Factorization-aware neural networks: NLO MEs and unweighting

Daniel Maître, IPPP Durham

Matrix element emulation

Work with Henry Truong

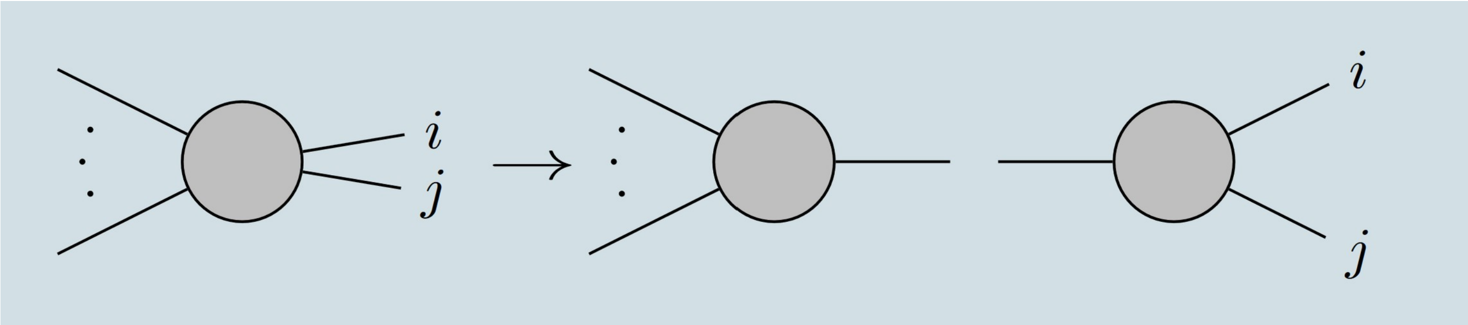
arXiv:2107.06625, arXiv:2302.04005

Difficulties emulating matrix elements

- Singular behaviour in soft and collinear limits makes a straight-forward emulation difficult:
  - Small change in phase-space input results in dramatic change in ME value
  - Selection of training set/loss can be difficult:
    - The loss can be dominated by singular configurations for loose training cuts
    - The extrapolation becomes unreliable if trained on too tight cuts

Using singular behaviour

- We know the behaviour of MEs in soft and collinear limits



- Use a NN to predict the regular factor and use the known analytic divergent behaviour
- Use a NLO subtraction-style ansatz to emulate the LO ME

Factorisation-aware emulation

- Write amplitude as an ansatz

$$\langle |\mathcal{M}_{n+1}|^2 \rangle = \sum_{\{ijk\}} C_{ijk} D_{ij,k}$$

Very redundant parametrisation

- Fit the coefficients

$$L_{\mathrm{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - p(\vec{x}_i; \theta)\right)^2$$

- Encourage NN to learn factorisation

$$L = L_{\mathrm{MSE}} + L_{\mathrm{pen}}$$

$$L_{\mathrm{pen}} = J \sum_i \frac{D_i^{-2}}{\sum_j D_j^{-2}} |C_i D_i|$$

Factorisation-aware emulation

- When approaching a singular limit only the relevant dipole is relevant
- Away from all singularities all terms combine to emulate the matrix element
- At LO we use Catani-Seymour dipoles, at one-loop we used antenna functions
- Azimuthal term added:

$$S_{ij} \sin\left(2\phi_{ij}\right) + C_{ij} \cos\left(2\phi_{ij}\right)$$

- The angle is the azimuthal angle of the decay particles in the plane perpendicular to the parent particle momentum

