Scaling laws for amplitude surrogates

Joaquín Iturriza Ramirez

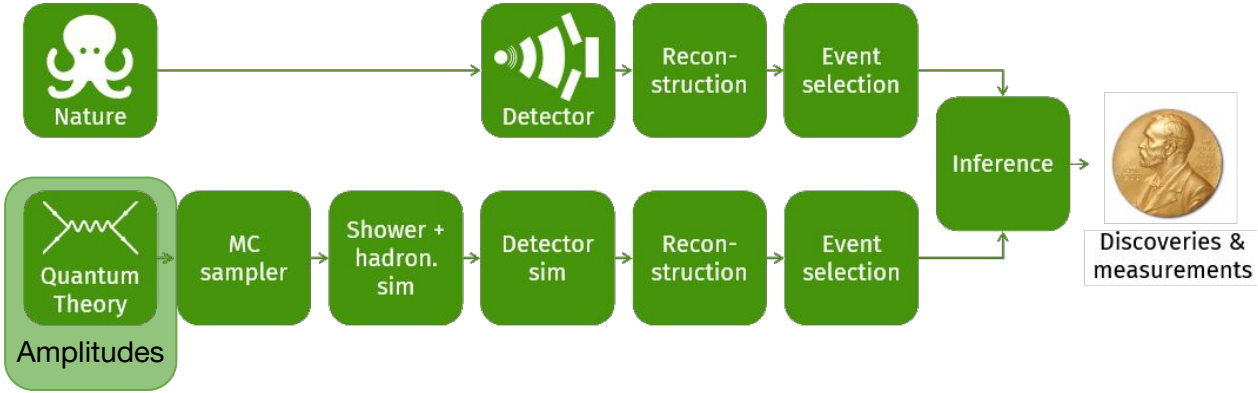
In collaboration with Anja Butter, Bertrand Laforge, Víctor Bresó Pla and Henning Bahl

IRN Terascale @ IJCLab Orsay

arXiv:2601.13308v1

Co-funded by the European Union

Motivation



Typical workflow starts with scattering amplitudes

Slow for complicated processes (higher order calculations, many particles in final state)

Interpolation reduces computation costs:

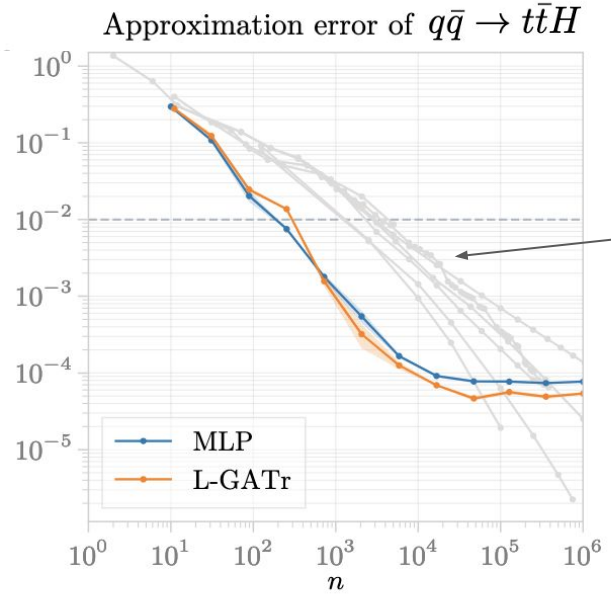
- Error must be negligible compared to the rest of the pipeline

Limitation of classical methods

- Hard to incorporate new data
- Curse of dimensionality

Motivation

ML surrogate models are excellent interpolates



Classical interpolation

Straight line in log-log plot and plateau

Motivation

Similar scaling behaviours have been observed in many applications of deep learning

Large Language Modeling[1]

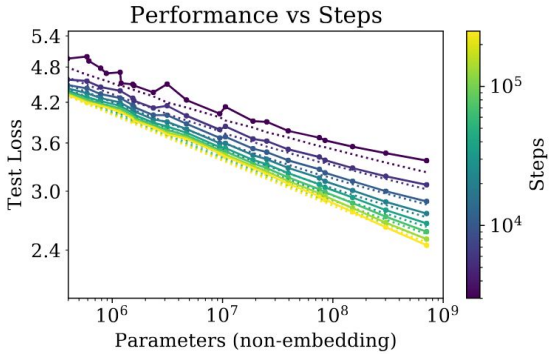
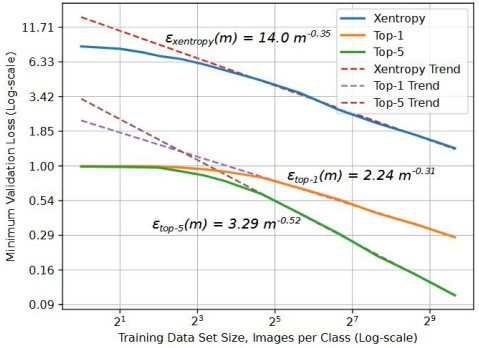
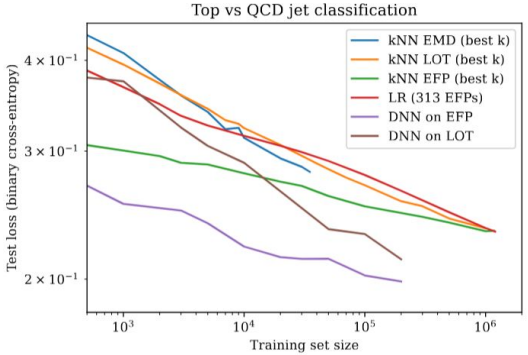


Image Classification[2]



Jet Classification[3]



