Update from A1

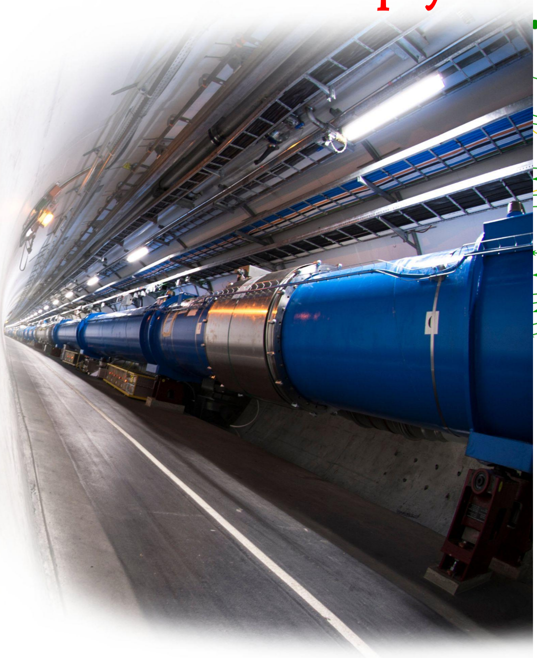
Tim Herrmann, Timo Janßen, Frank Siegert

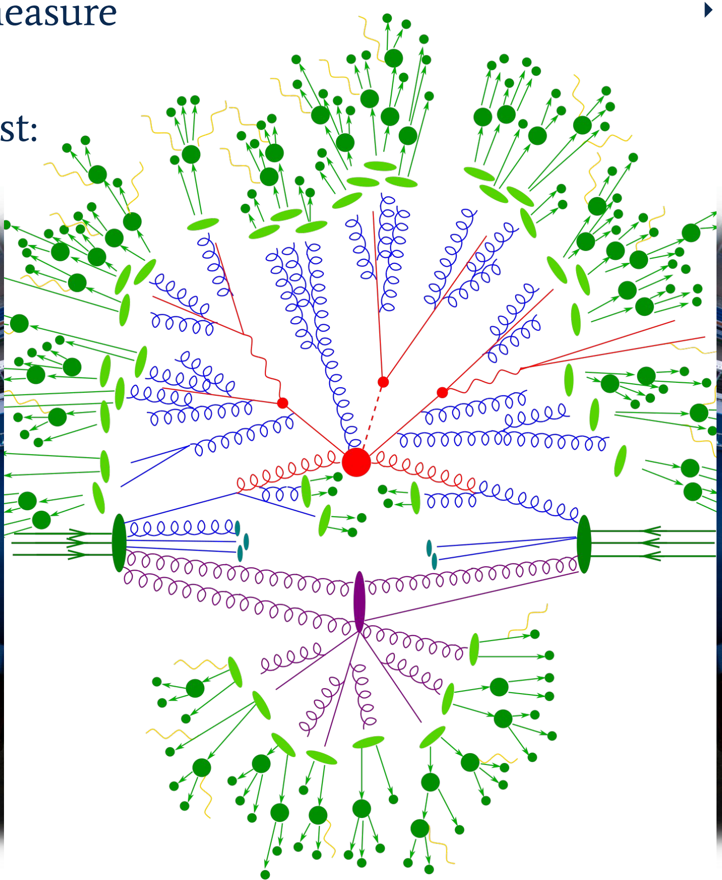
ErUM-Data KISS Annual Meeting

February 2024

Digital twins: LHC collision events

- In our detectors we measure **stable particles**
- Our theoretical interest: **fundamental physics**