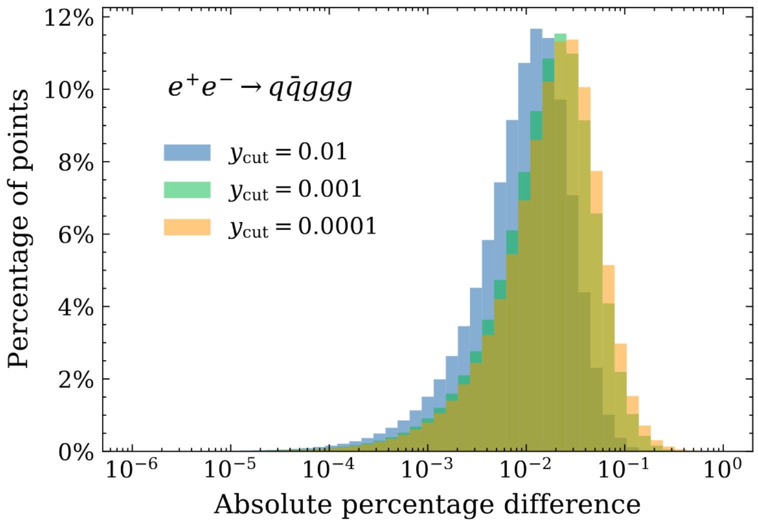
Results

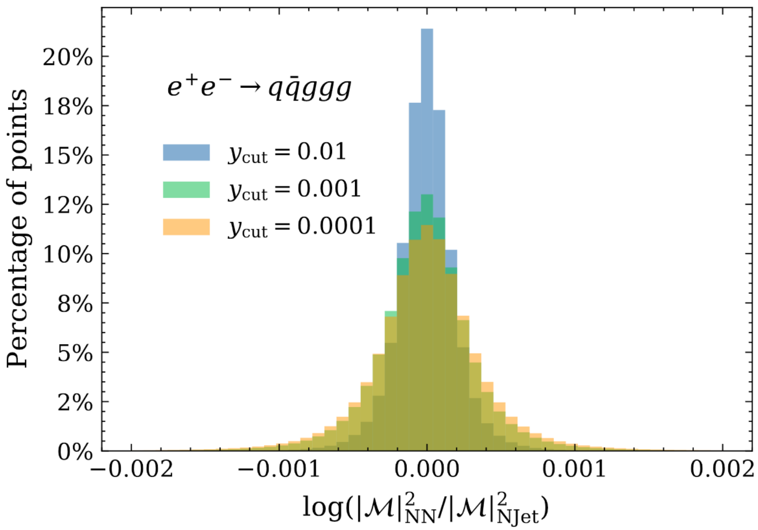
- e^+e^- annihilation into jets
- Train on 3 different training set
  - y_{cut} = 0.01, 0.001, 0.0001

- Lower cut means larger range for the matrix element
  - Expect lower precision
- Use 40M PS point training set

LO Results

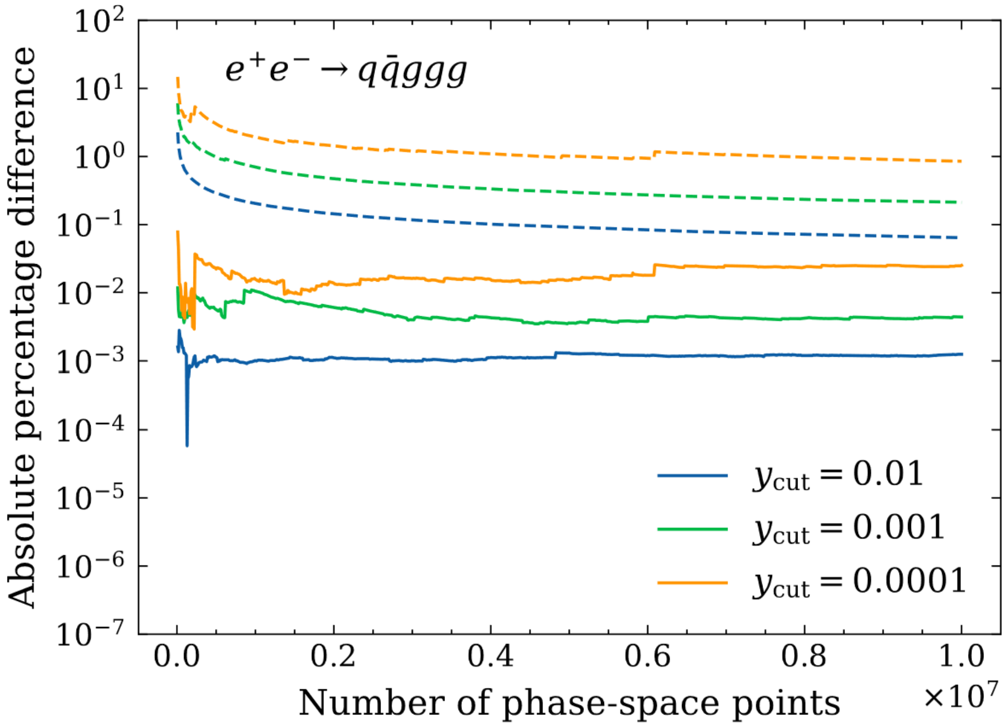
- No discernible bias
- 3-4 digits accuracy





