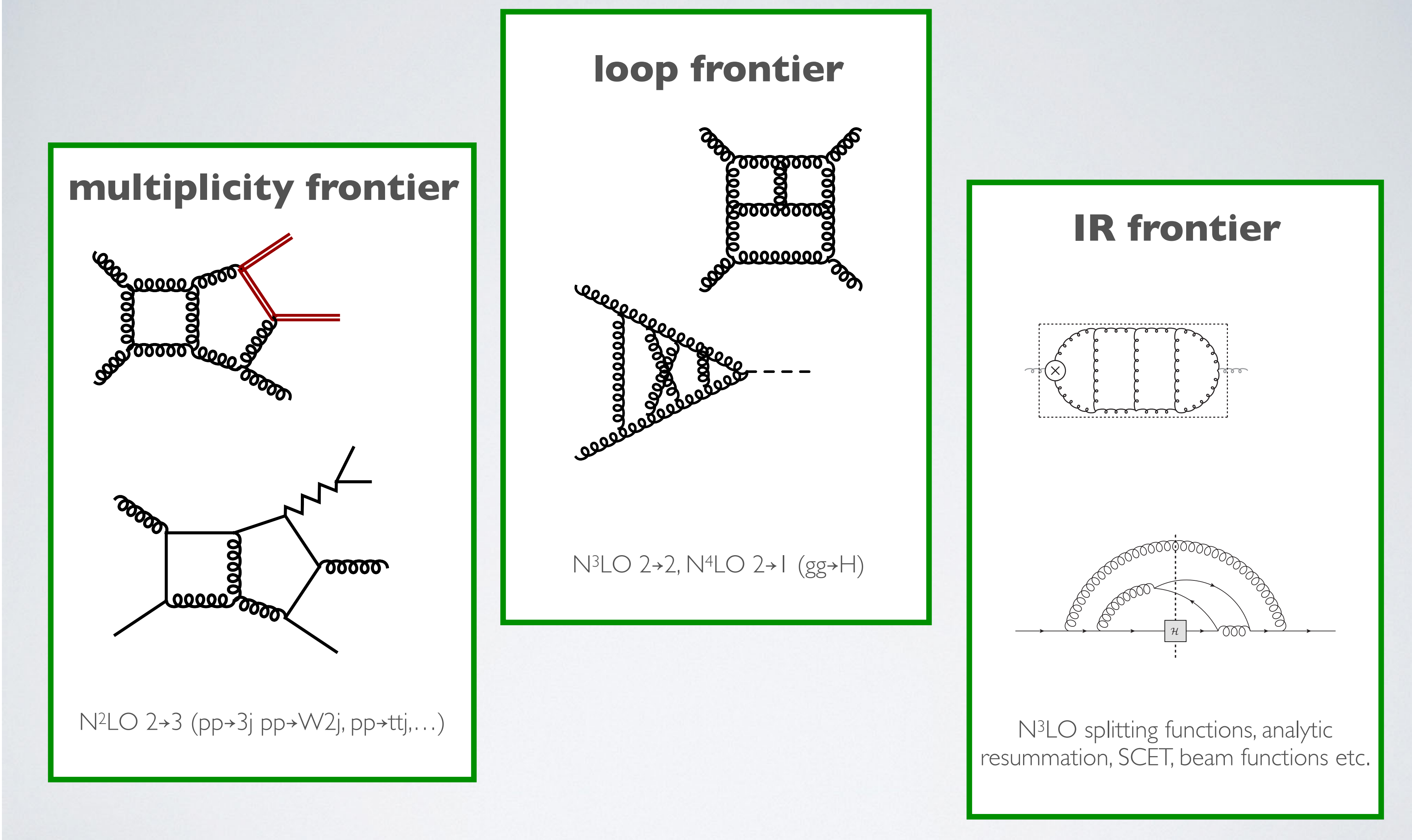
Overview of future experiments as of 2022

	Experiment uncertainty			Theory uncertainty
	ILC	CEPC	FCC-ee	Current
M_W[MeV]	3-4	3	$\cancel{1}$0.3	4
$\sin^2\theta^{l}_{\text{eff}}[10^{-5}]$	1	2.3	?0.6	4.5
Γ_Z[MeV]	0.8	0.5	$0\cancel{.}$10.025	0.4
$R_f[10^{-5}]$	14	17	$\cancel{6}$1	15

- Recent update from [Alain Blondel, Patrick Janot, Eur.Phys.J.Plus 137 (2022) 1]
- To match the precision of the experiment we compute 3-loop and 4-loop Standard Model predictions

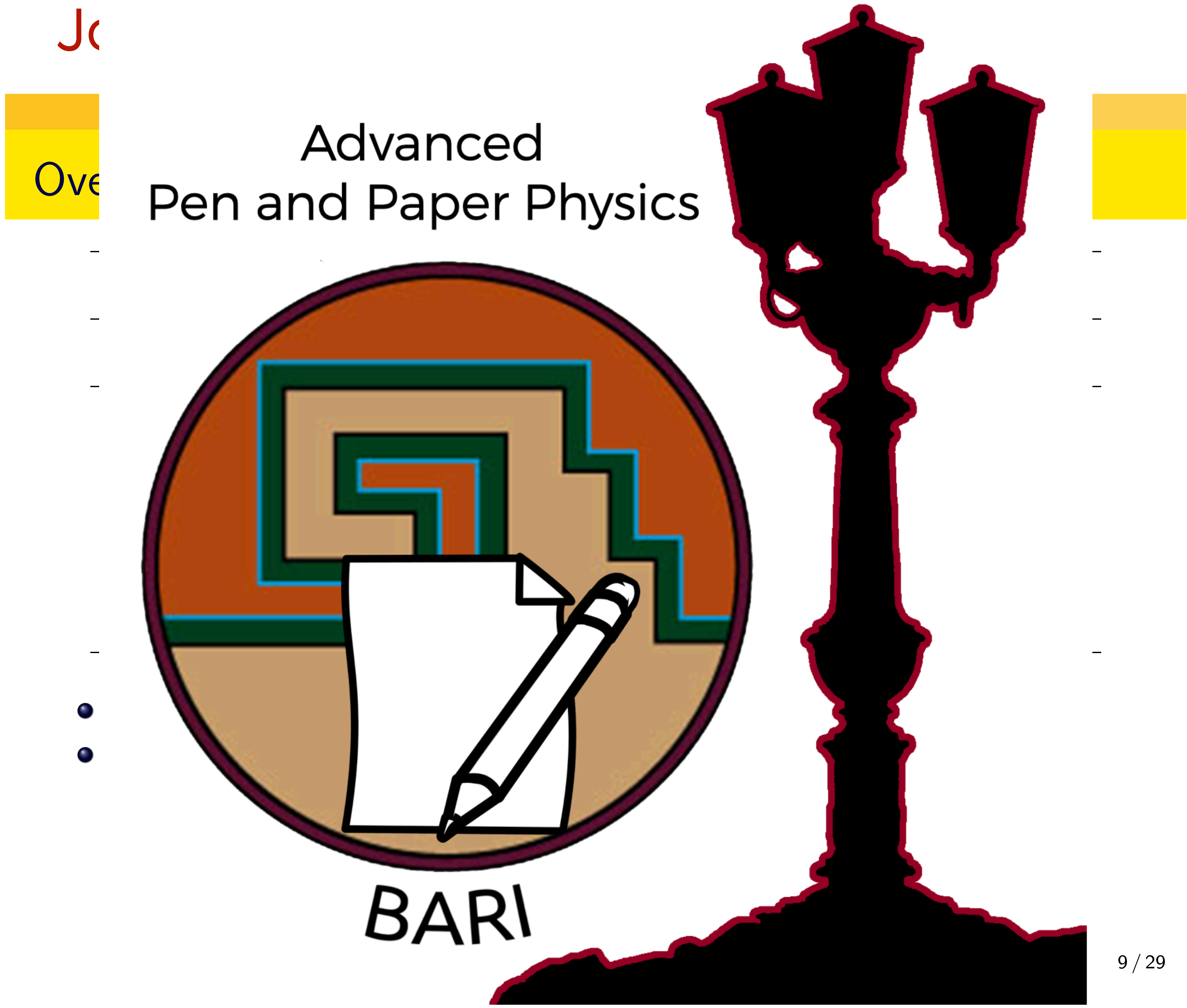
9 / 29

Simon Badger

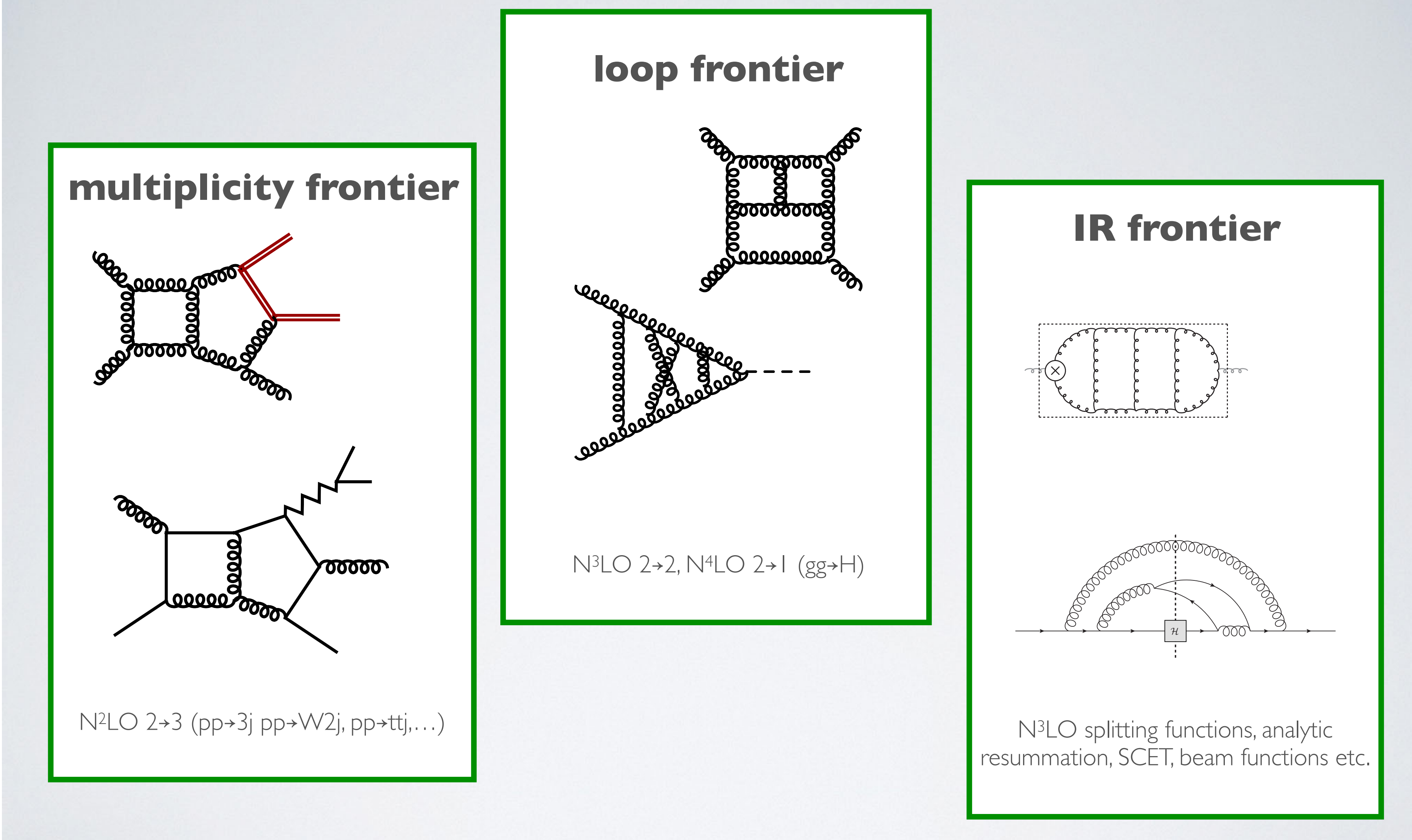


- Numerical methods
  - Avoiding algebraic complexity
- Physics informed
  - Exploiting known structures