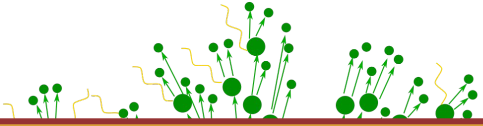


- **Connection**: Monte Carlo simulation of (QCD) dynamics

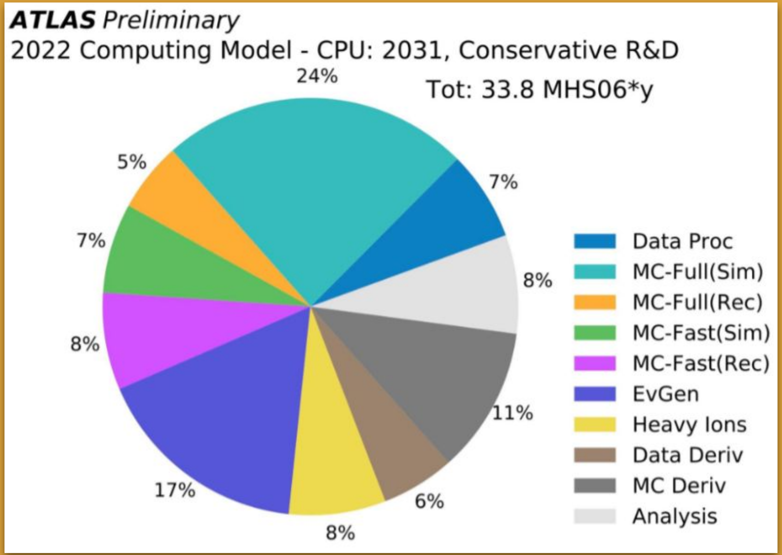
TECHNISCHE UNIVERSITÄT DRESDEN

Is event generation expensive?

- In our detector we measure **stable particles**
- Our theoretical interest: **fun**

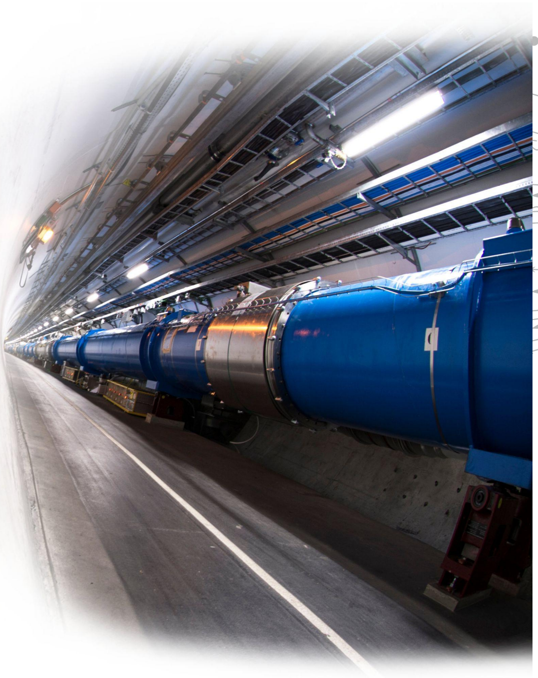
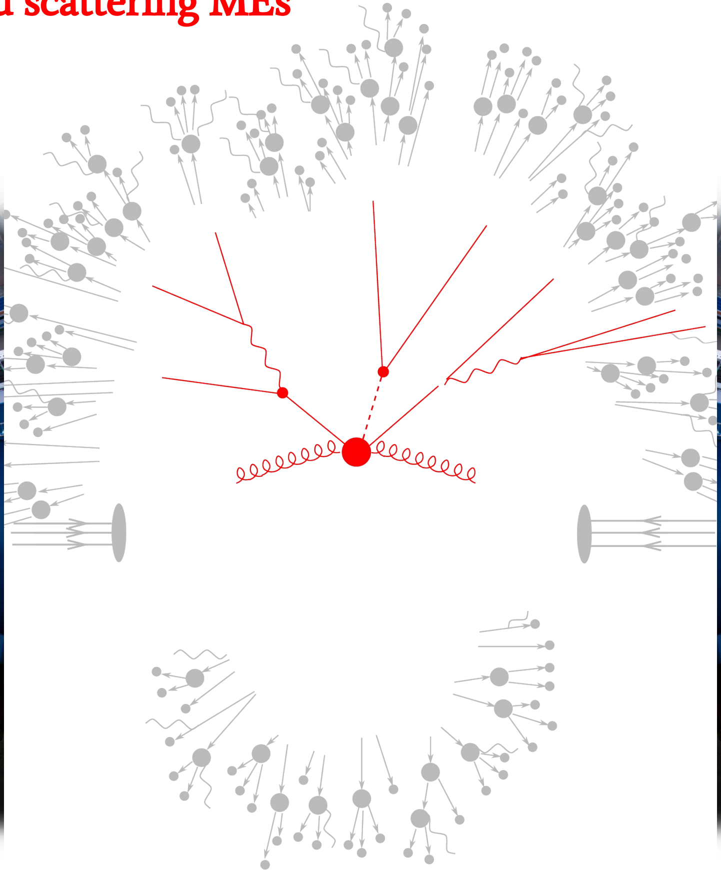