Cross section

- Calculate the error on the total cross section for the test set
- The error is smaller than the statistical uncertainty
- The model can be used to augment the training set



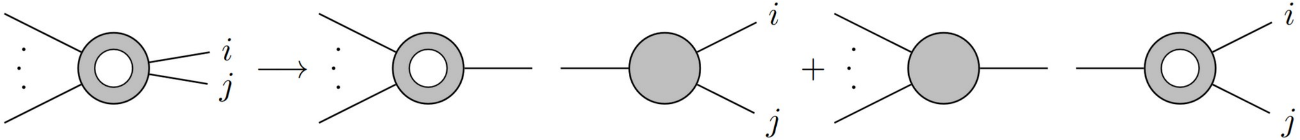
One-loop amplitudes

- Much more CPU intensive to calculate
- Choose to model the k-factor

$$k_n = \frac{2\Re\left\{\mathcal{M}^{(n,0)}\mathcal{M}^{(n,1)\,*}\right\}}{|\mathcal{M}^{(n,0)}|^2} \equiv \frac{|\mathcal{M}^{(n,1)}|^2}{|\mathcal{M}^{(n,0)}|^2}$$

- Factorisation is more complicated, we use antenna factorisation

$$|\mathcal{M}^{(n+1,1)}|^2 \longrightarrow X^0_{ijk}|\mathcal{M}^{(n,1)}|^2 + X^{1,F}_{ijk}|\mathcal{M}^{(n,0)}|^2$$



k-factor ansatz

- Given

$$k_{n+1} \longrightarrow \frac{X^0_{ijk}|\mathcal{M}^{(n,1)}|^2 + X^{1,F}_{ijk}|\mathcal{M}^{(n,0)}|^2}{X^0_{ijk}|\mathcal{M}^{(n,0)}|^2}$$

$$k_{n+1} \longrightarrow \frac{|\mathcal{M}^{(n,1)}|^2}{|\mathcal{M}^{(n,0)}|^2} + \frac{X^{1,F}_{ijk}}{X^0_{ijk}}$$

$$k_{n+1} \longrightarrow k_n + \frac{X^{1,F}_{ijk}}{X^0_{ijk}}$$

- We use the ansatz:

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

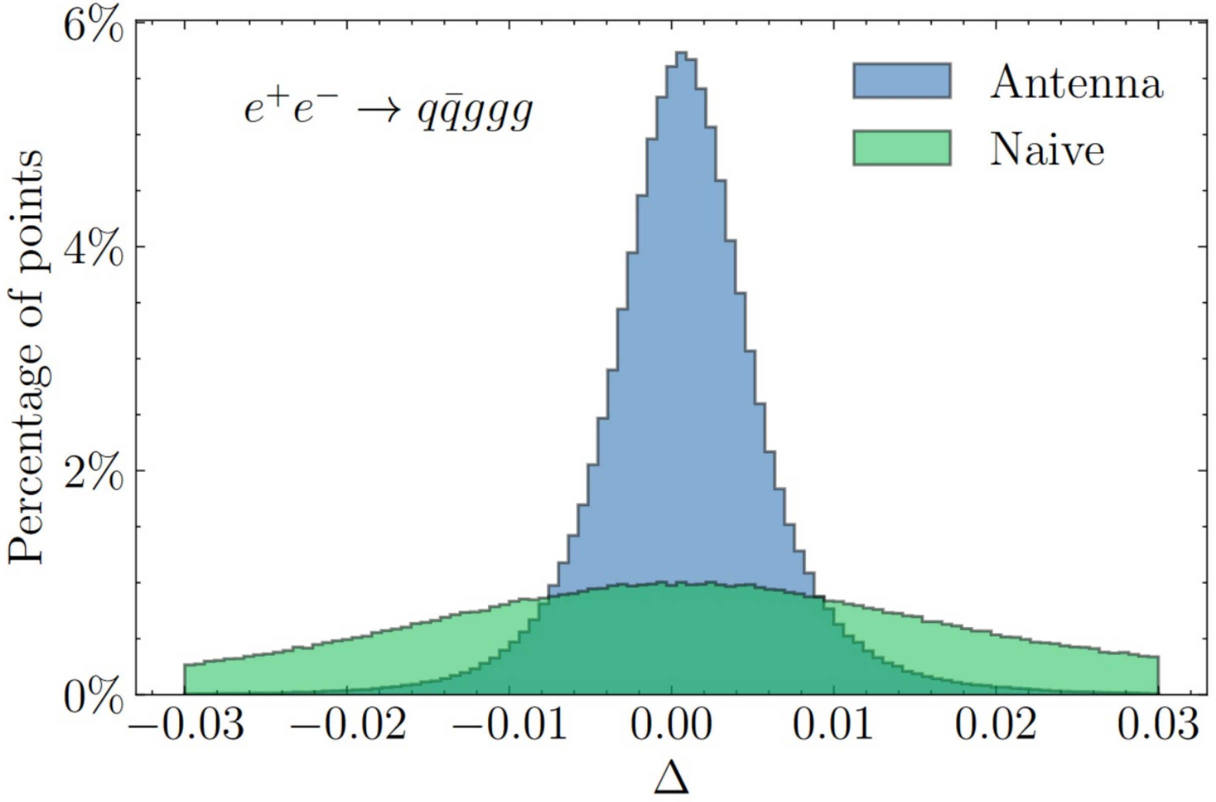
1L results

- Evaluate precision using

$$k_{\text{true}} - k_{\text{pred}} = \frac{|\mathcal{M}^{(n,1)}|^2_{\text{true}} - |\mathcal{M}^{(n,1)}|^2_{\text{pred}}}{|\mathcal{M}^{(n,0)}|^2_{\text{true}}} = \Delta$$