Precision Frontier



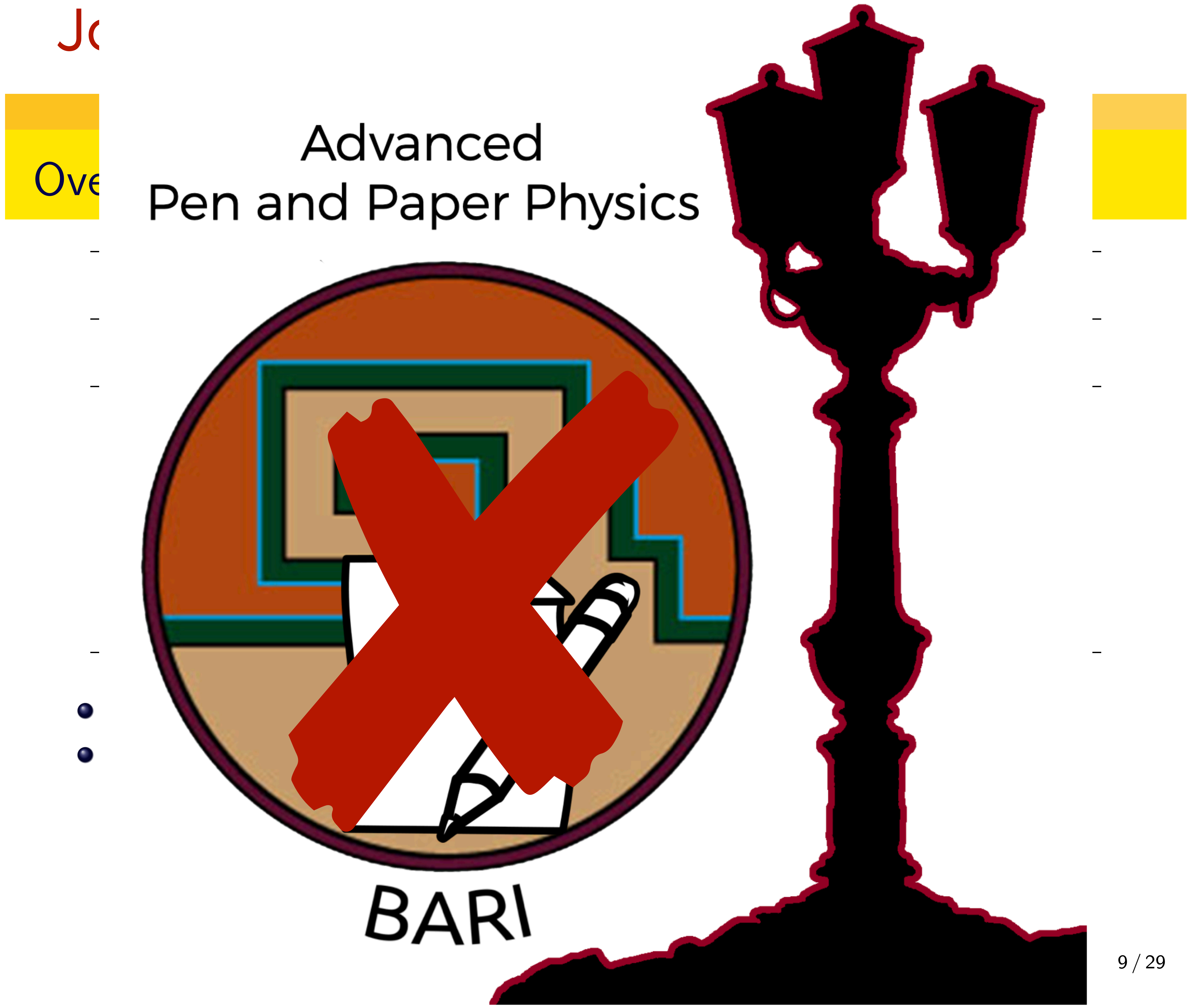
Simon Badger

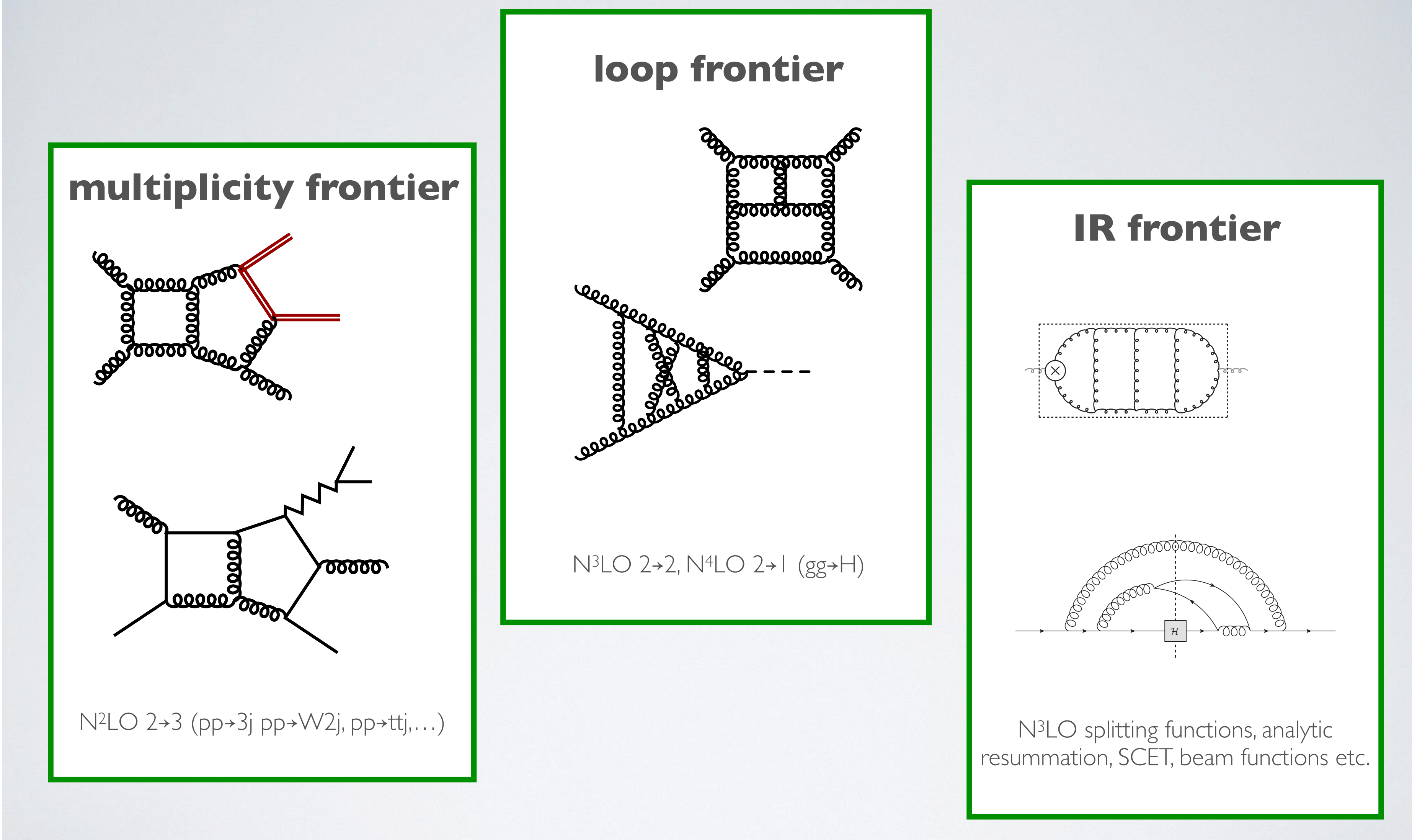


- Numerical methods
  - Avoiding algebraic complexity
- Physics informed
  - Exploiting known structures

Precision Frontier

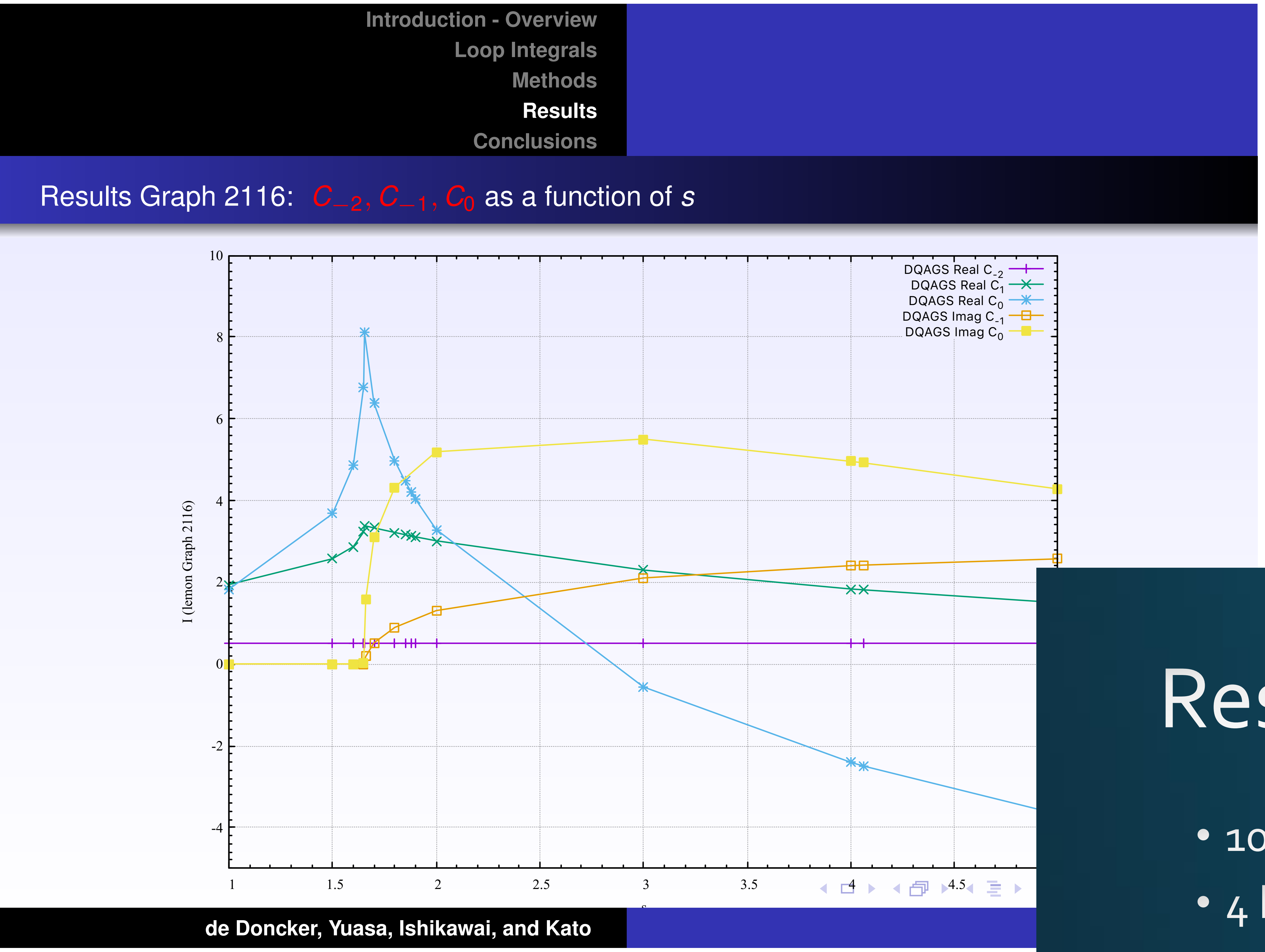
Simon Badger



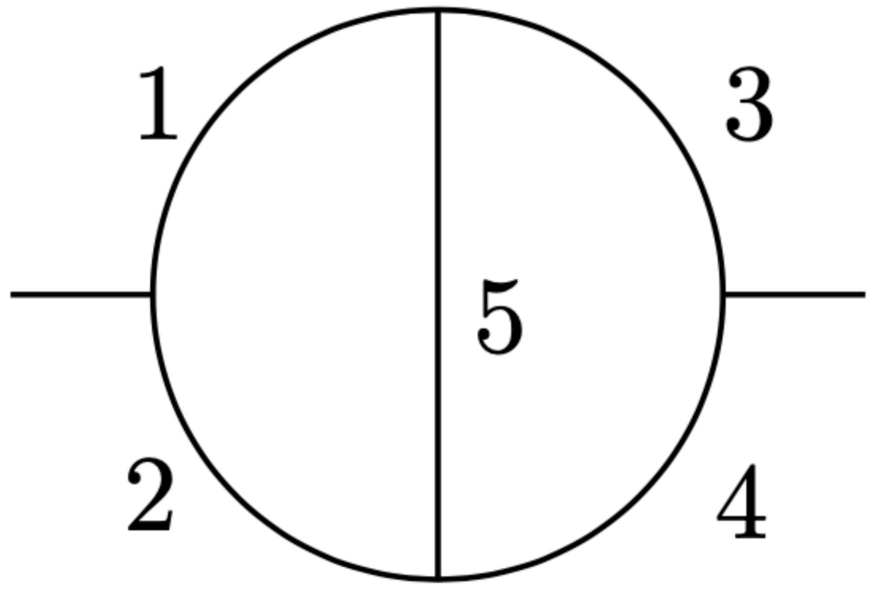


- Numerical methods
  - Avoiding algebraic complexity
- Physics informed
  - Exploiting known structures

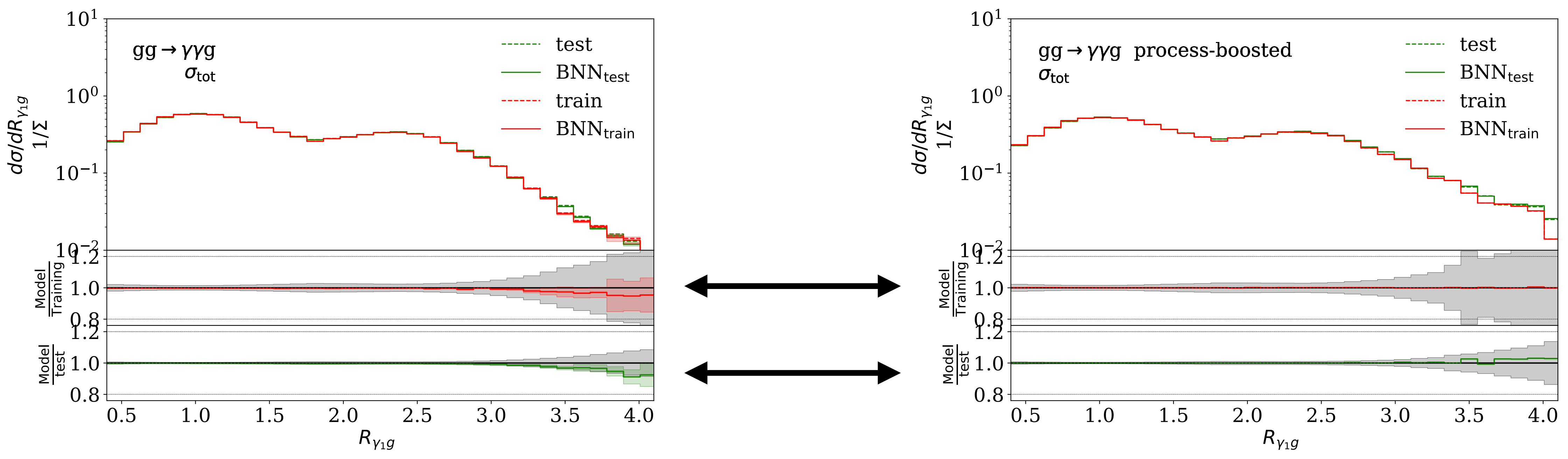
Numerical approaches



Elise de Doncker



Kinematic distributions



Uncertainties

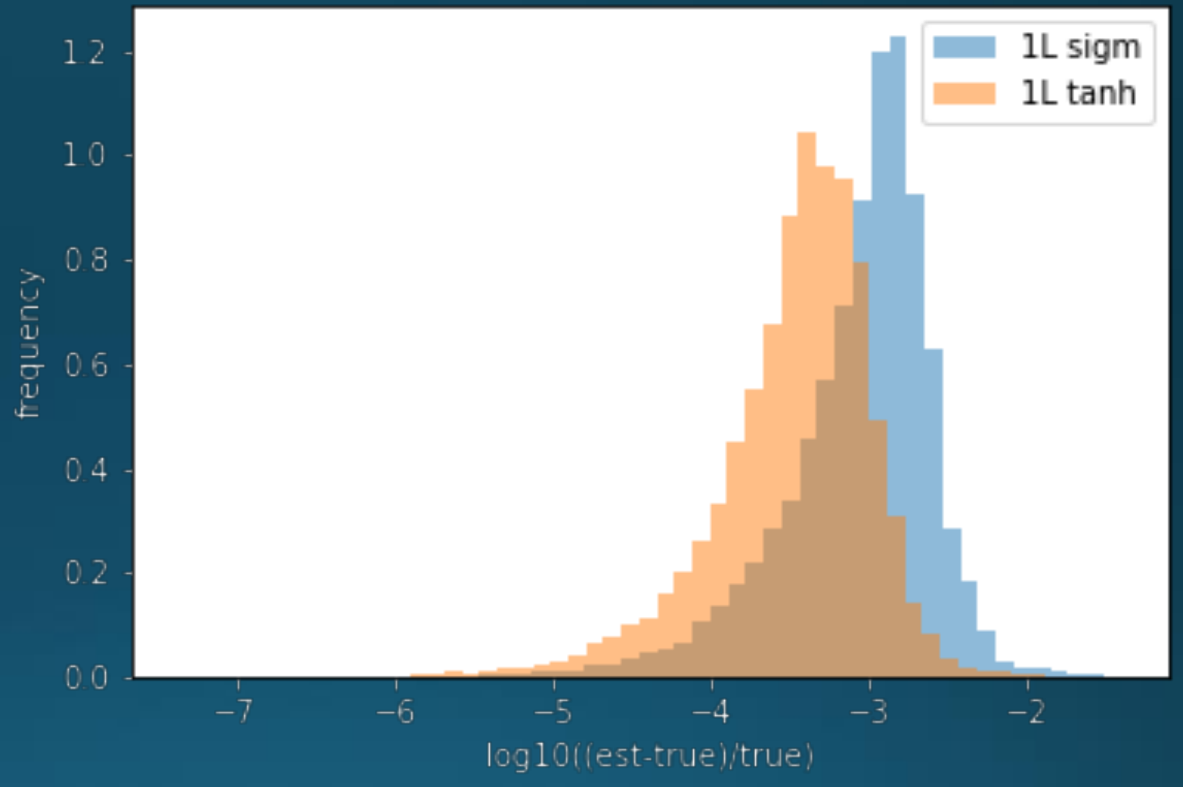
Gray shades indicate statistical limitation of training data.