Performance improves predictably as a power law with **training dataset size** D**, computing resources** C and **number of parameters** N

$$L(X,Y,Z) = (X_c/X)^{\alpha_X} + K(Y,Z)$$

Exponent of power law could be related to **intrinsic dimensionality**[4]

[1] arXiv:2001.08361, [2] arXiv:1712.00409, [3] arXiv:2312.02264, [4] arXiv:2102.06701

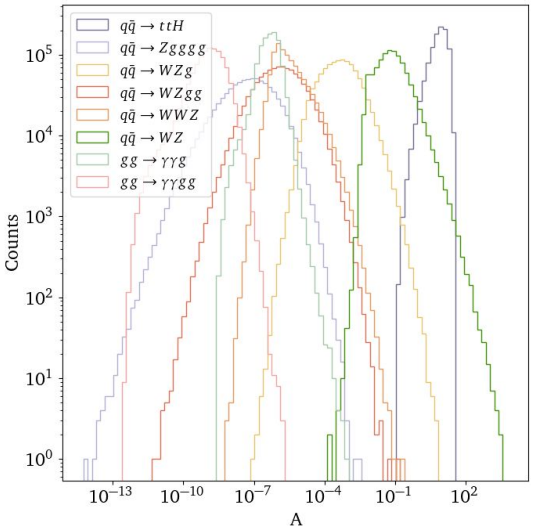
Data

4-momentum of particles in the process → **Amplitude**

All the data is generated with MadGraph, all amplitudes calculated at **lowest order**

Main case study $q\bar{q} \to t\bar{t}H$

Jet-associated Z production: $q\bar{q} \to Z + ng$
Jet-associated W Z production: $q\bar{q} \to WZ, q\bar{q} \to WZg, q\bar{q} \to WZgg$
W W Z production: $q\bar{q} \to WWZ$
Jet-associated di-photon production: $gg \to \gamma\gamma g, gg \to \gamma\gamma gg$



A **diverse set** of processes covering a broad range of physical characteristics

Symmetries in ML

Neural networks use resources learning the structure of the data

Performance improves substantially by using the structure in the data to our advantage

Particle physics respects **space-time symmetries**

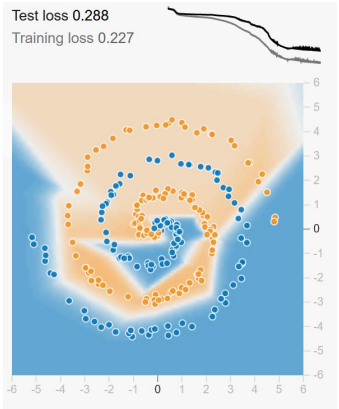
We can:

- Preprocess the data ⟶ Train on Lorentz invariant quantities
- Work with **Lorentz equivariant** networks

A Neural Network Playground

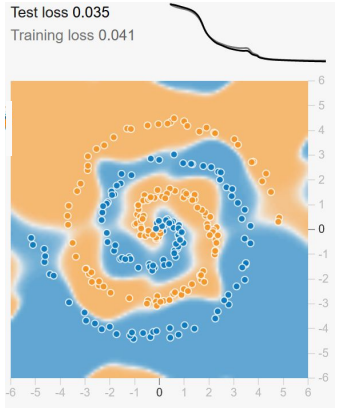
Trained on

x_1, x_2

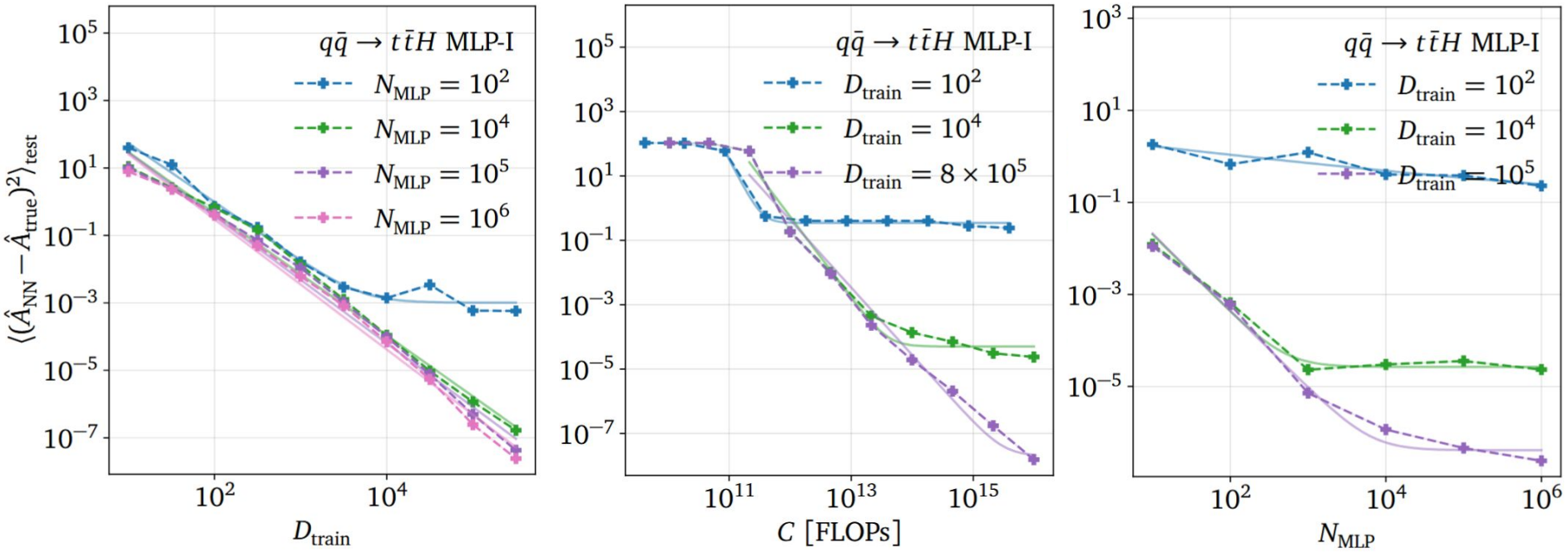


Trained on

$\sin(x_1), \sin(x_2)$



Results: MLP-I, $q\bar{q} \to t\bar{t}H$, MSE loss



Very clean power laws, consistent slopes

Losses

MSE: $\mathcal{L}_{\text{MSE}} = \left\langle (A_{\text{true}}(x) - A_{\text{NN}}(x))^2 \right\rangle_{x \sim D_{\text{train}}}$