- **Connection**: Monte Carlo simulation of (QCD) dynamics



[ATLAS HL-LHC Computing Roadmap]()

- Expensive CPUh: **hard scattering MEs**

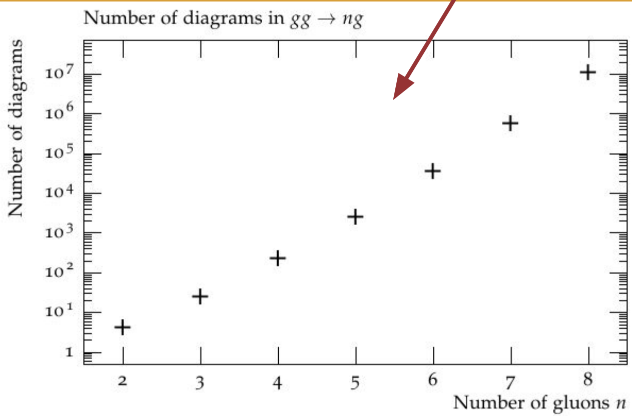




High-multiplicity matrix elements

- Expensive CPUh: **hard scattering MEs**

$$\hat{\sigma}_N = \int_{\text{cuts}} \mathrm{d}\hat{\sigma}_N = \int_{\text{cuts}} \left[\prod_{i=1}^{N} \frac{\mathrm{d}^3 q_i}{(2\pi)^3 2E_i} \right] \delta^4 \left(p_1 + p_2 - \sum_i^N q_i \right) |\mathcal{M}(p_1, p_2, q_1, \ldots, q_N)|^2$$

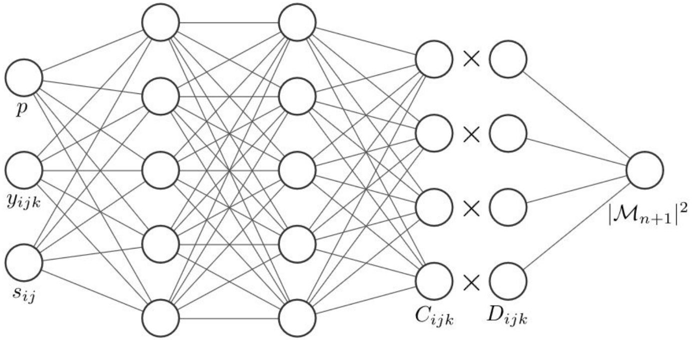
n	#diagrams
2	4
3	25
4	220
5	2,485
6	34,300
7	559,405
8	10,525,900



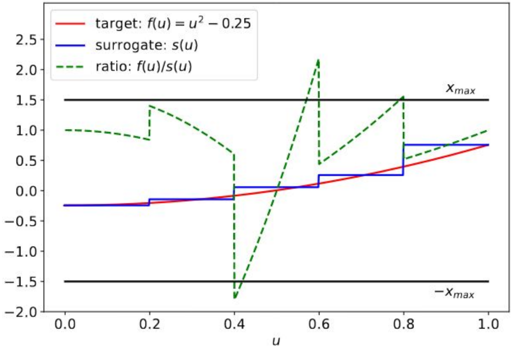
ME surrogates

KISS A1: Can we replace MEs with ML-based surrogates?