Anja Butter

Result

Daniel Maitre

- 100 nodes
- 4 hidden layers
- 4M x 800 = 3.2B PS points



$$p = \log_{10}\left|\frac{e-t}{t}\right|$$

$$L = \mathrm{MSE}\left(f(s_1, \ldots, s_m; x_1, \ldots, x_k), \frac{d\mathcal{N}(s_1, \ldots, s_m; x_1, \ldots, x_k)}{dx_1 \ldots dx_k}\right)$$

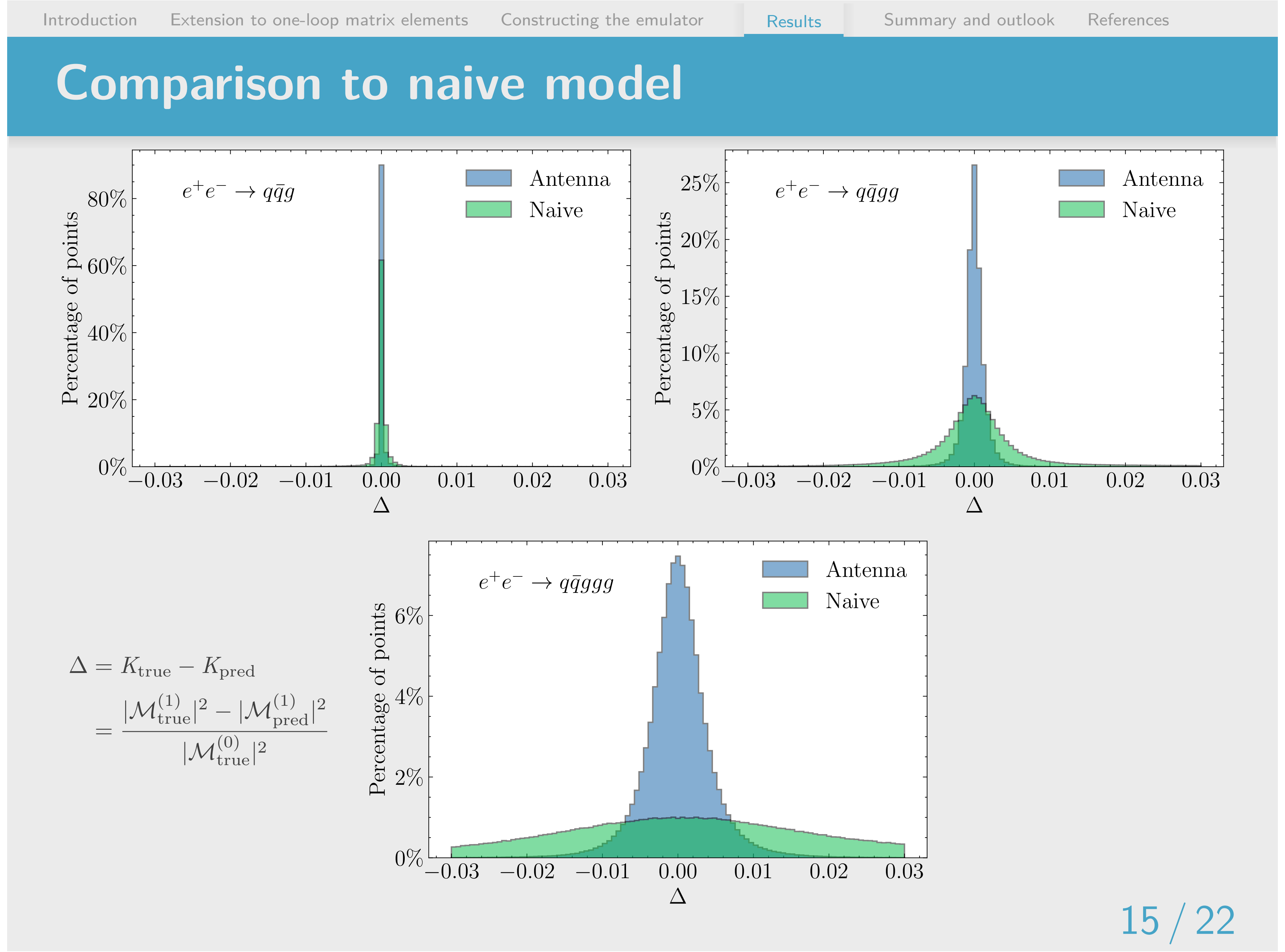
Machine learning the primitive

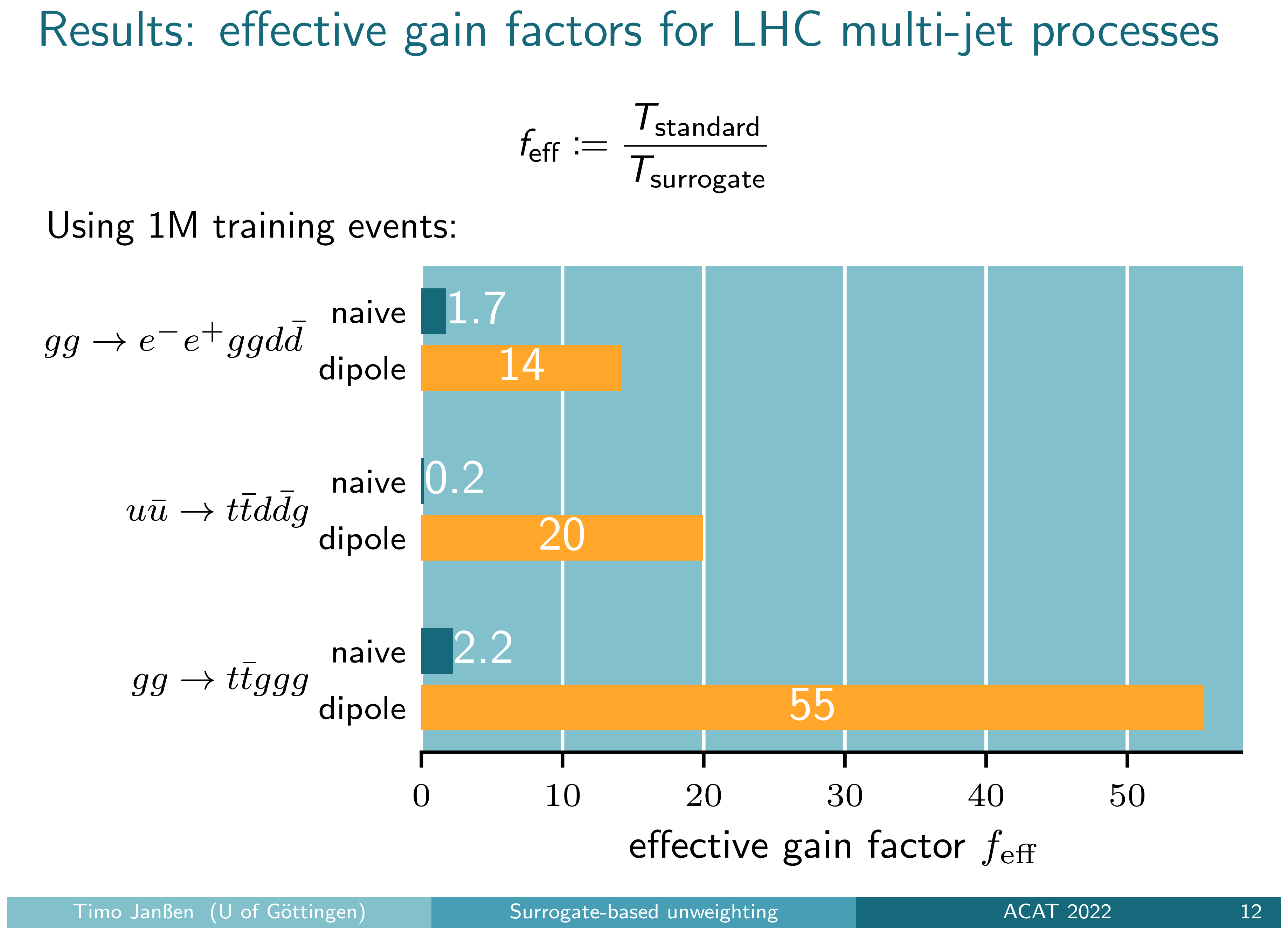
Numerical approaches - physics informed

$$K_{n+1} = C_0 + \sum_{\{ijk\}} C_{ijk} \frac{X^1_{ijk}}{X^0_{ijk}}$$

Learning K factors/matrix elements

- coefficients of antenna functions





Henry Truong

Timo Janssen

Analytic approaches

Ryan Moodie

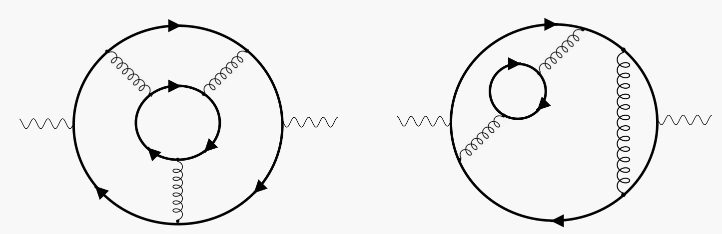
Timing

Channel	f64/f64		Evaluation strategy	
	Time (s)	f (%)	Time (s)	f (%)
$gg \to ggg$	1.39	69	1.89	77
$gg \to \bar{q}qg$	1.35	91	1.37	91
$qg \to qgg$	1.34	92	1.57	93
$q\bar{q} \to ggg$	1.34	93	1.38	93
$\bar{q}Q \to Q\bar{q}g$	1.14	99	1.16	99
$\bar{q}\bar{Q} \to \bar{q}\bar{Q}g$	1.36	99	1.39	99
$\bar{q}g \to \bar{q}Q\bar{Q}$	1.36	99	1.39	99
$\bar{q}q \to Q\bar{Q}g$	1.14	99	1.14	99
$\bar{q}g \to \bar{q}q\bar{q}$	1.84	99	1.90	99
$\bar{q}\bar{q} \to \bar{q}\bar{q}g$	1.82	99	1.94	99
$\bar{q}q \to q\bar{q}g$	1.71	99	1.77	99
$gg \to \gamma\gamma g$ *	9	99	26	99

Finite fields

Local unitarity

N_f part @N^{3}LO $e^+e^- \to jj$

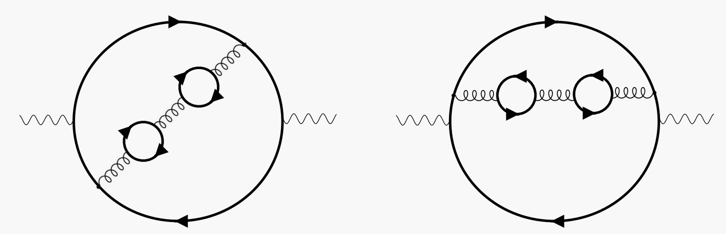


(+ many more)

$K_{\alpha_s^3}^{N_f,\mathrm{LU}} = -77.1(1.7)$

$K_{\alpha_s^3}^{N_f,\mathrm{bm}} = -76.8086$

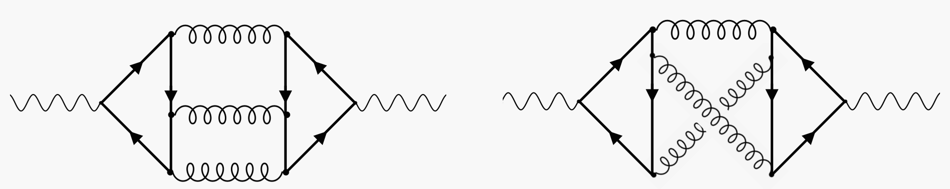
N_f^2 part @N^{3}LO $e^+e^- \to jj$



$K_{\alpha_s^3}^{N_f^2,\mathrm{LU}} = -0.35(24)$

$K_{\alpha_s^3}^{N_f^2,\mathrm{bm}} = -0.331415$

singlet part @N^{3}LO $e^+e^- \to jj$



$K_{\alpha_s^3}^{\mathrm{singlet},\mathrm{LU}} = -25.6(1.5)$

$K_{\alpha_s^3}^{\mathrm{singlet},\mathrm{bm}} = -26.4435$

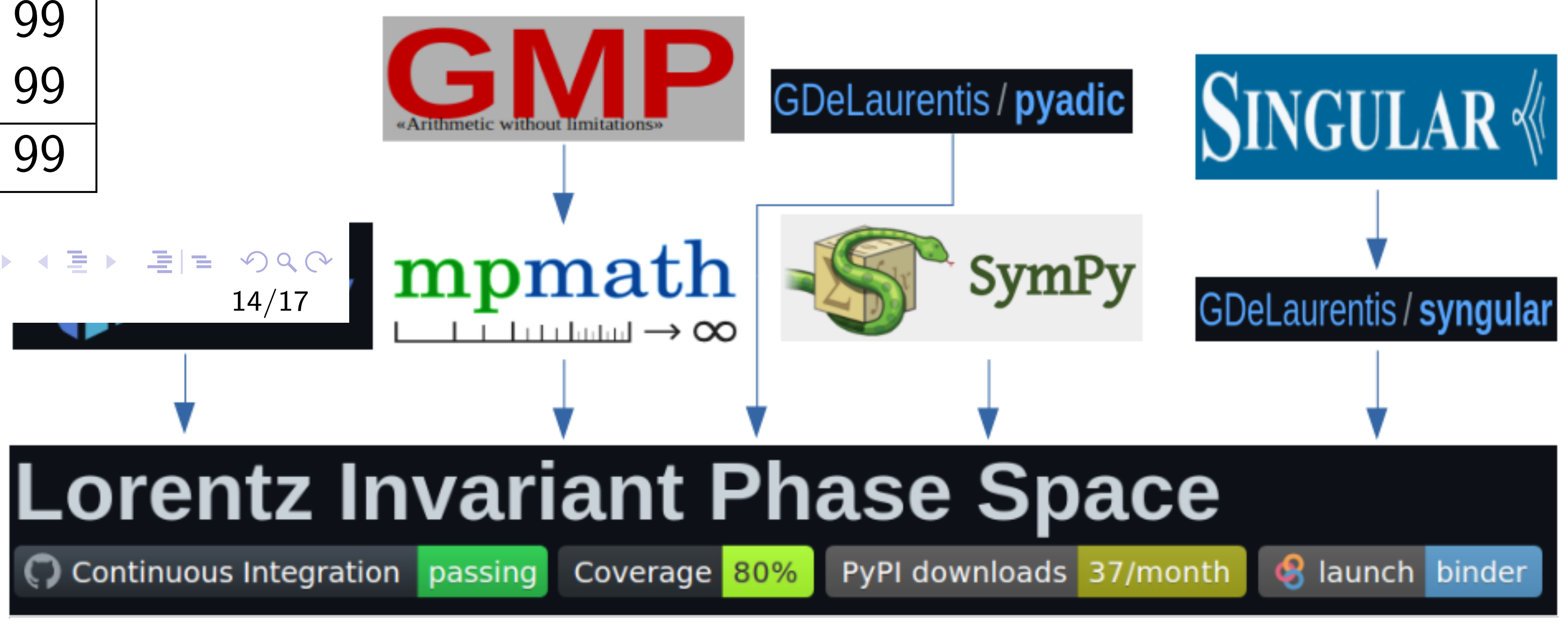
Benchmarks:
Herzog, Ruijl, Ueda, Vermaseren, Vogt : arXiv:1707.01044