Heteroscedastic Loss:

Assume amplitude regression follows a normal distribution $p(A|x) = \mathcal{N}(A|\bar{A}(x), \sigma^2(x))$

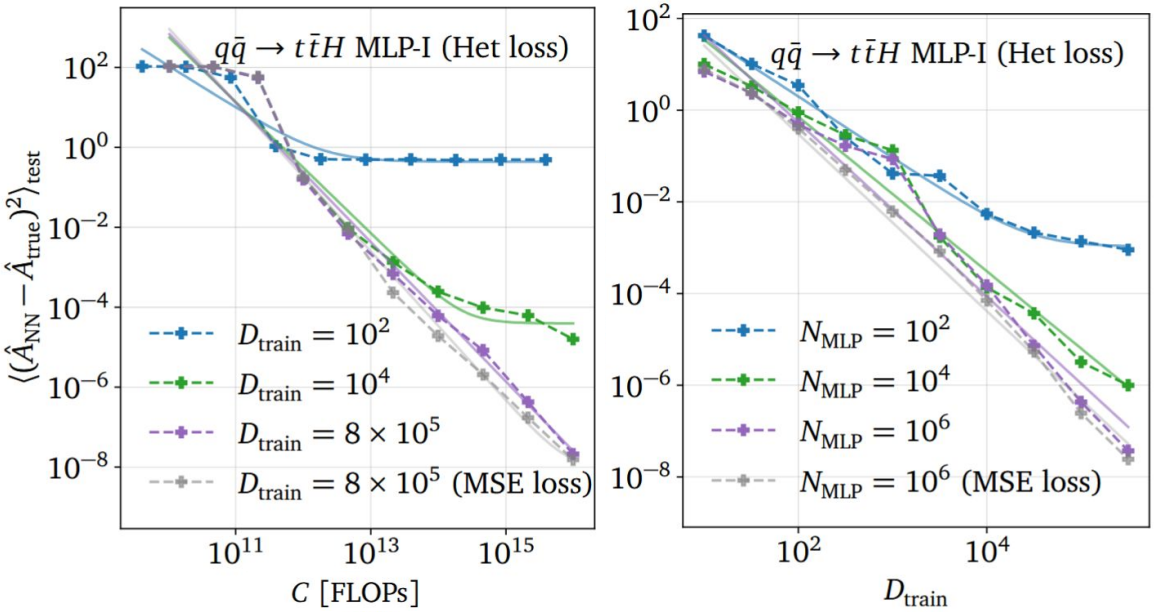
Minimize negative log-likelihood:

$$\mathcal{L} = -\left\langle \log p(A|x) \right\rangle_{x \sim D_{\text{train}}} \longrightarrow \mathcal{L}_{\text{het}} = \left\langle \frac{(A_{\text{true}}(x) - \bar{A}(x))^2}{2\sigma^2(x)} + \log \sigma(x) \right\rangle_{x \sim D_{\text{train}}}$$

The NN predicts 2 outputs: $\bar{A}(x)$ and $\sigma(x)$

If it's well calibrated $t(x) = \dfrac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma(x)}$ should follow a $\mathcal{N}(0,1)$

Results: MLPI, $q\bar{q} \to t\bar{t}H$, Heteroscedastic loss

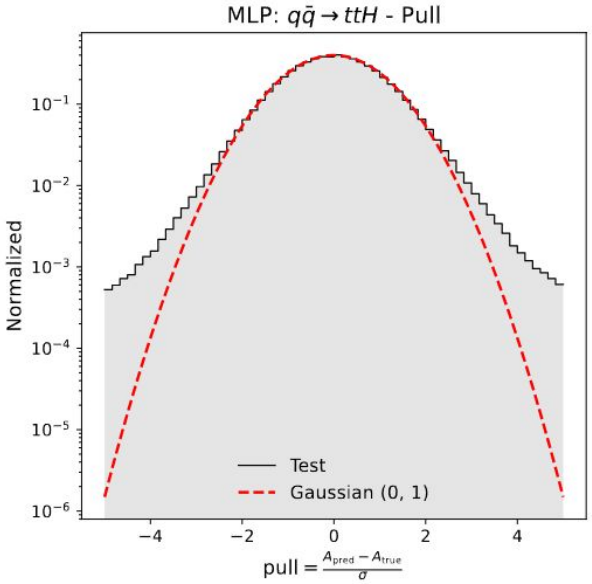


Similar behavior overall, slightly worse performance

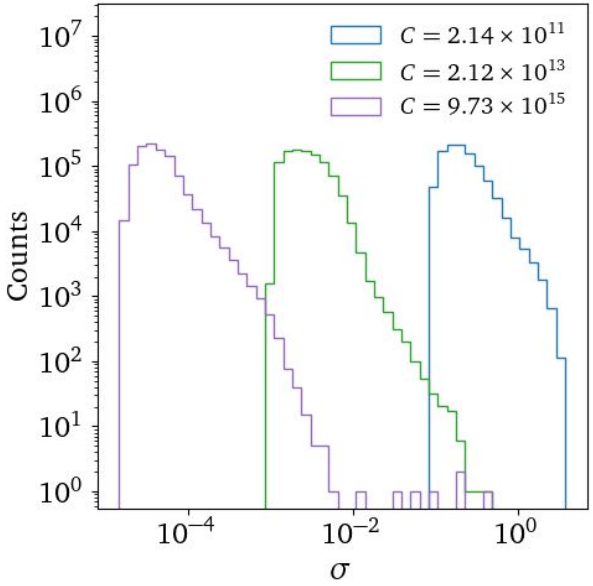
Important to note: Trained on Heterosc loss, showing MSE