Machine Learning:
Design and train **suitable surrogate models**.



Monte Carlo:
Develop **unbiased algorithm** to use surrogates.

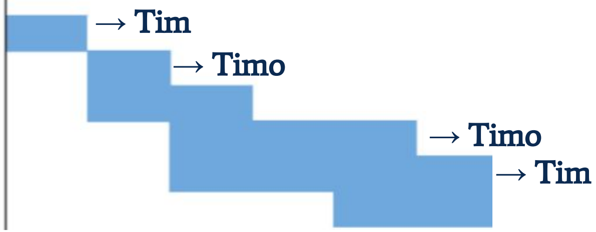


→ Monte Carlo unweighting in 2 stages: surrogate, real ME

Previous work and KISS plan

- Proof of principle:
  - MA theses (Johannes Krause, TU Dresden 2015 & Katharina Danziger, TU Dresden 2020)
  - Danziger, Janßen, Schumann, Siegert [2109.11964]
- First generalisation and more advanced NN:
  - Janßen, Maitre, Schumann, Siegert, Truong [2301.13562]
- Early milestones for KISS:

Arbeitspaket	Meilenstein		Beteiligte AGs	Monate					
				1-6	7-12	13-18	19-24	25-30	31-36
Teilprojekt A:									
A1: Erzeugung von Teilchenereignissen	M1	Schnittstelle Sherpa/Surrogate	TUD/UGOE						
	M2	Benchmarking	TUD/UGOE						
	M3	Dynamisches Training	UHD/TUD						
	M4	NLO-Genauigkeit	UGOE/TUD						
	M5	Optimierung zu Multi-Leg	TUD/UHD/UGOE						
	M6	Invertierung	UHD/UGOE						



TECHNISCHE UNIVERSITÄT DRESDEN

Tim (TU Dresden)

Modularity: Sherpa Interface

Interface available for external surrogate providers

→ available in both Sherpa2 and Sherpa3

Output from Sherpa for training:

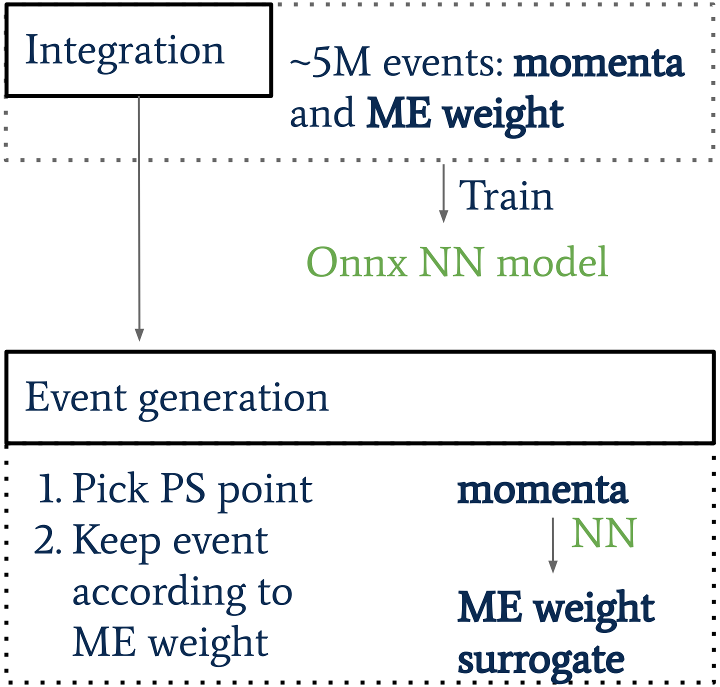
- **momenta** of initial and final state ME partons and **ME (+PhS) weight**

Training outside of Sherpa (your NN could go here):

- Train Onnx model

Input to Sherpa:

- Onnx model which calculates **ME weight surrogate** from **momenta**



NN optimisation

Original prototype: fixed hyperparameters

Problem: different scattering processes

→ very different optimal hyperparameters?