Testing
- ✓ N3LO IR cancellations
- ✓ 3-loop UV renormalisation
- ✓ 1,2,3-loop self-energies

Zeno Capatti

Giuseppe de Laurentis

CS GRAPH[1234]



[1] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

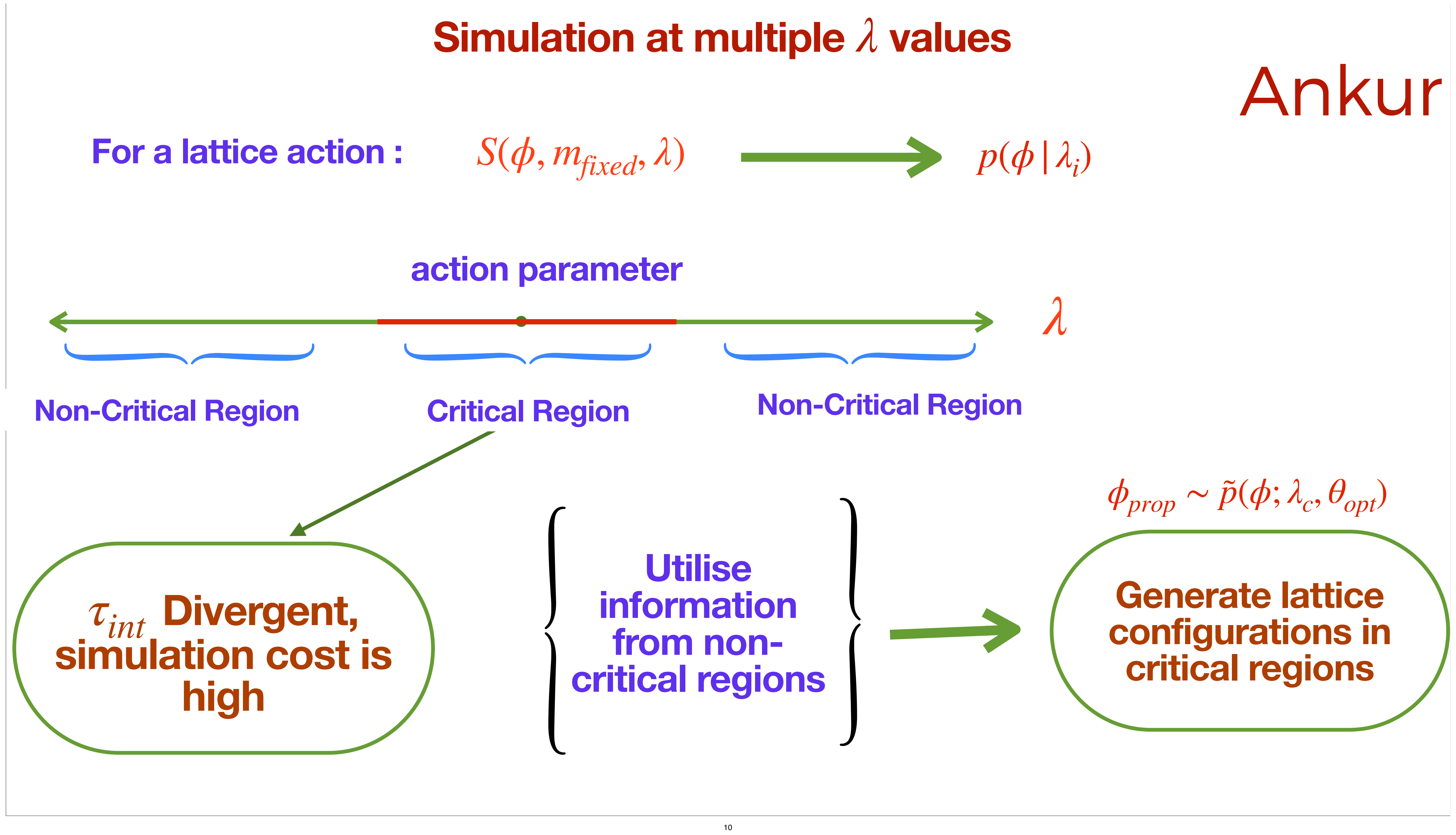
[2] Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18).* http://mpmath.org/. 2013.

[3] Aaron Meurer et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (Jan. 2017), e103. ISSN: 2376-5992.

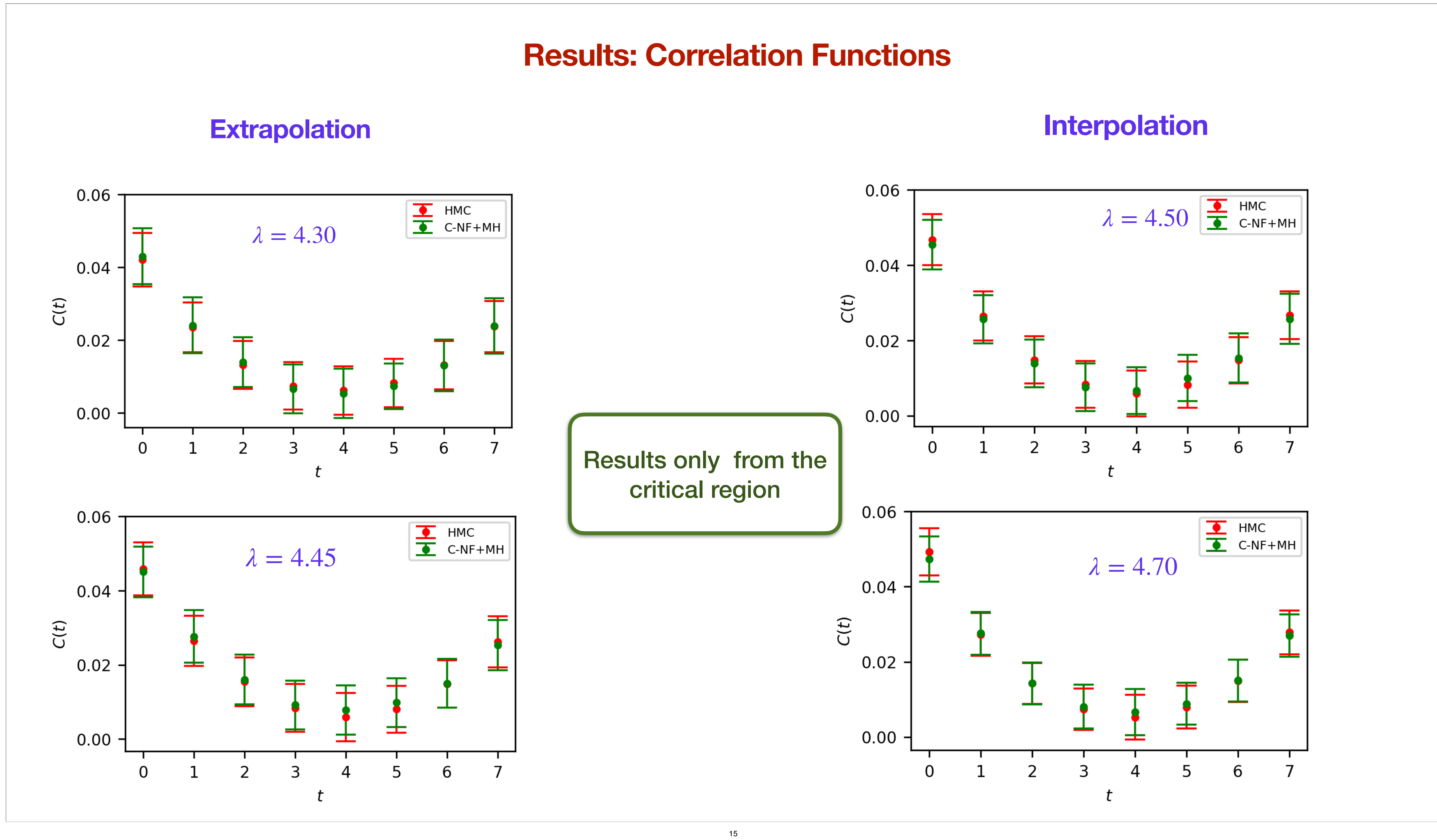
[4] Wolfram Decker et al. SINGULAR *4-3-0 — A computer algebra system for polynomial computations.* http://www.singular.uni-kl.de. 2022.