Uncertainties

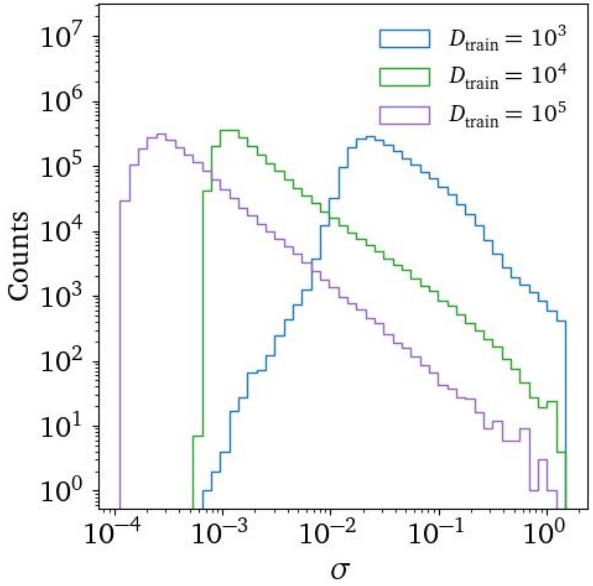
Calibration



N=10^6

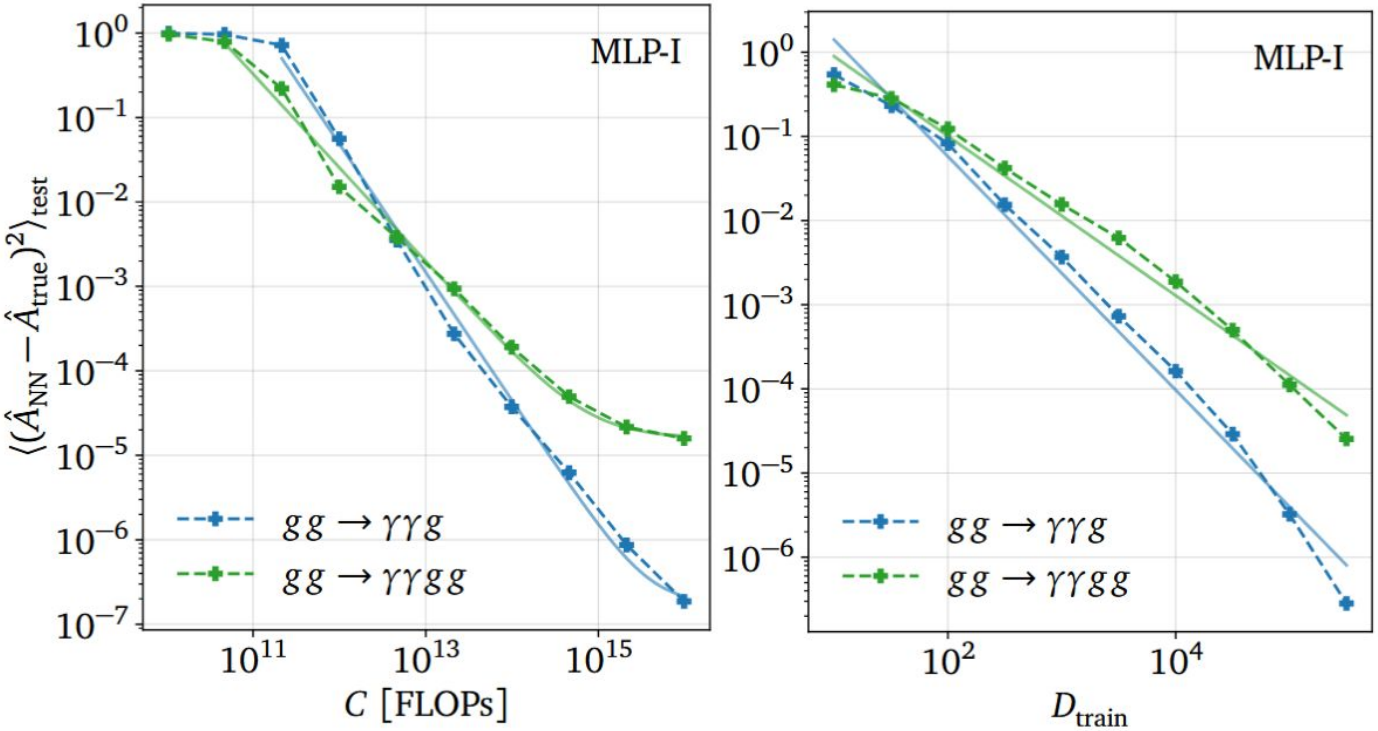


N=10^4



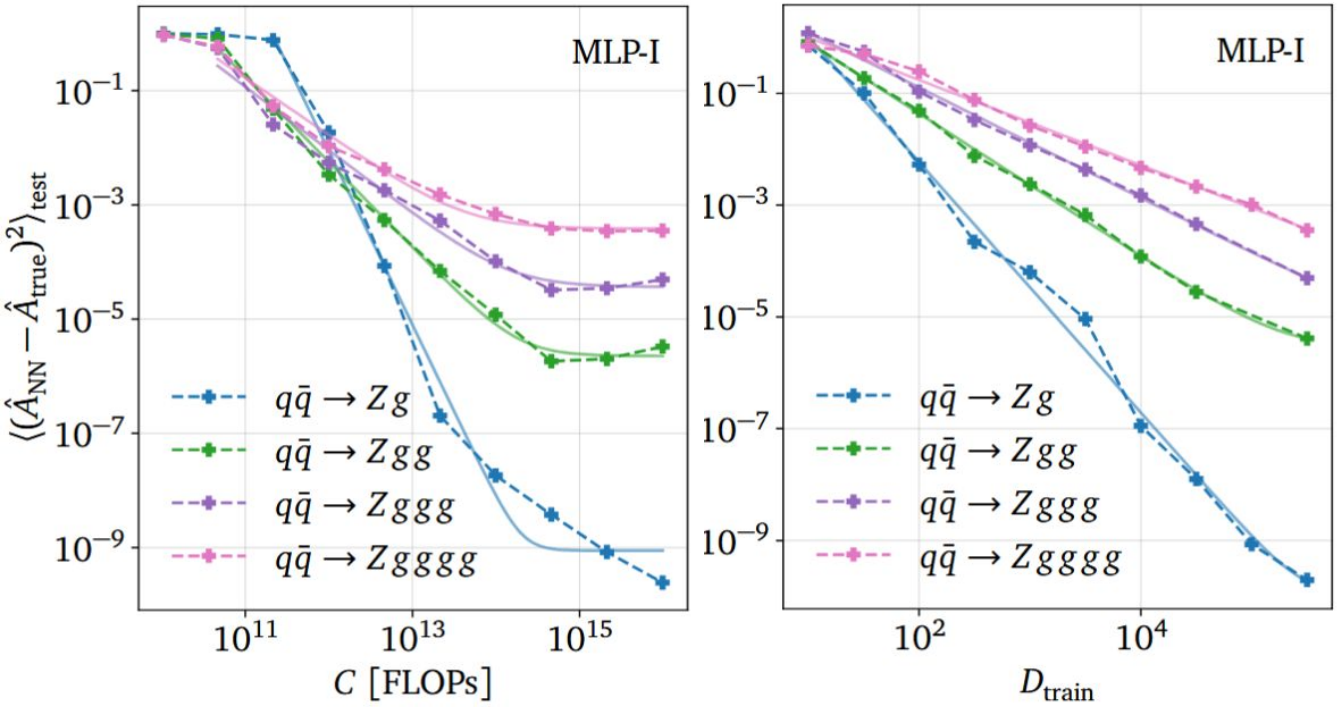
Well calibrated uncertainties for large enough dataset sizes