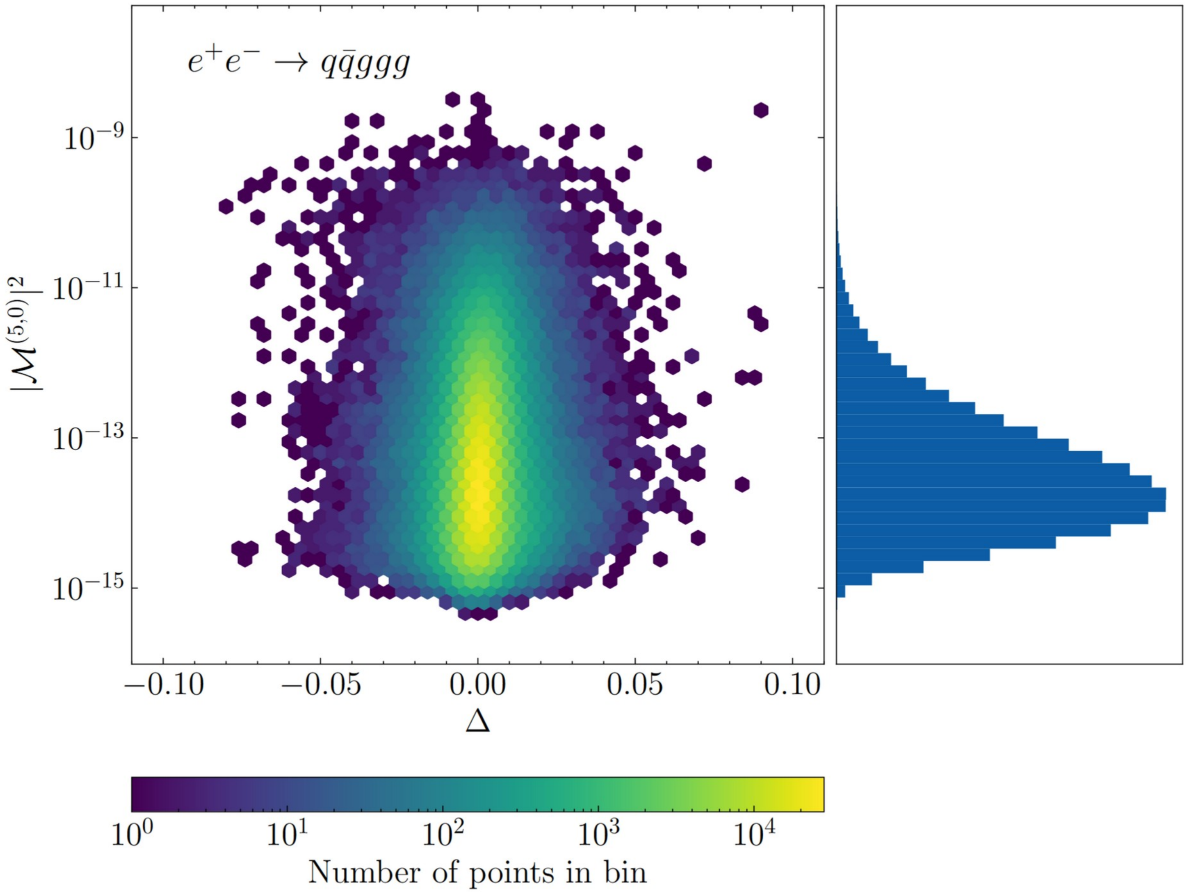
1L Results

- Train on 100k PS points
- Test on 1M points
- Compare with a single NN model for the *k*-factor