Anke Biekötter for Track 3

Non-perturbative physics



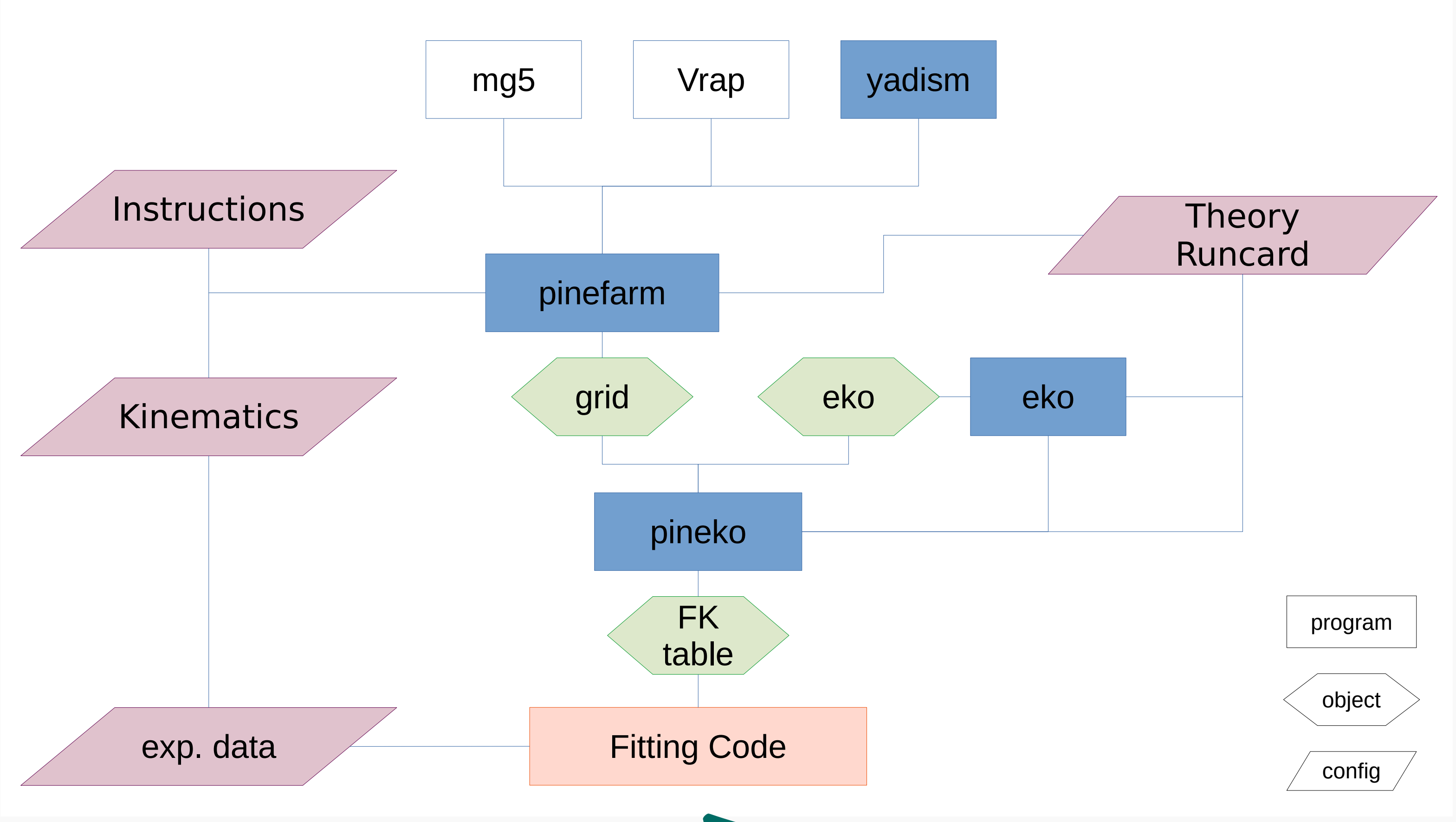
Ankur Singha



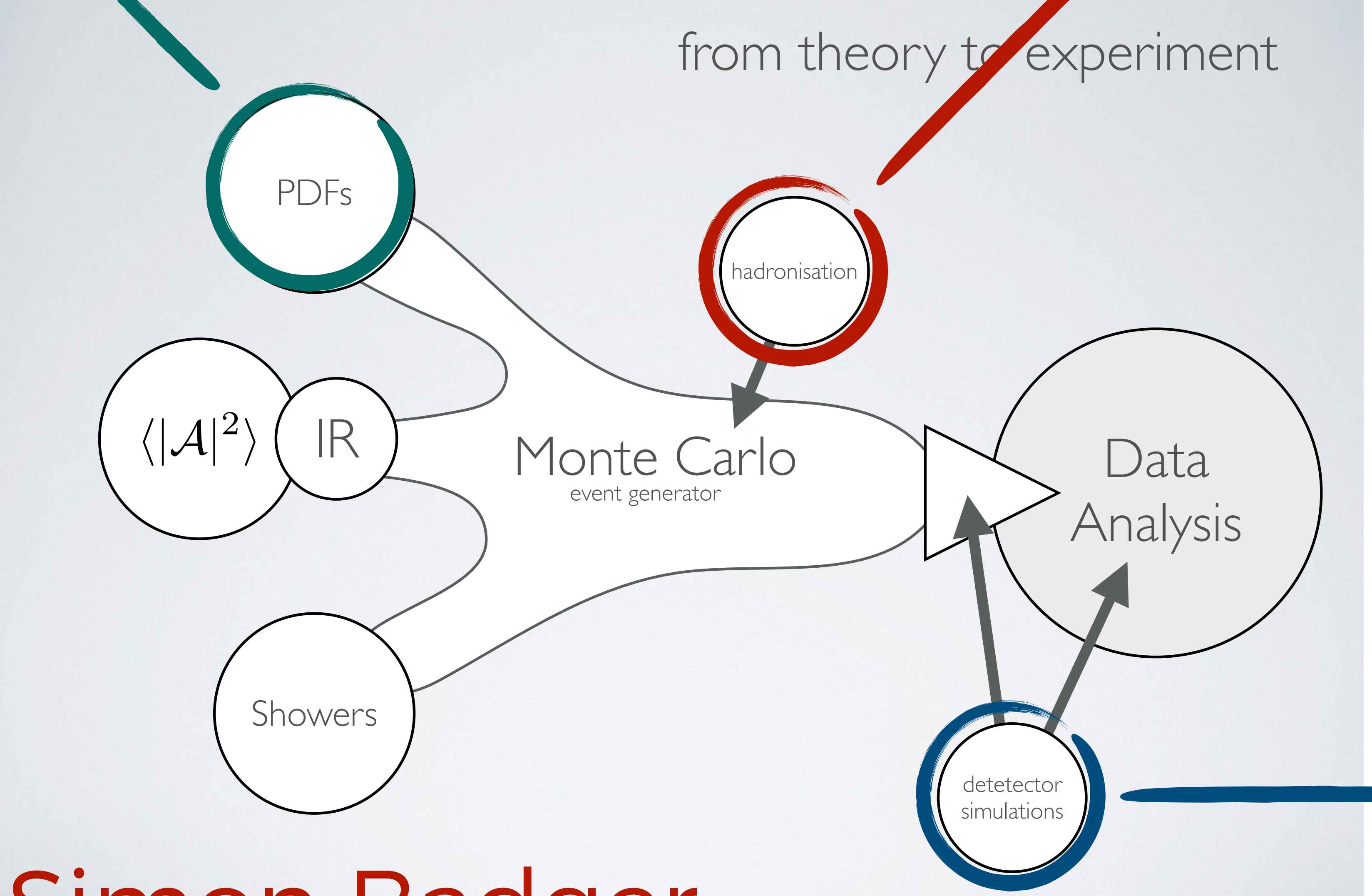
New Theory Prediction Pipeline

Produce FastKernel (FK) tables!

Felix Hekhorn



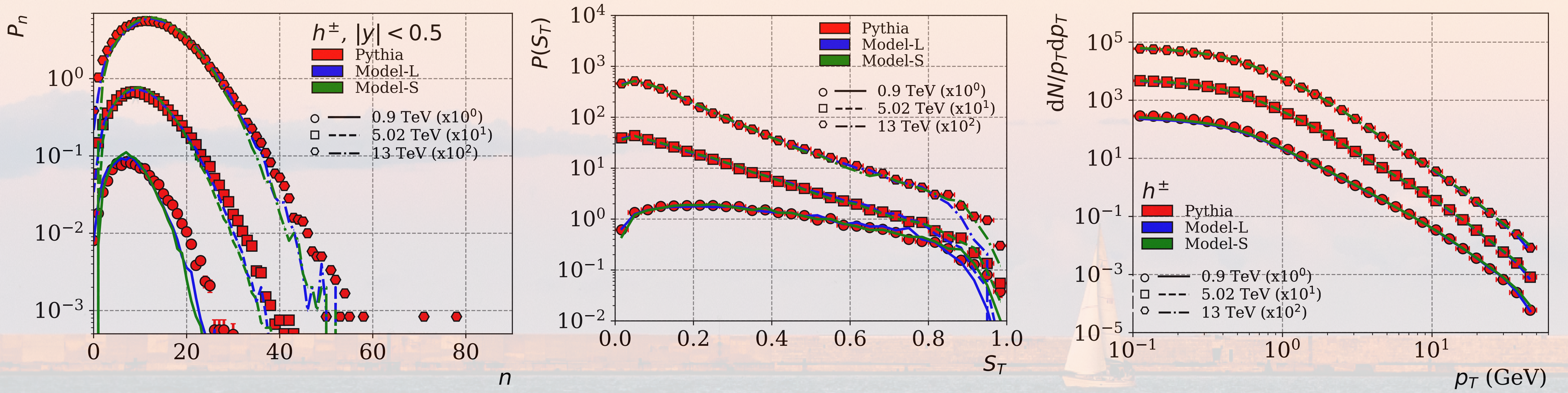
The workhorse in the background



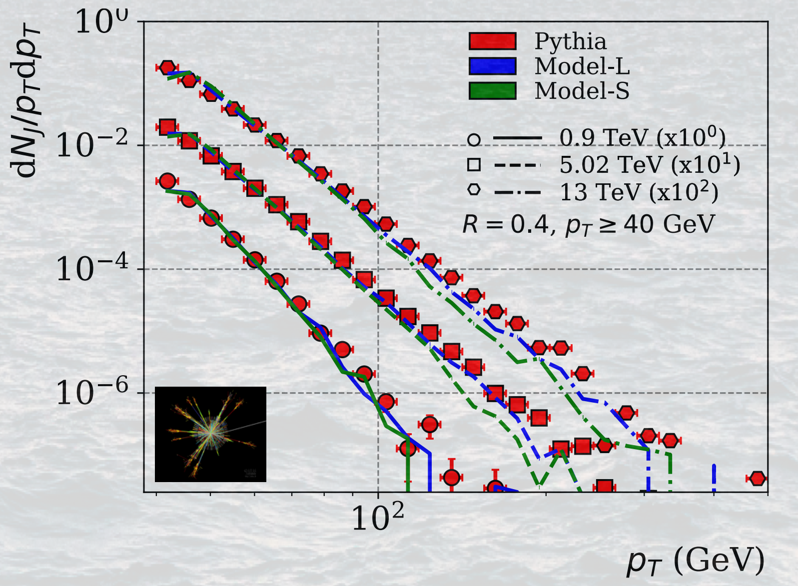
Simon Badger

Proton-proton @ 0.9-13 TeV, Predictions

Gabor Biro



- So far: everything at √s = 7 TeV → the **ONLY** energy, where the models were trained
  - Good agreement for all observable quantities as **predictions** for other LHC energies
- **Multiplicity scaling?**



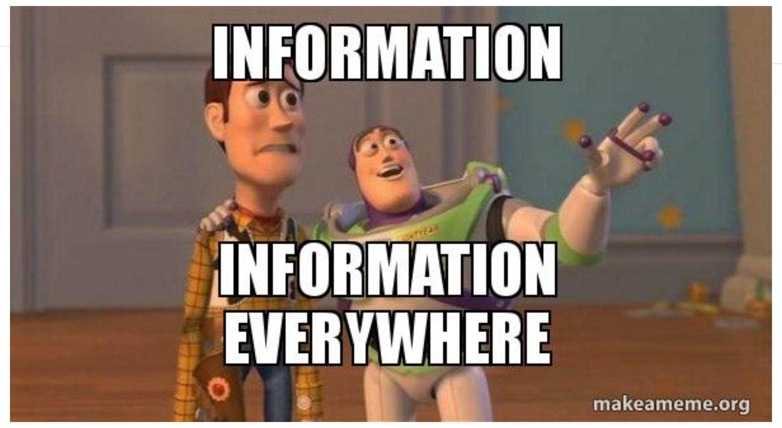
Preparation of detector configuration

MCrd should have data-driven detector configurations

Manipulate the data calibration constants to provide MC payloads → gather together detector related quantities from each sub-detector

Alignment constants

Dead channel mapping



Hit times

Trigger map

Detector geometry

PDFs of the energy depositions

Alberto Martini

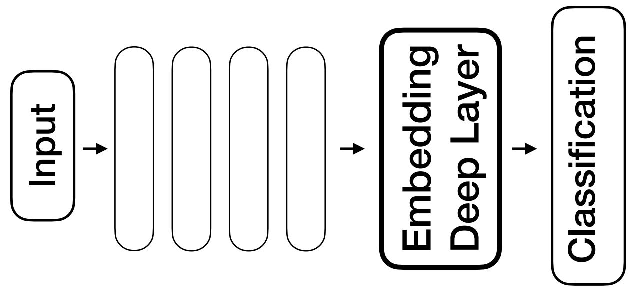
Beyond Standard Model

- Looking for new physics
  - LHC and beyond
- Model independence
- Recasting
- Symmetries

Sven Krippendorf

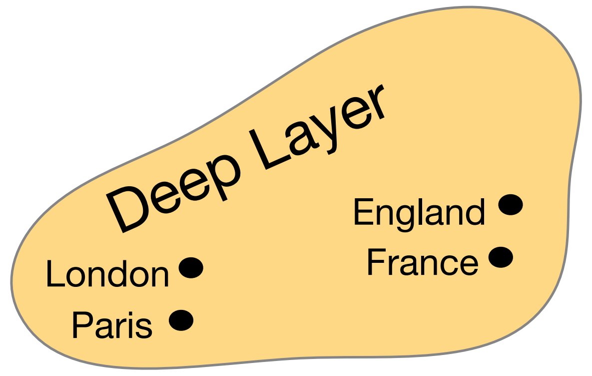
How to search for symmetries?

No direct optimisation available: embedding in deep layer



We need: group input with the same meaning together

Word2Vec does it:
(England - London = Paris - France)

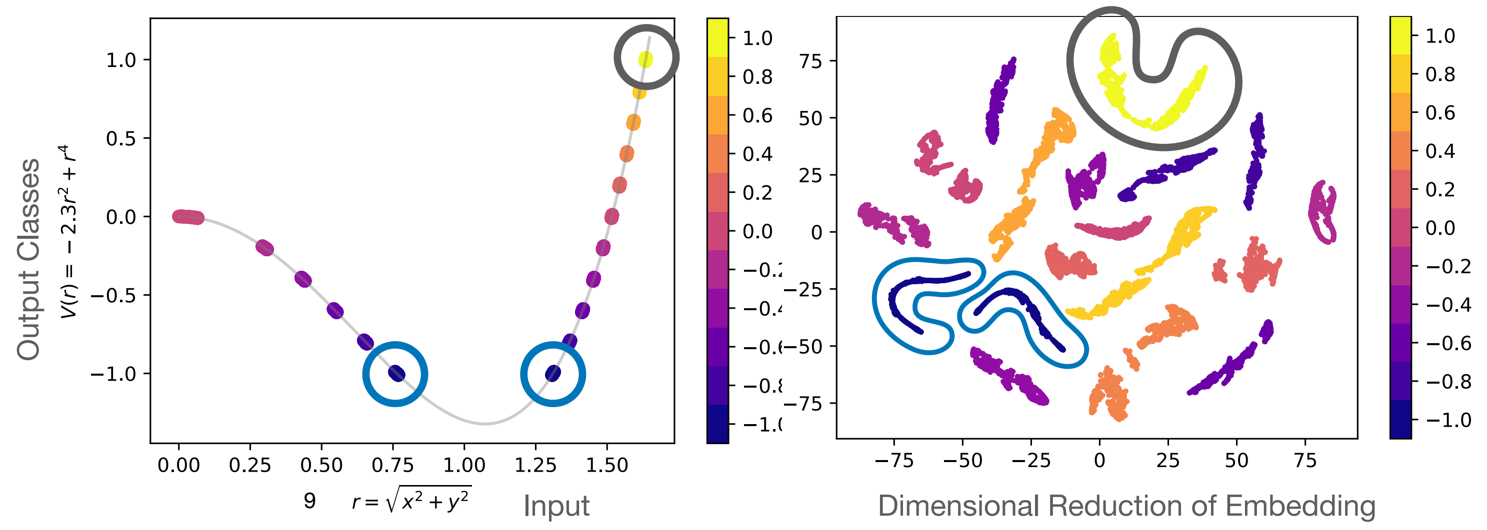


[1301.3781, used for re-discovering periodic table 1807.05617, classifying scents of molecules 1910.10685]

Can we search for symmetries in this way?

Yes!

Examples: SO(2), SU(2), discrete symmetries (CICY)



Krippendorf, Syvaeri 2020

Andrea Wulzer

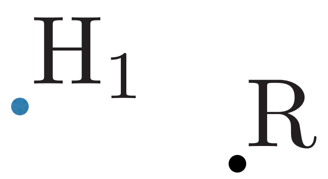
The LHC g.o.f. challenge

By analysing the LHC data, we would like to find evidence of **failure of the SM theory**, suggesting need of **BSM.**

This is a tremendously hard gof problem!