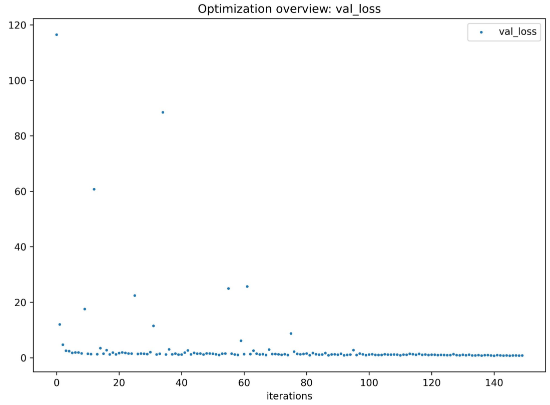
→ need flexible and automatic optimisation

Solution: OPTIMA

- Hyperparameter optimization using:
  - Bayesian optimisation and/or
  - Population based training (PBT)
- (Input variable selection → to be explored)
- Technicalities: Wrapper around
  - Tune (e.g. Keras/Lightning) and
  - Ray (Parallelisation)
- Erik Bachmann: [thesis]() and [gitlab]()

Table 1: Hyperparameters of the neural network and their values.

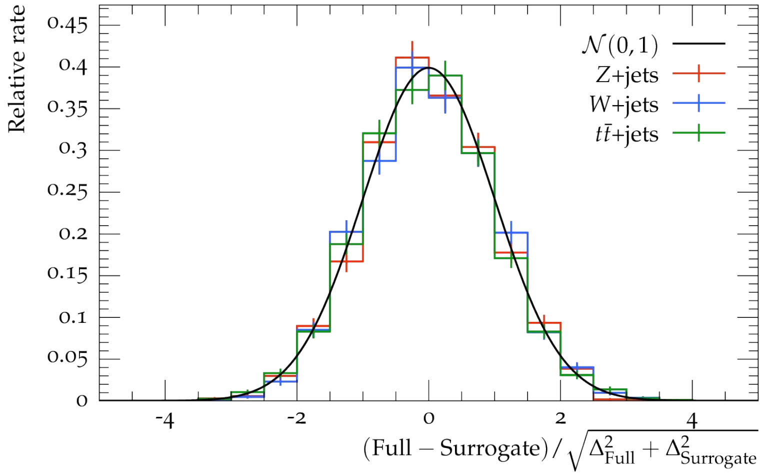
Parameter	Value
Hidden layers	4
Nodes in hidden layers	128
Activation function	swish [25, 26]
Weight initialiser	Glorot uniform [27]
Loss function	MSE
Batch size	512
Optimiser	ADAM [28]
Initial learning rate	10^{-3}
Callbacks	EarlyStopping, ReduceLROnPlateau

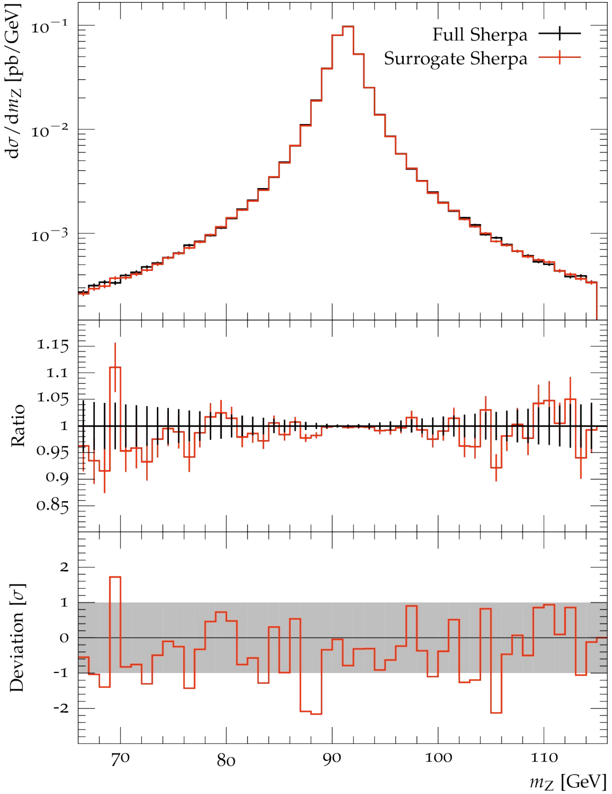


Timo (Uni Göttingen)