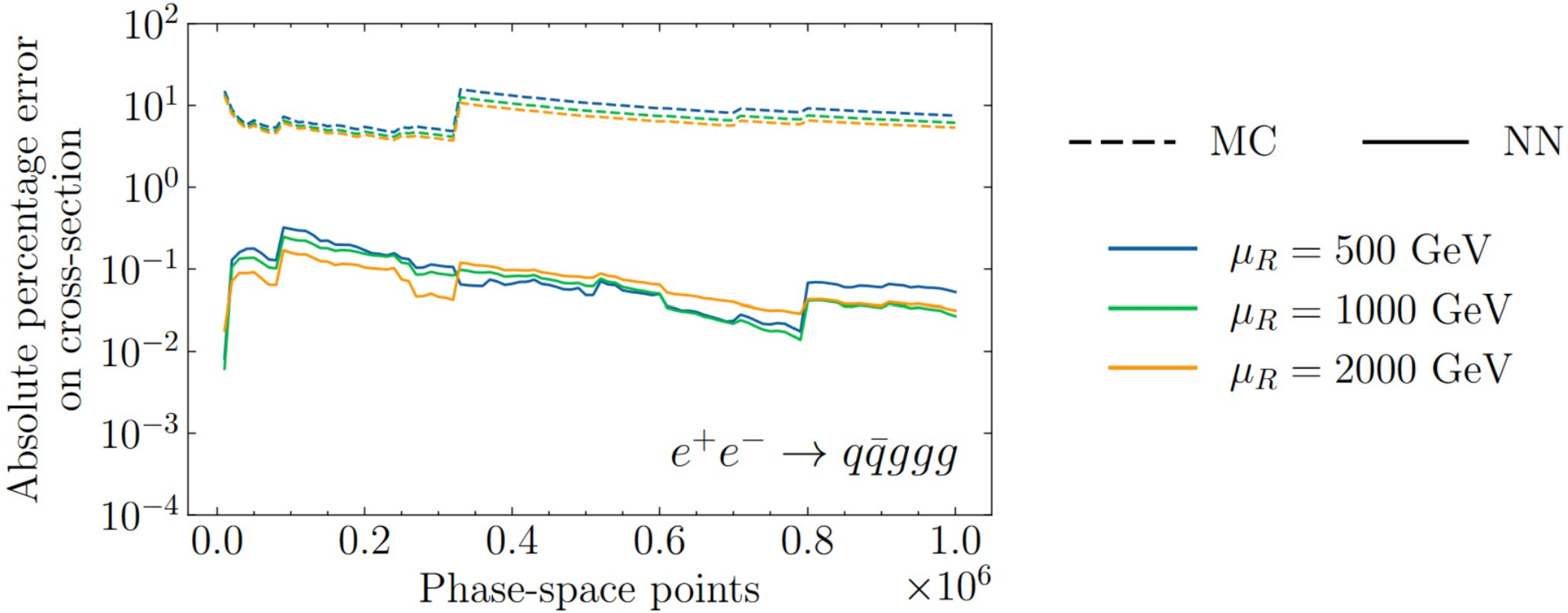


Error vs ME size

- No clear bias w.r.t weight size

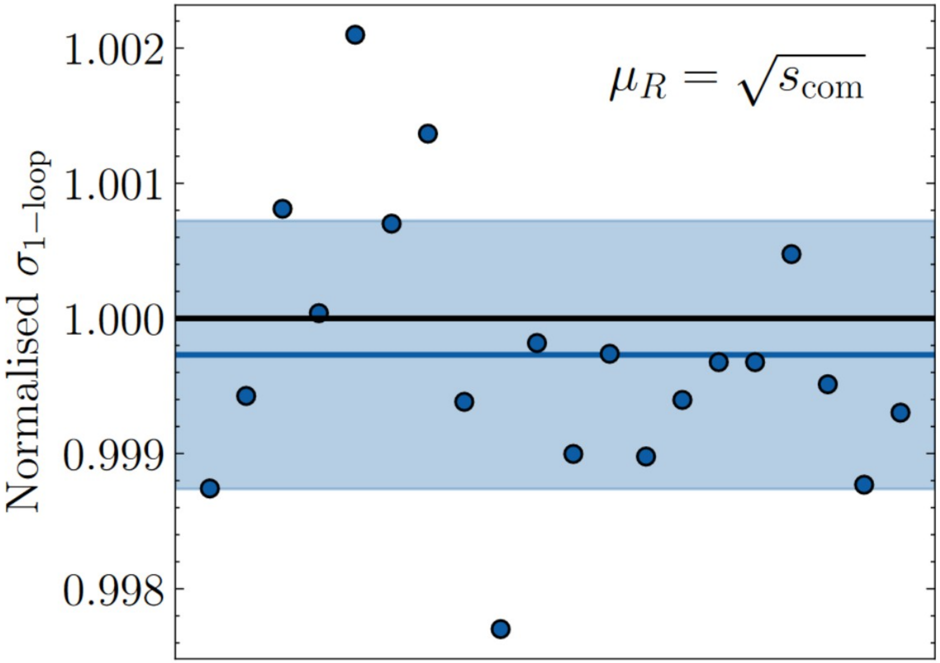


Cross section



Augmenting dataset

- Use 20 replicas to estimate variance and bias of model
- Statistical MC integration error is of order 10%!
- Can use NN model to augment dataset!
- Use variance as an estimate of the NN error



Unweighting with NN approximation

Work with Timo Janßen, Stefan Schumann, Frank Siegert and Henry Truong [arXiv:2301.13562]