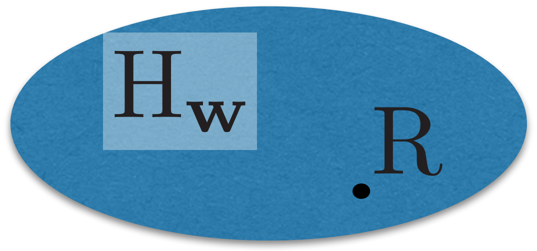
BSM is tiny departure from SM, or large in tiny prob. region
Affecting few (unknown) observables over ∞ many we can measure

Model-dependent BSM searches



- Optimise sensitivity to **one specific BSM model**
- Fail to discover other models. **What if the right theoretical model is not yet formulated?**

Model-independent searches



- Could reveal **truly unexpected** new physical laws.
- No hopes to find Optimal strategy. For a Good strategy, we need a **good choice of H_w.**

7

Track 3

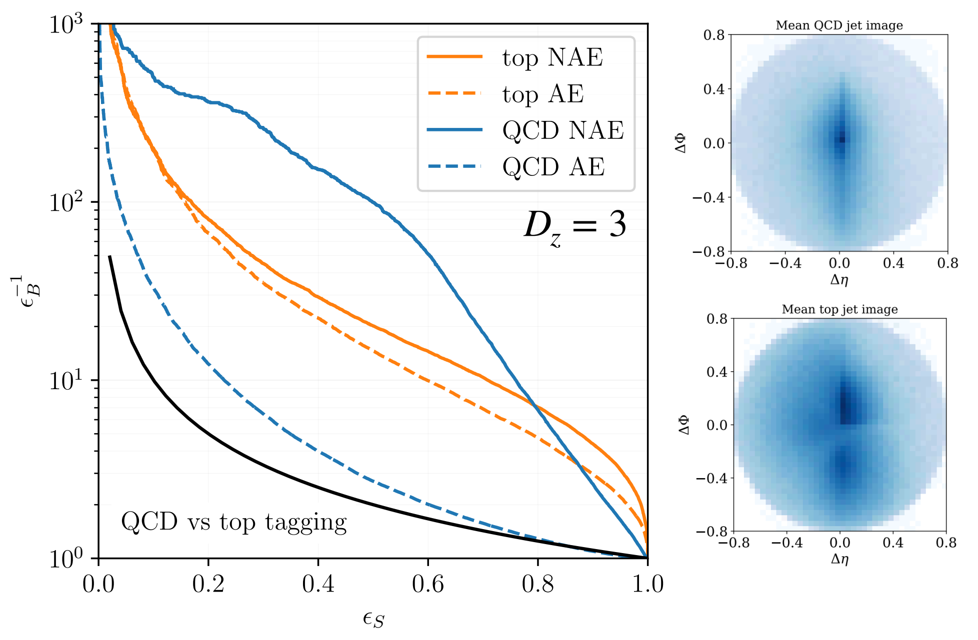
Beyond Standard Model

The Normalised AutoEncoder

[‘A normalised autoencoder for LHC triggers’ - Dillon, Favaro, Plehn, Sorrenson, Krämer]

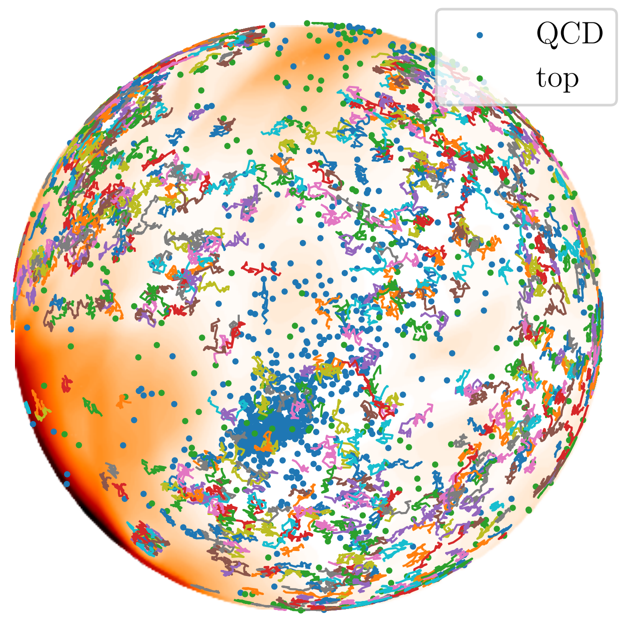
No complexity bias!

More rubust and reliable anomaly detection



Visualisation of the latent space

Looks like a mess, but very useful for interpreting the results and diagnosing problems with the training!



Barry Dillon

Anomaly detection

Theo Heimel

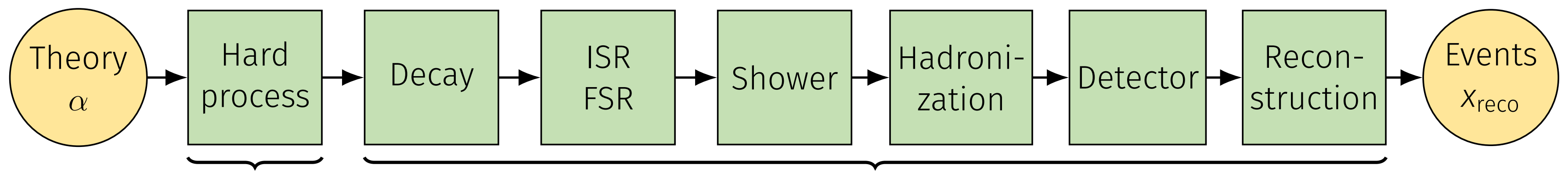
Matrix Element Method

4

- Process with theory parameter α, hard-scattering momenta x_{hard}
- Likelihood at hard-scattering level given by differential cross section

$$p(x_{\text{hard}}|\alpha) = \frac{1}{\sigma(\alpha)} \frac{d\sigma(\alpha)}{dx_{\text{hard}}}$$

- Neyman-Pearson lemma $\Longrightarrow$ optimal use of information
- Differential cross section only known analytically at hard-scattering level



likelihood known

likelihood intractable
$\Downarrow$
use neural network

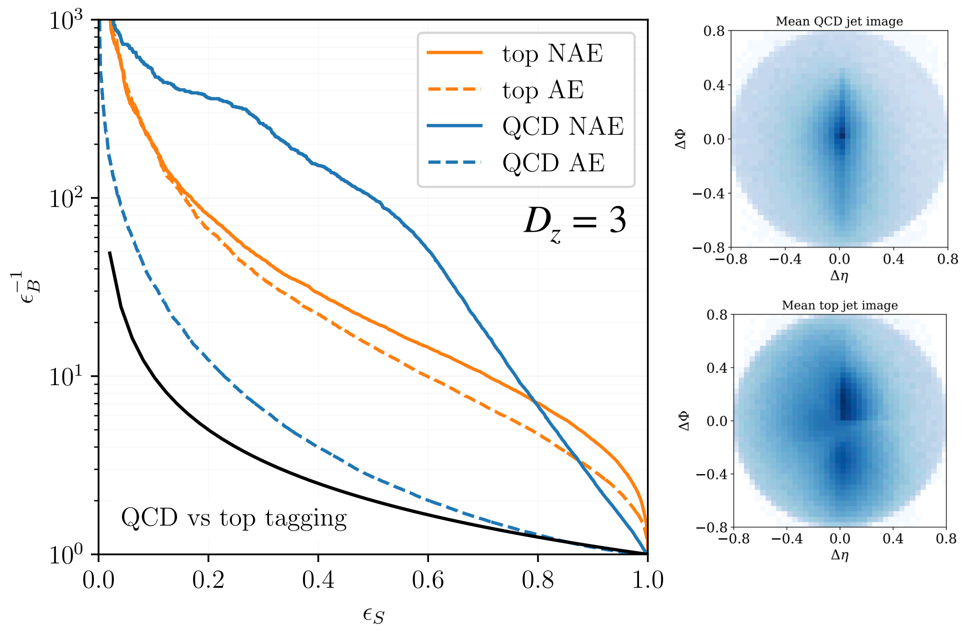
Beyond Standard Model

The Normalised AutoEncoder

[‘A normalised autoencoder for LHC triggers’ - Dillon, Favaro, Plehn, Sorrenson, Krämer]

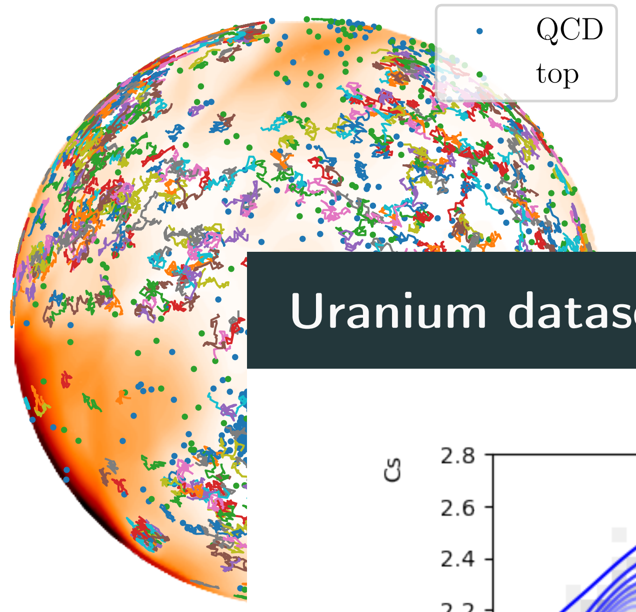
No complexity bias!

More rubust and reliable anomaly detection



Visualisation of the latent space

Looks like a mess, but very useful for interpreting the results and diagnosing problems with the training!



Barry Dillon

Barry Dillon — Universität Heidelberg — Anomaly searches for new physics at the LHC

Anomaly detection

Uranium dataset [Pasquale et al., 2022 in preparation]

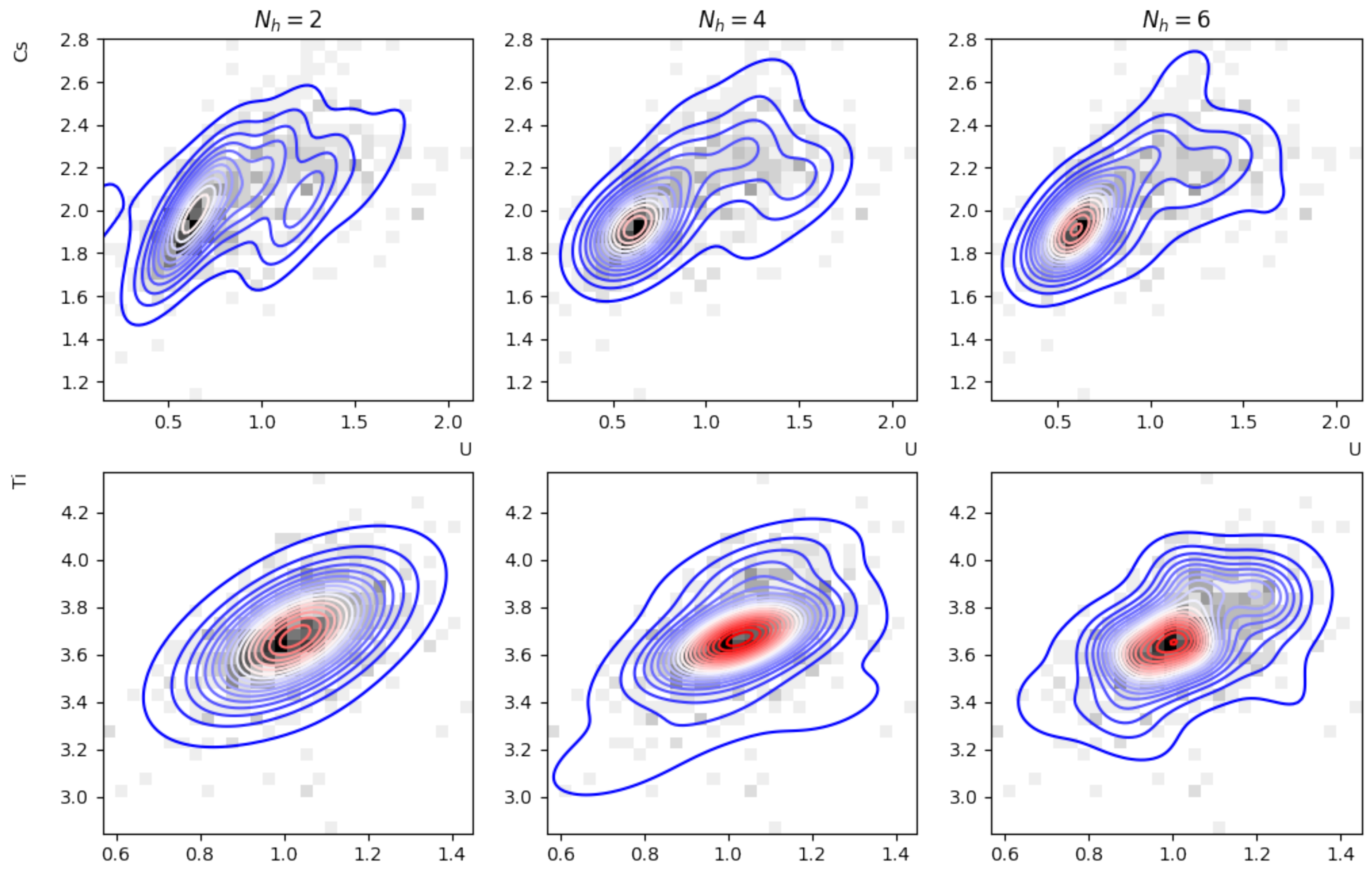


Figure 3: rRTBMs modelling the concentrations of Uranium and Cesium (first row), Cobalt and Titanium (second row) and, Cesium and Scandium (third row) for $N_h = 2, 4, 6$ (left,center,right). The rRTBM contours and histograms of the original data are shown.

15

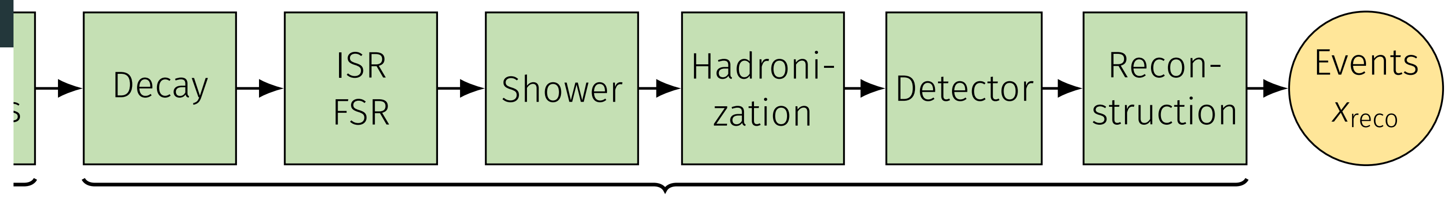
Theo Heimel

Matrix Element Method 4

- Process with theory parameter α, hard-scattering momenta x_{hard}
- Likelihood at hard-scattering level given by differential cross section

$$p(x_{\text{hard}}|\alpha) = \frac{1}{\sigma(\alpha)} \frac{d\sigma(\alpha)}{dx_{\text{hard}}}$$