Scalings



Simpler process improves much faster

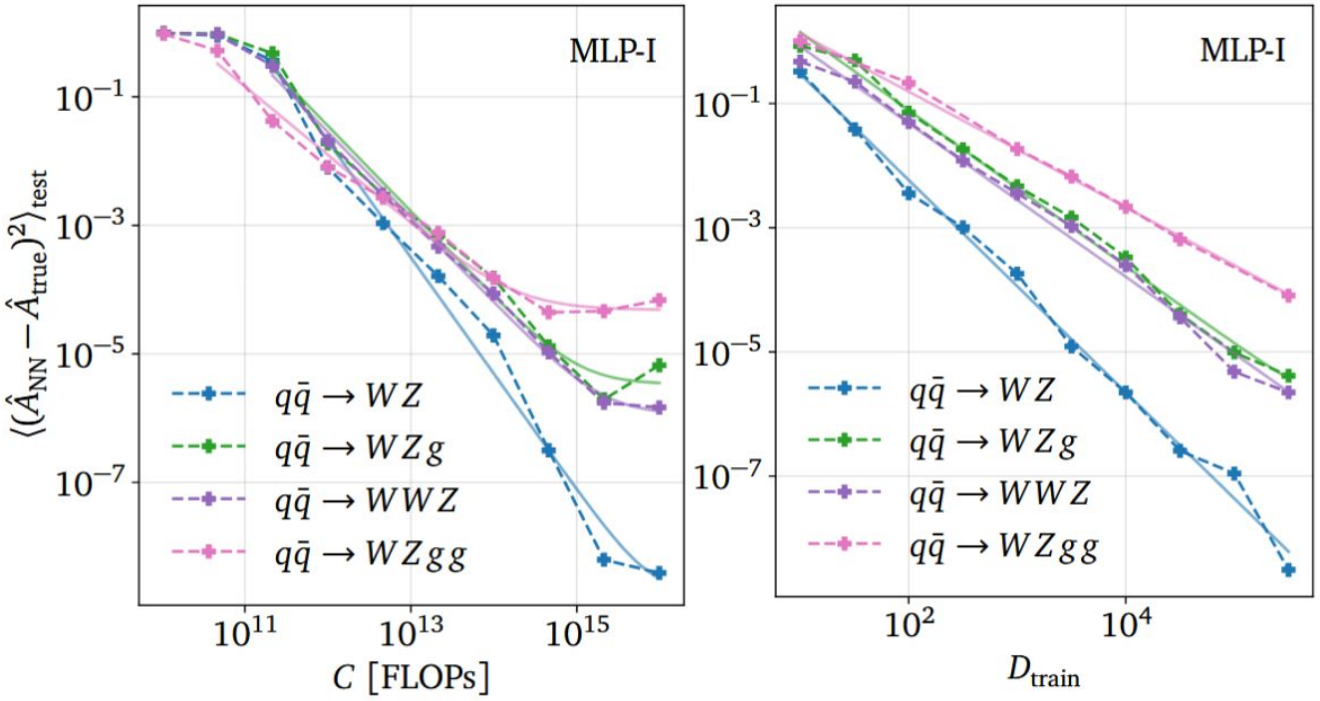
Scalings



The pattern holds with more particles

Well defined scaling laws

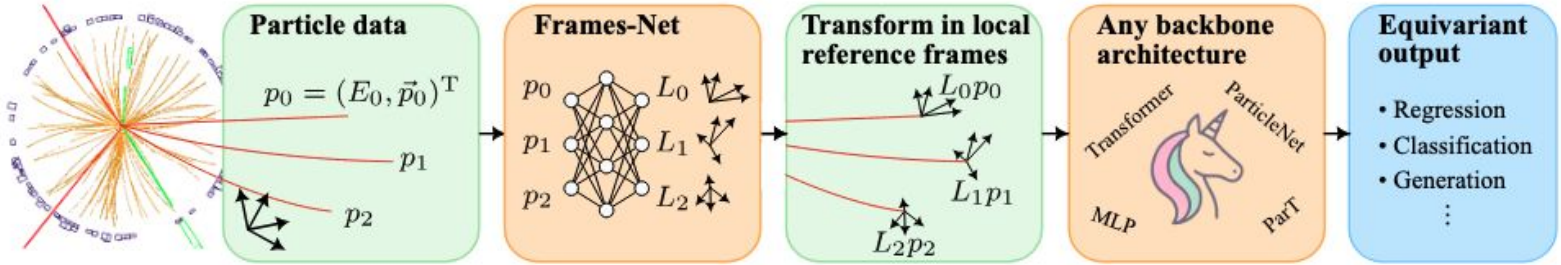
Scalings



Despite being physically very different processes, $q\bar{q}\to WZg$ and $q\bar{q}\to WWZ$ behave similarly

LLoCa[1]

Lorentz **Lo**cal **Ca**nonicalization