Unweighting

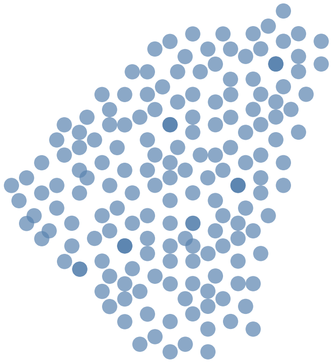
- Unweighting high multiplicity matrix elements can be extremely unefficient
- Can be improved by
  - Generate a set of unweighted values according to an approximation that is easy to unweight
  - Unweight this set according to
  - If ratio is close to 1 we get to keep many more calculated ME
- Tolerate a small amount of weights above 1

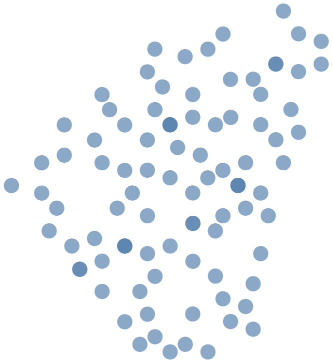
Two-stage unweighting

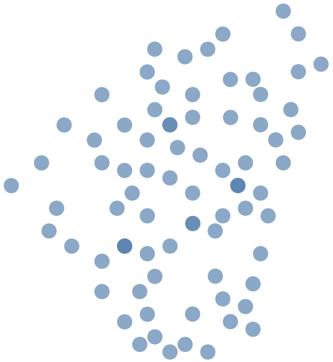
PS points

Cheap ME surrogate

True ME





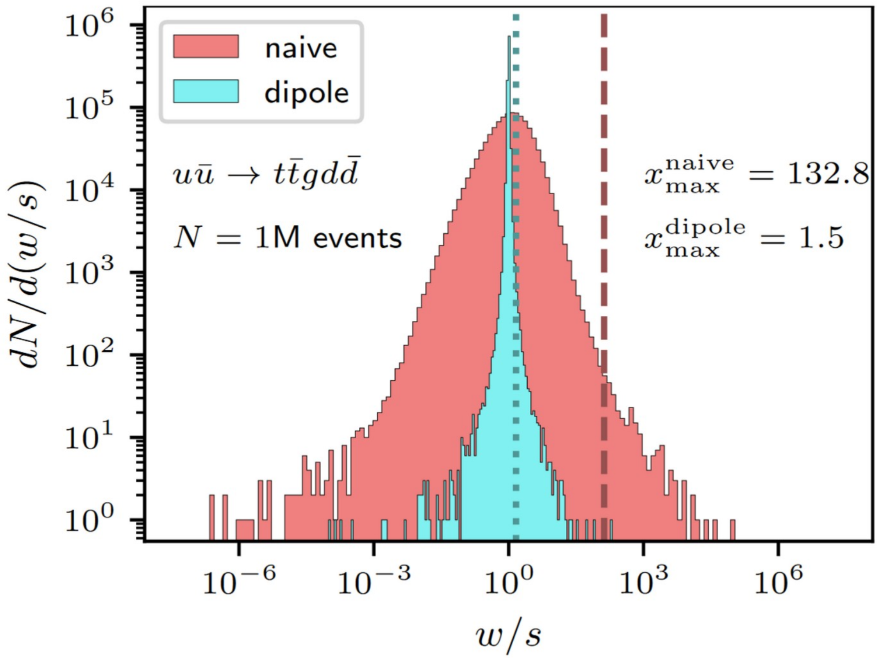


Two-stage unweighting in Sherpa

- K. Danziger, T. Janßen, S. Schumann, F. Siegert implemented such a two-stage unweighting in Sherpa and used a NN surrogate [arXiv:2109.11964]
- Z/W +4 jets and +3 jets
- Obtained speed up of up to 10 compared with AMEGIC
- Use a factorisation aware emulator instead of their NN surrogate
- Needed to implement initial-state and massive dipoles