- Neyman-Pearson lemma $\Longrightarrow$ optimal use of information
- cross section only known analytically at hard-scattering level



likelihood intractable

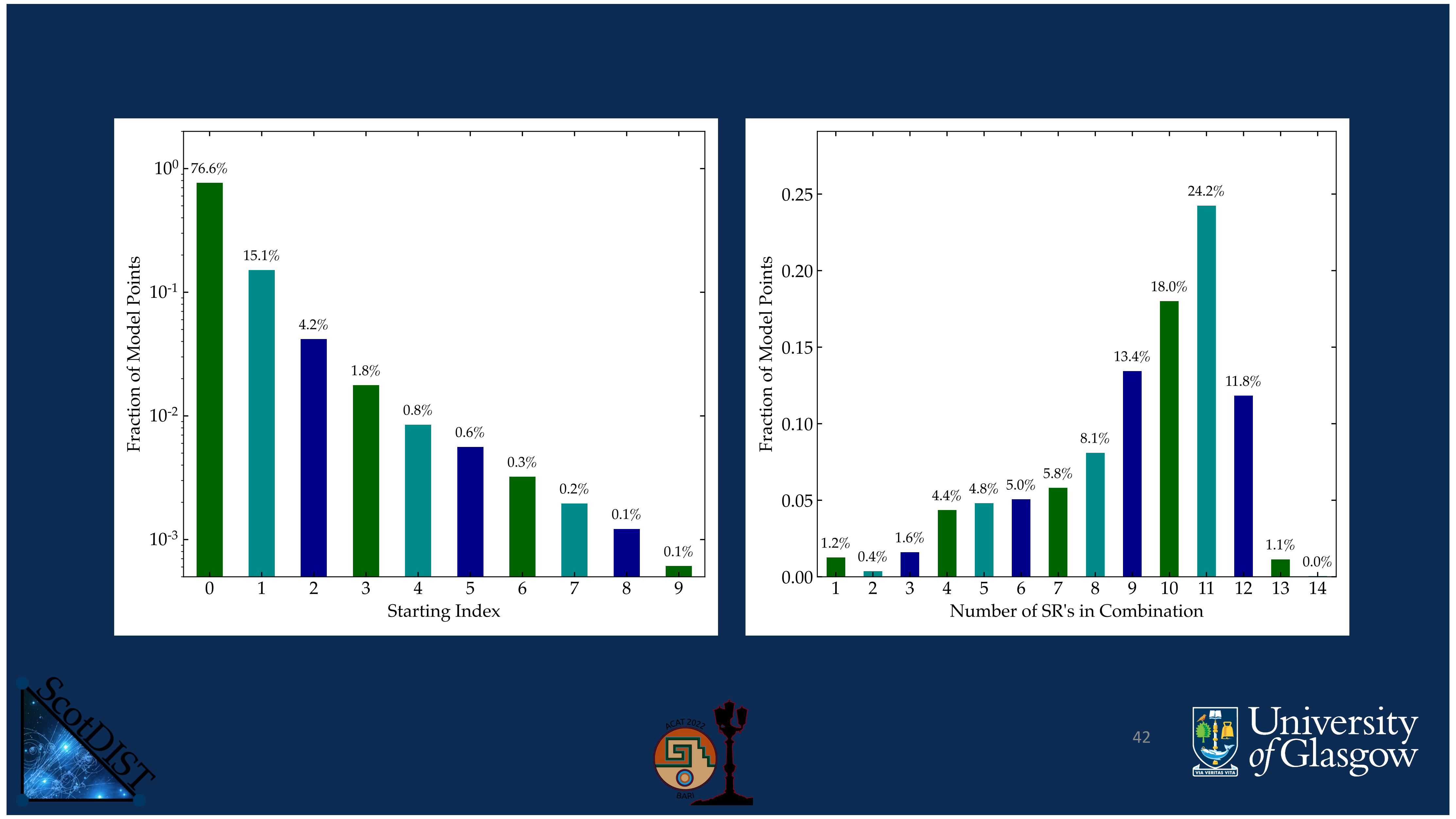
$\Downarrow$

use neural network

Andrea Pasquale

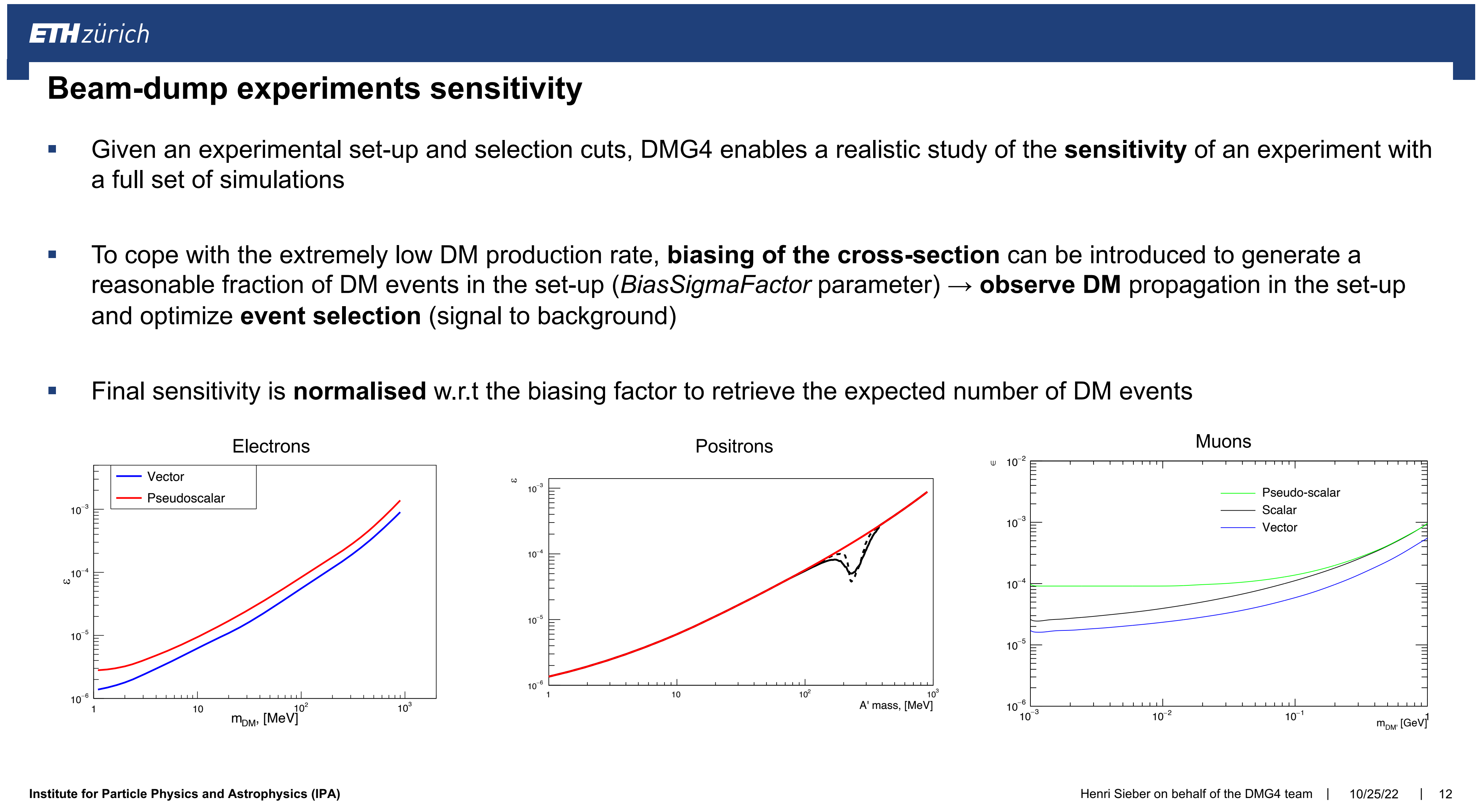
Beyond Standard Model

Recasting



Jamie Yellen

Dark Matter MC event generation



Henri Sieber

Useful Near-Term Quantum Algorithms

Jungsang Kim

- **Quantum Machine Learning (QML)**: Quantum advantages proven for

  Learning complex patterns w/ quantum feature maps (arXiv:2010.02174)

  Exponential gain in predicting certain worst-case error (arXiv:2101.02464)

  Quantum correlations used in generative modeling (arXiv:2101.08354)

- **Quantum Chemistry** and **Materials Studies**

  Variational quantum eigensolvers (VQE) for energy estimation

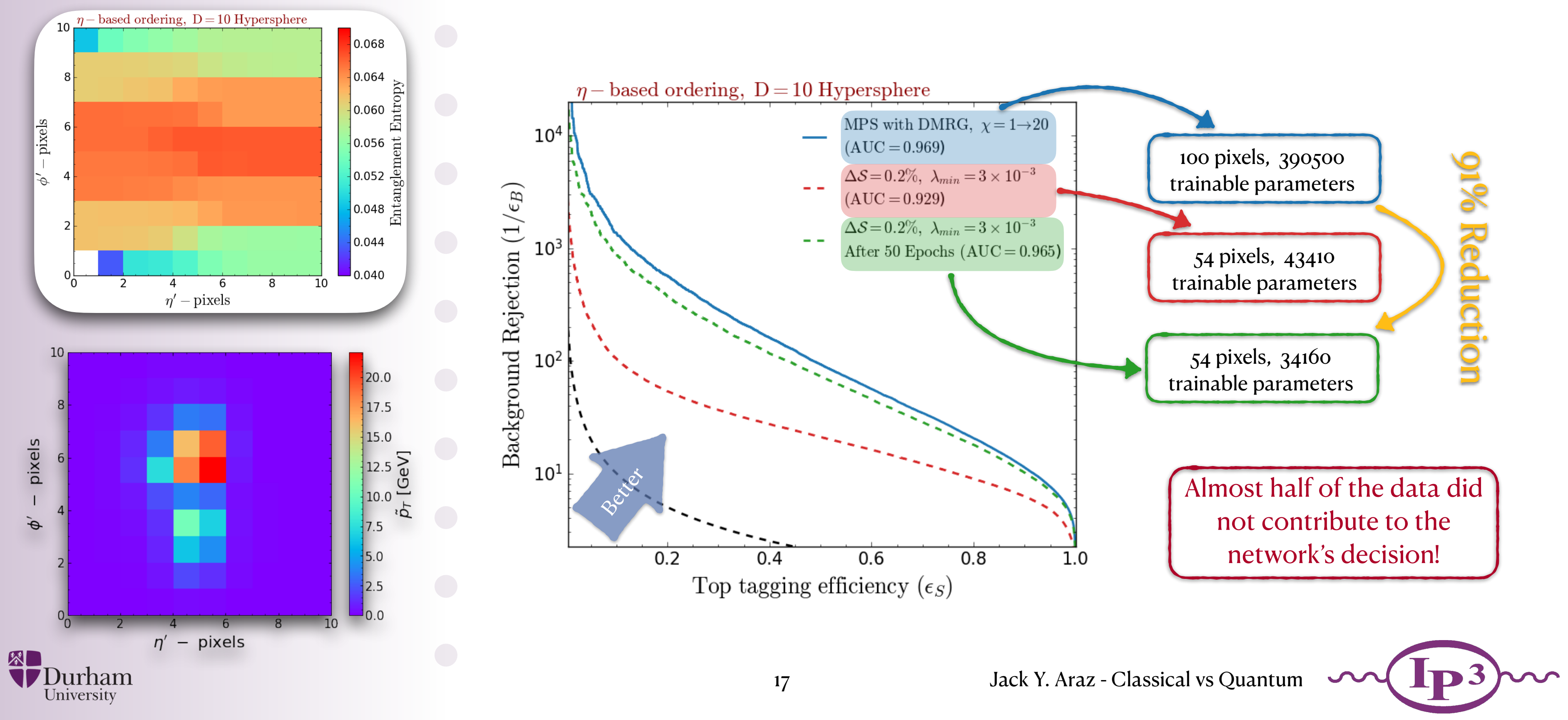
  Quantum simulation of dynamics of excitation

  Study of quantum many-body phenomena

- **Optimization Problems**: Quantum Approximate Optimization Algorithm

Quantum Machine Learning

Top Tagging through MPS



Durham University

17

Jack Y. Araz - Classical vs Quantum

IP3

Jack Araz

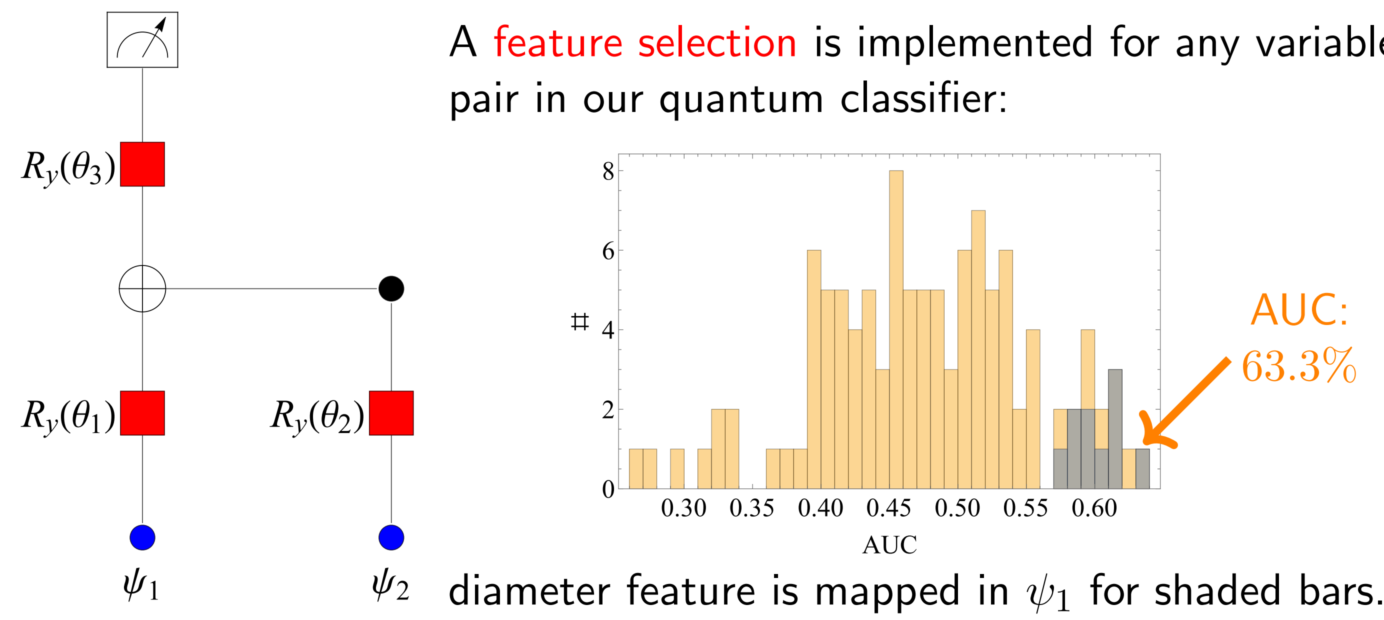
A hard decision making process

Our best classical (not quantum!) classifier with diameter, grading, histologic type, multifocality, in situ component, PgR:

AUC (%)	Accuracy (%)	Specificity (%)	Sensitivity (%)
70.8 (70.3-71.1)	69.8 (69.3-70.2)	74.8 (72.9-75.1)	61.0 (60.3-61.7)

reporting the 1^{st}-3^{rd} interquartile range after 10 ten-fold cross-validations.

A feature selection is implemented for any variables pair in our quantum classifier:



diameter feature is mapped in ψ_1 for shaded bars.