Learns the local frames of each particle and transforms them to their local frame

For any transformation $\mathcal{N}$ of the neural network and any Lorentz transformation $\mathcal{G}$

$$\mathcal{N}(\mathcal{G}(x)) = \mathcal{G}(\mathcal{N}(x))$$

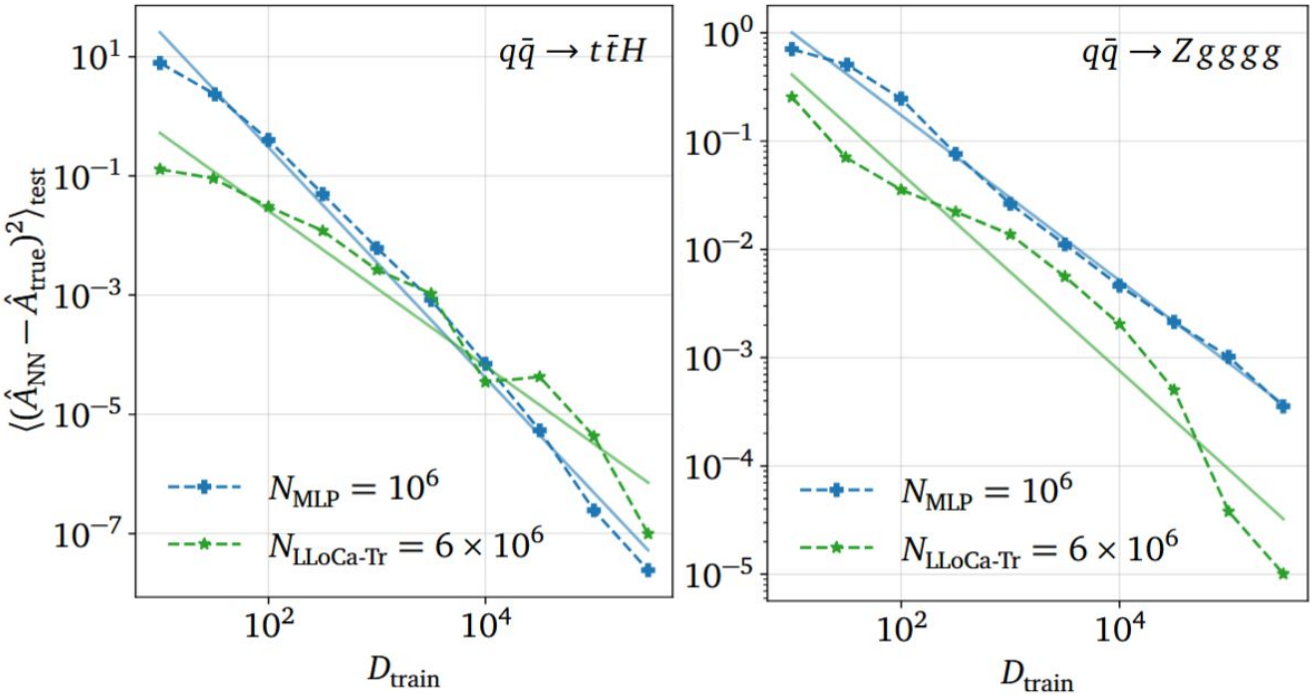
Vectors are transformed to a common frame for message passing (ex. attention)