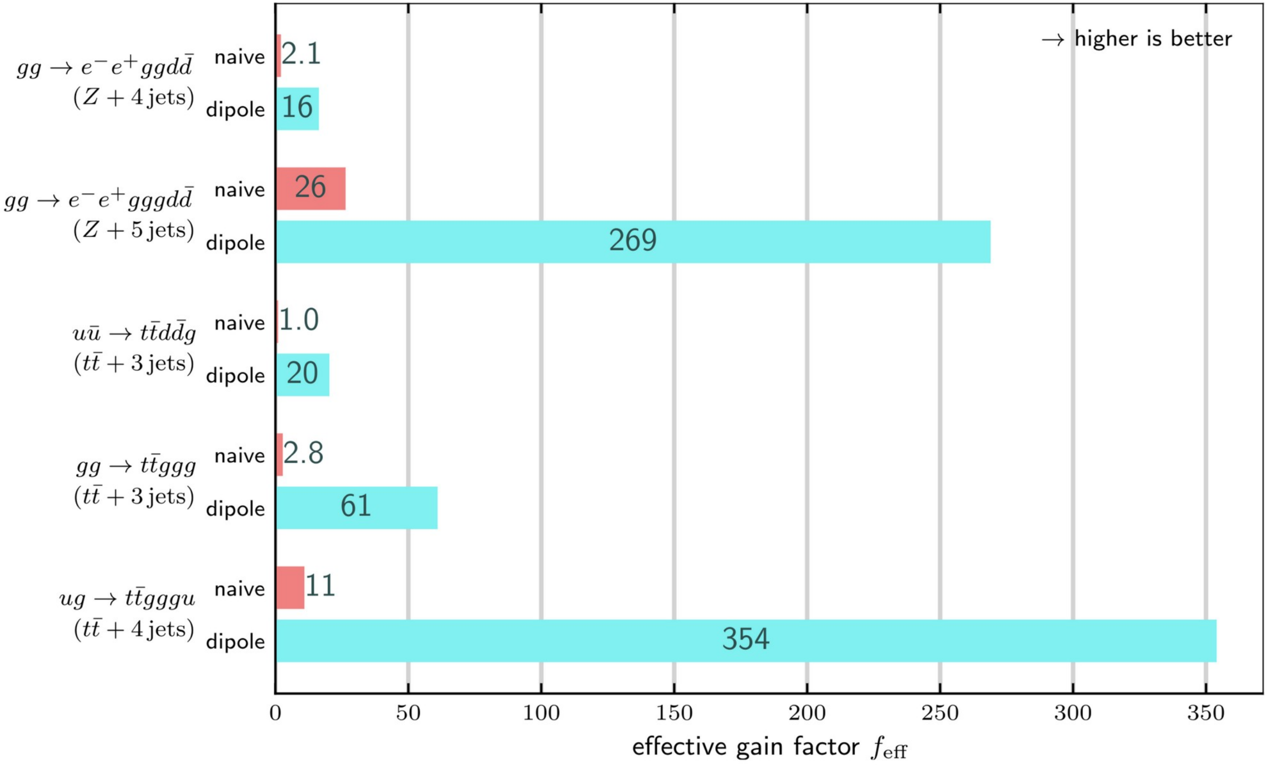
Unweighting

- First unweight w.r.t NN model , then correct with true weight
- Factorisation-aware NN is much more precise
- Up to Z/W+5 jets and tt+4 jet
- Result in very large efficiency gains (16-350)
- Largest gains for the most complicated processes



(b) Channel $u\bar{u} \to t\bar{t}gd\bar{d}$ ($t\bar{t} + 3$ jets).

Unweighting results



Caveat

- There are caveats:
  - This is using color-summed MEs
  - In practice color-sampled MEs are used for high multiplicity processes
  - A straight-forward attempt at generalising the method to color sampled MEs did not give as good results
    - More thoughts have to be put into this!

Conclusions

- NN can approximate MEs very well if some physics information is injected!
- They can be modelled with limited training data and small NN
- Precision is not perfect but:
  - Can be way smaller than the statistical error
  - Can be used as a first stage:
    - Unweighting: two (or more?) stage
    - Integration: integrate (ME-NN) if precision is not sufficient