Validation

- for validation we compare against established tools
- use 1D histograms of physical observables
- analyse distribution of bin-wise deviations
- multivariate nonparametric tests could also be interesting (e.g. MMD, energy distance, Wasserstein distance, ...)





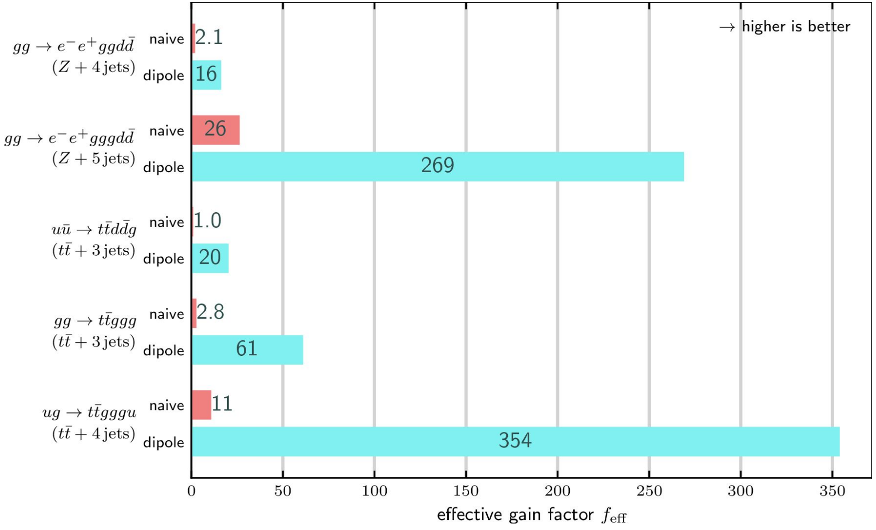
Effective gain factor

How much do we gain overall?

$$f_{\text{eff}} := \frac{T_{\text{standard}}}{T_{\text{surrogate}}}$$

$$f_{\text{eff}} = \frac{N_{\text{full}}^{\text{trials}} \cdot \left(\langle t_{\text{ME}}\rangle + \langle t_{\text{PS}}\rangle\right)}{N_{\text{1st,surr}}^{\text{trials}} \cdot \left(\langle t_{\text{surr}}\rangle + \langle t_{\text{PS}}\rangle\right) + N_{\text{2nd,surr}}^{\text{trials}} \cdot \langle t_{\text{ME}}\rangle}$$

$$= \frac{1}{\frac{\langle t_{\text{surr}}\rangle + \langle t_{\text{PS}}\rangle}{\langle t_{\text{ME}}\rangle + \langle t_{\text{PS}}\rangle} \cdot \frac{\epsilon_{\text{full}}}{\epsilon_{\text{1st,surr}}\epsilon_{\text{2nd,surr}}} + \frac{\langle t_{\text{ME}}\rangle}{\langle t_{\text{ME}}\rangle + \langle t_{\text{PS}}\rangle} \cdot \frac{\epsilon_{\text{full}}}{\epsilon_{\text{2nd,surr}}}}$$



We need:

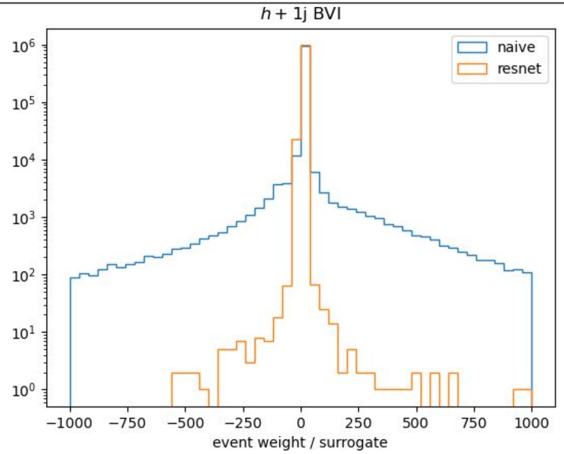
- low original unw. eff.: $\varepsilon_{\text{full}} \ll 1$
- fast surrogate: $\langle t_{\text{surr}}\rangle \ll \langle t_{\text{ME}}\rangle$
- accurate surrogate: $\varepsilon_{\text{2nd,surr}} \cong 1$

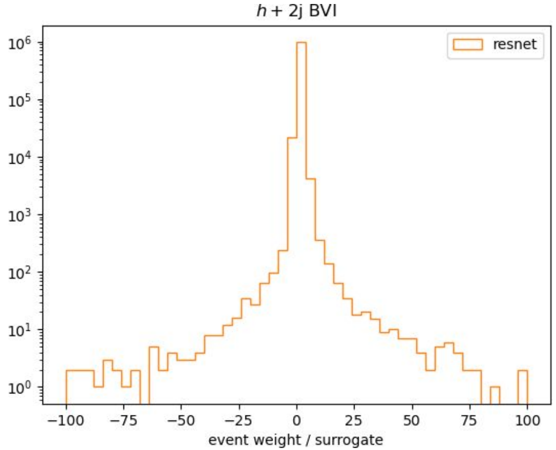
Going to Next-to-leading order

- at NLO negative weights appear
- our algorithm is easily adapted to deal with this (incl. sign errors)
- need a fast&accurate surrogate for loop amplitudes

Examples:

- Bayesian networks with boosted training, $gg \longrightarrow \gamma\gamma g(g)$ [Badger *et al.* SciPost Phys. Core 6, 034 (2023)]
- factorisation-aware model based on antenna functions, $e^+e^- \longrightarrow q\bar{q} + \{1..3\}g$ [Maître&Truong JHEP **2023**, 159]





Colour sampling

- so far we considered colour-summed matrix elements
- at some number of final-state particles, sampling the colours becomes more efficient (better scaling)
- requires fast&accurate surrogate for partial amplitudes
- challenges:
  - additional colour dimensions increase complexity
  - partial amplitudes are much faster (single term instead of large sum)
- maybe we can build a model that "knows" about the SU(3) colour structure

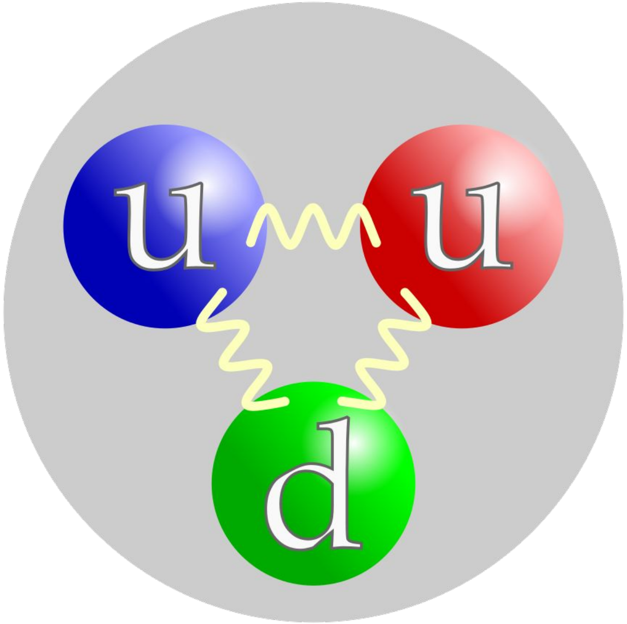


Image: Arpad Horvad (2006), Wikimedia Commons, CC BY-SA 2.5