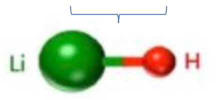
Domenico Pomarico

Application to Force Fields (Chemistry)

Kiss, Tacchino, *et al.*, *Mach. Learn.: Sci. Technol.* **3** 035004 (2022)

bond length r

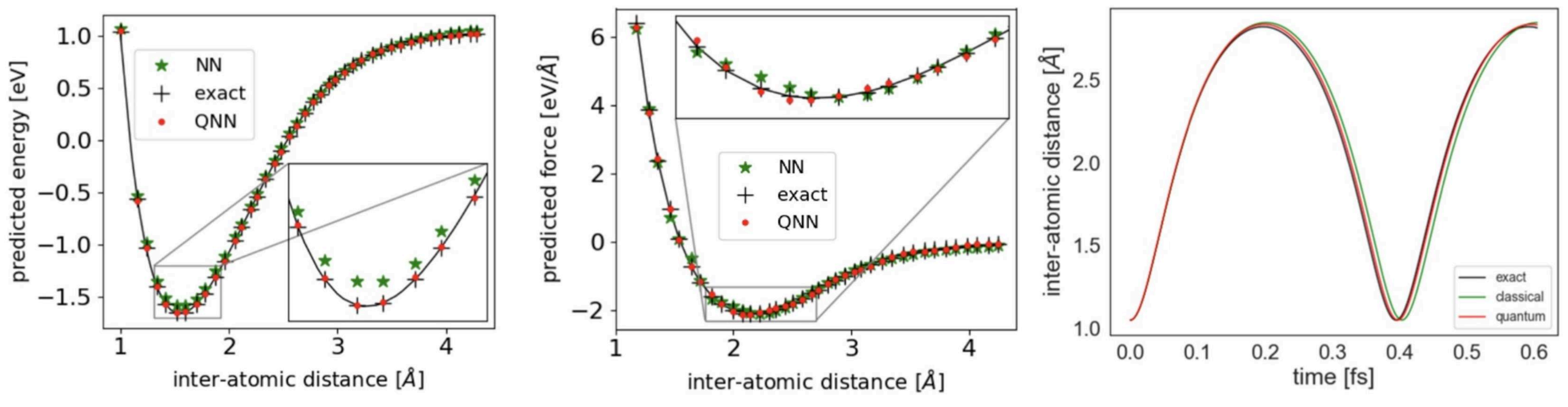
Li-H



$W : [-1,1] \to [-\pi,\pi]^3$

$r \mapsto \begin{pmatrix} \pi r \\ \arcsin(r) \\ \arccos(r) \end{pmatrix}$

QNN with 10 layers (50 parameters)



(a) LiH Energy (b) LiH Force (a) Time evolution of the inter-atomic distance

Comparison with a neural network of equal complexity

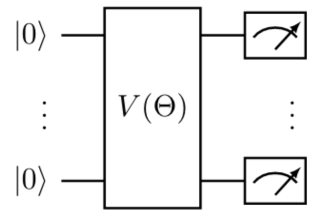
$\vec{F} = -\vec{\nabla}_r E(r)$

Molecular Dynamics

Oriel Kiss

Quantum Circuit Born machine (QCBM)

1. ***Sample*** *from a variational pure state* $|\psi(\theta)\rangle$ *by projective measurement with probability given by the* ***Born rule***: $p_\theta(x) = |\langle x|\psi(\theta)\rangle|^2$.



n dimensional binary strings map to 2^n bins of the discretized dataset.

2. ***Training*** (Hybrid loop):
   - KL divergence — Delgado and Hamilton, arXiv:2203.03578 (2022).
   - Adversarial (QGAN) — Zoufal, et al., *npj Quantum Inf* **5**, 103 (2019).
   - In the phase space. — Kyriienko, et al., arXiv: 2202.08253 (2022).
   - Maximum Mean Discrepancy

$$\mathrm{MMD(P,Q)} = \mathbb{E}_{\substack{X\sim P\\Y\sim P}}[K(X,Y)] + \mathbb{E}_{\substack{X\sim Q\\Y\sim Q}}[K(X,Y)] - 2\mathbb{E}_{\substack{X\sim P\\Y\sim Q}}[K(X,Y)]$$

3. Why the ***Maximum Mean Discrepancy*** MMD ?
   - Resource efficient for NISQ devices.
   - Stable.
   - However, empirically less performant than adversarial.

Michele Grossi

Quantum chemistry and fluids

Variational quantum eigensolver

Hardware results (IBMQ)

- Start from the classical solution (warm start).
- Qubit Based Excitation descending UCC Ansatz.
- 10 runs on the 27 qubits IBMQ mumbai chip.

Error mitigation

- **Readout:** individually inverse the error matrices

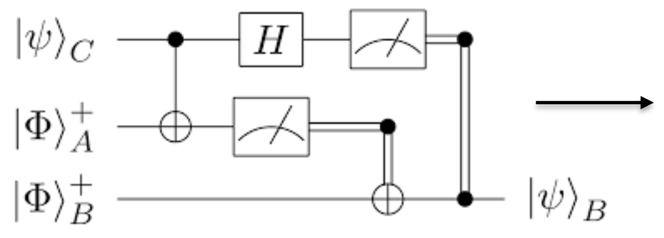
$$S_k = \begin{pmatrix} P^{(k)}_{0,0} & P^{(k)}_{0,1} \\ P^{(k)}_{1,0} & P^{(k)}_{1,1} \end{pmatrix}.$$

- Here, $P^{(k)}_{i,j}$ is the probability of the k-th qubit to be in state $j \in \{0,1\}$ while measured in state $i \in \{0,1\}$.

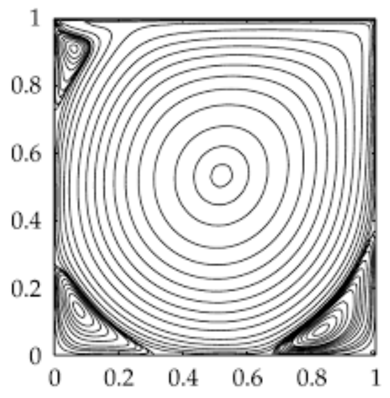
hardware	No. parameters	No. CNOT	mean	st. deviation	exact	error ratio
ibmq_mumbai raw (g.s.)	9	209	-6.27	0.269	-5.529	13.36%
ibmq_mumbai mitigated (g.s.)	9	209	-5.319	0.24	-5.529	3.81%
ibmq_mumbai raw (1st es)	3	41	-2.907	0.87	-3.420	14.97%
ibmq_mumbai mitigated (1st es)	3	41	-3.424	0.08	-3.420	0.12%

NEXT STEPS

- Build Quantum Circuit for the Collision and Streaming of qLBM
- Implement the Quantum Circuit in the Intel Quantum SDK
- Finally, Solve a simple Fluid Dynamics problem using this Circuit.
- Validate the results.




```
const int N = 5;

/* global array of qubits*/
qbit qreg[N];
cbit creg[N];

quantum_kernel void cphase(qbit ctr, qbit tgt, double phase) {
  RZ(ctr, phase / 2);
  CNOT(ctr, tgt);
  RZ(tgt, -1 * phase / 2);
  CNOT(ctr, tgt);
  RZ(tgt, phase / 2);
}

quantum_kernel void prepare() {
  for (int i = 0; i < N; i++) {
    PrepZ(qreg[i]);
  }
}

quantum_kernel void measure() {
  for (int i = 0; i < N; i++) {
    MeasZ(qreg[i], creg[i]);
  }
}

quantum_kernel void QFT() {

  int n = N - 1;
```



Oriel Kiss

Tejas Shinde

Optimisation

Quantum annealers

Juan Carlos Criado

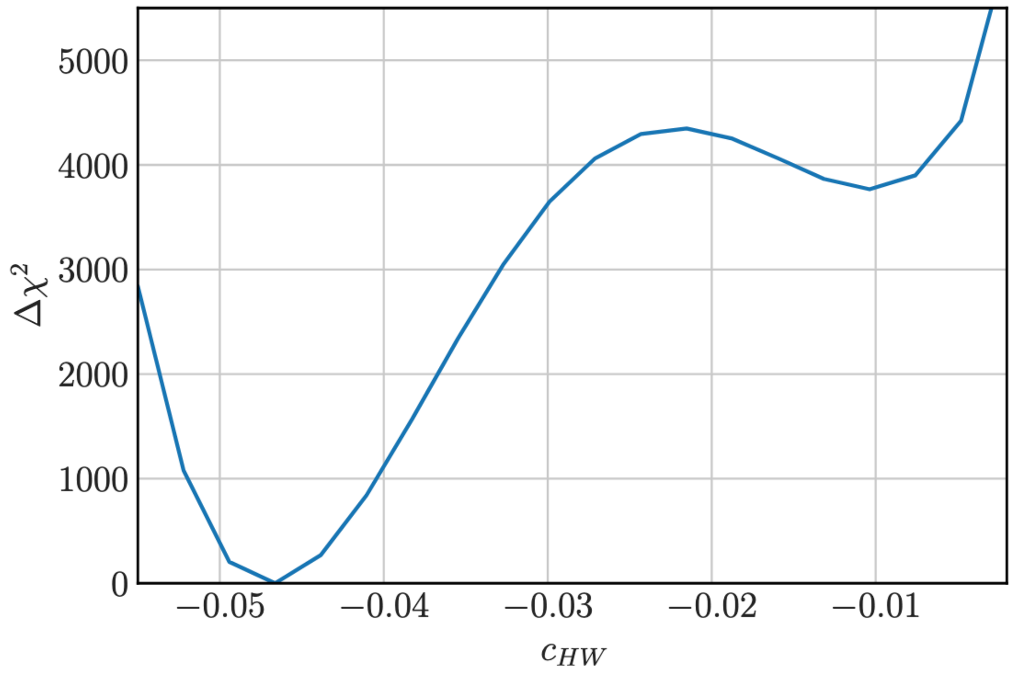
QFitter: EFT Wilson coefficient fits

2207.10088

$$\chi^2 = \sum_{ij} V_a C^{-1}_{ab} V_b, \qquad V_a = O_a^{(\mathrm{exp})} - O_a^{(\mathrm{th})}(c)$$

$$O_a^{(\mathrm{th})}(c) = A_a + \sum_i B_{ai} c_i + \sum_{ij} C_{aij} c_i c_j$$

$$\begin{aligned}\mathcal{L} = {} & \frac{c_{u3} y_t}{v^2}(\phi^\dagger\phi)(\bar{q}_L \tilde{\phi} u_R) + \frac{c_{d3} y_b}{v^2}(\phi^\dagger\phi)(\bar{q}_L \phi d_R) \\ & + \frac{i c_W g}{2m_W^2}(\phi^\dagger \sigma^a D^\mu \phi) D^\nu W^a_{\mu\nu} + \frac{c_H}{4v^2}\left(\partial_\mu(\phi^\dagger\phi)\right)^2 \\ & + \frac{c_\gamma (g')^2}{2m_W^2}(\phi^\dagger\phi) B_{\mu\nu} B^{\mu\nu} + \frac{c_g g_S^2}{2m_W^2}(\phi^\dagger\phi) G^a_{\mu\nu} G^{a\mu\nu} \\ & + \frac{i c_{HW} g}{4m_W^2}(\phi^\dagger \sigma^a D^\mu \phi) D^\nu W^a_{\mu\nu} \\ & + \frac{i c_{HB} g'}{4m_W^2}(\phi^\dagger D^\mu \phi) D^\nu B_{\mu\nu} + \text{h.c.}\end{aligned}$$

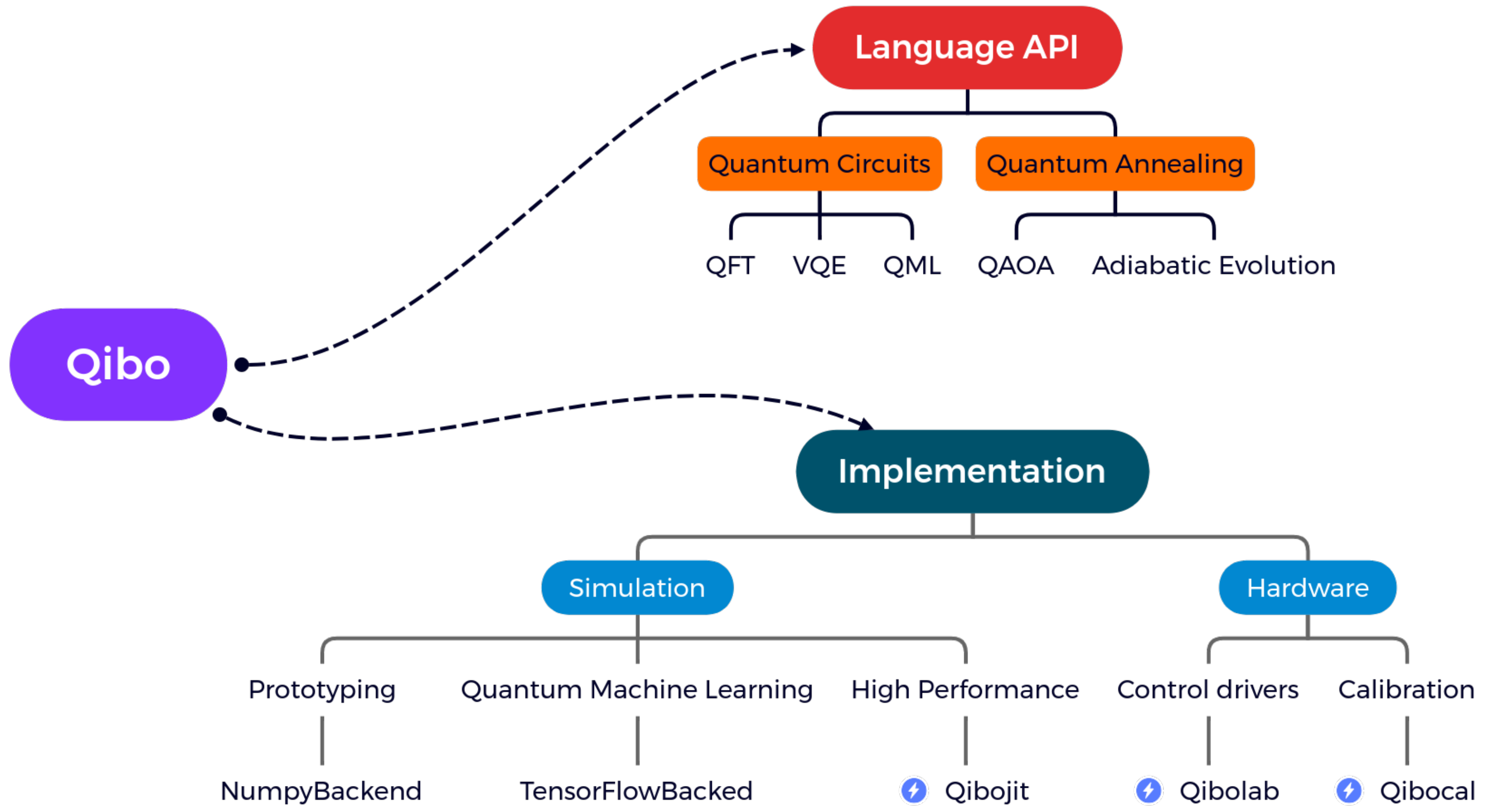


Frameworks

Andrea Pasquale

Qibo

Qibo is an **open-source** full stack API for quantum simulation and quantum hardware control and calibration.



https://github.com/qiboteam/qibo

The future

- Faster, more precise calculations and event generation
- New physics: model independent searches
- Towards Quantum Computing

There are a lot of exciting ideas for us. Happy coding!

Anke Biekötter - Leonardo Cosmai -
Joshua Davies - Latifa Elouadrhiri