Makes any backbone network **Lorentz equivariant**

[1] arXiv:2505.20280

LLoCa scalings

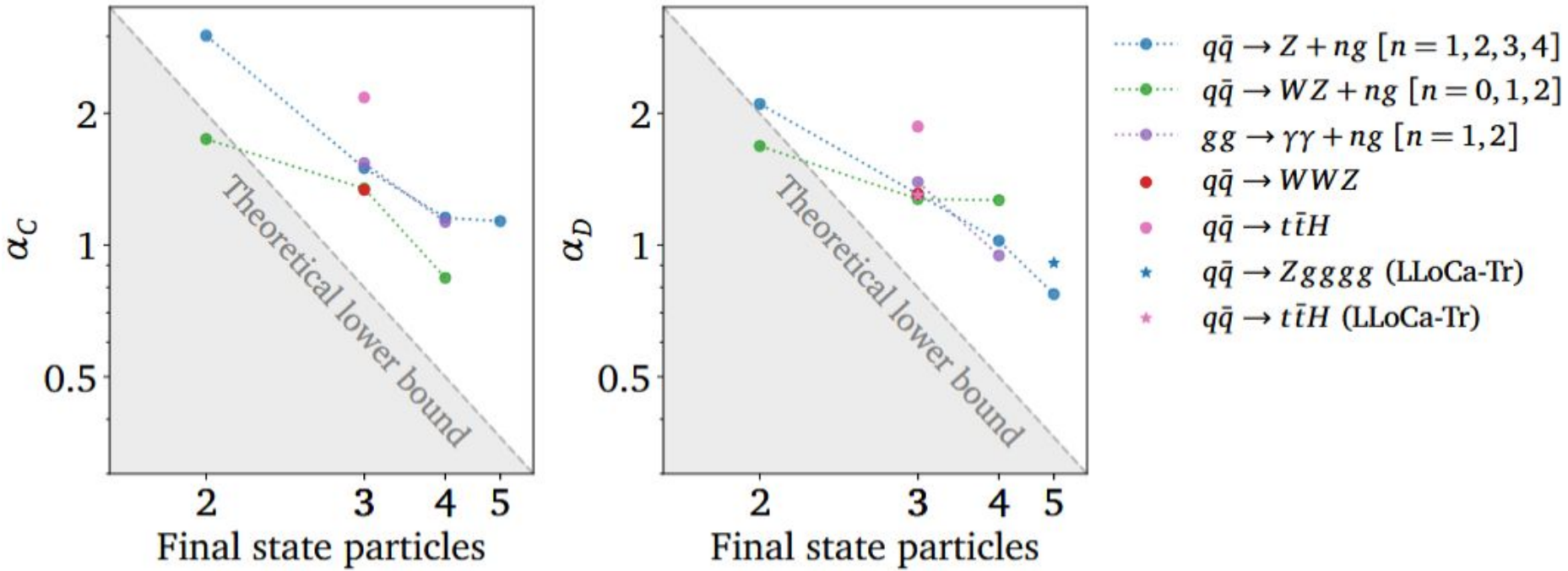
HP optimization was costly and difficult



Slightly worse performance on simple process, uniformly better performance on hard process

Scaling seems to be roughly similar

Scalings and degrees of freedom



Clear trend between scaling exponent and N. of particles / DOFs

Conclusions and future work

- Observed very clean and predictable scaling laws for amplitude surrogates
- Dataset size is the dominant bottleneck
- Scaling with MSE and Heterosc loss
- Well calibrated uncertainties across many orders of magnitude
- Observed relationship between scaling and DOFs
- Promising results with equivariant NNs

Next step: Foundational model for amplitude regression

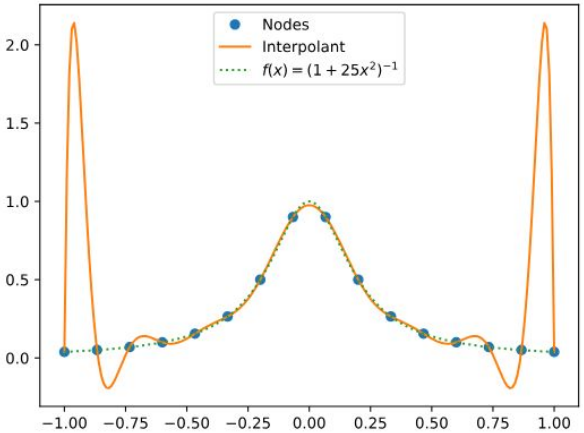
Thanks!

Motivation

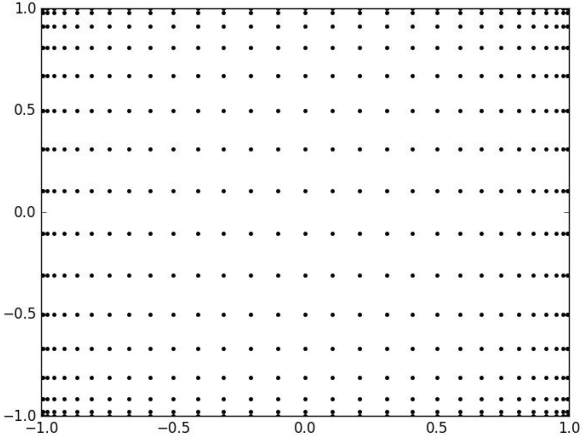
Interpolation problems:

Runge's phenomenon

Hard to incorporate new data

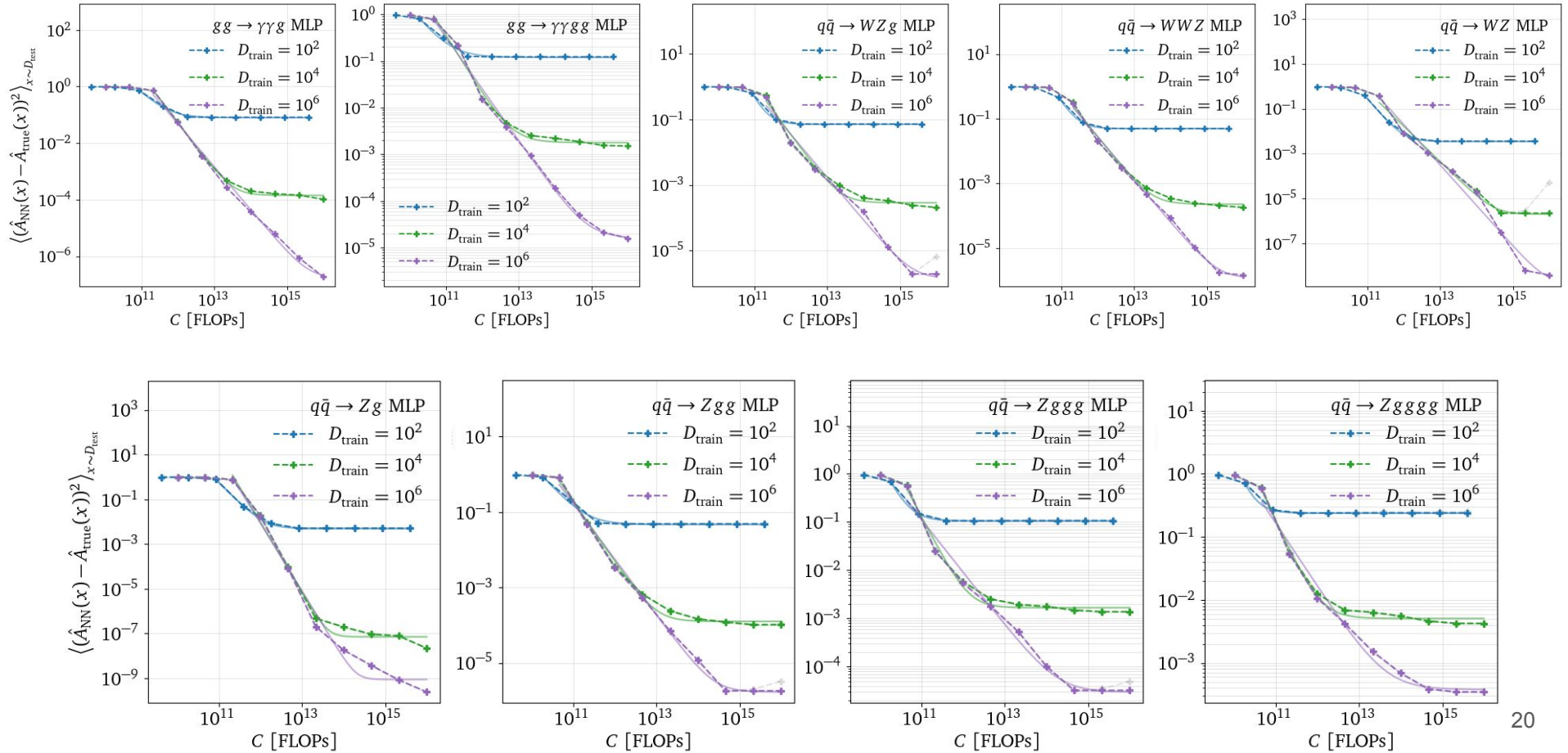


Chebyshev nodes

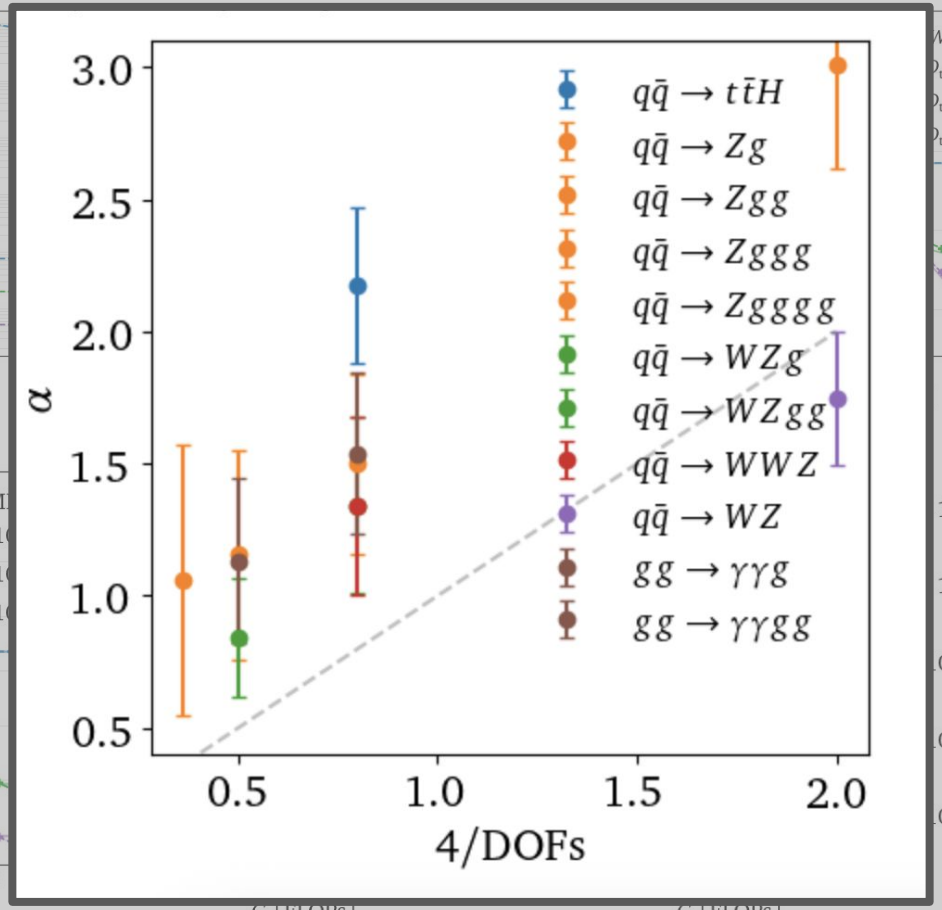


Machine learning addresses all these issues

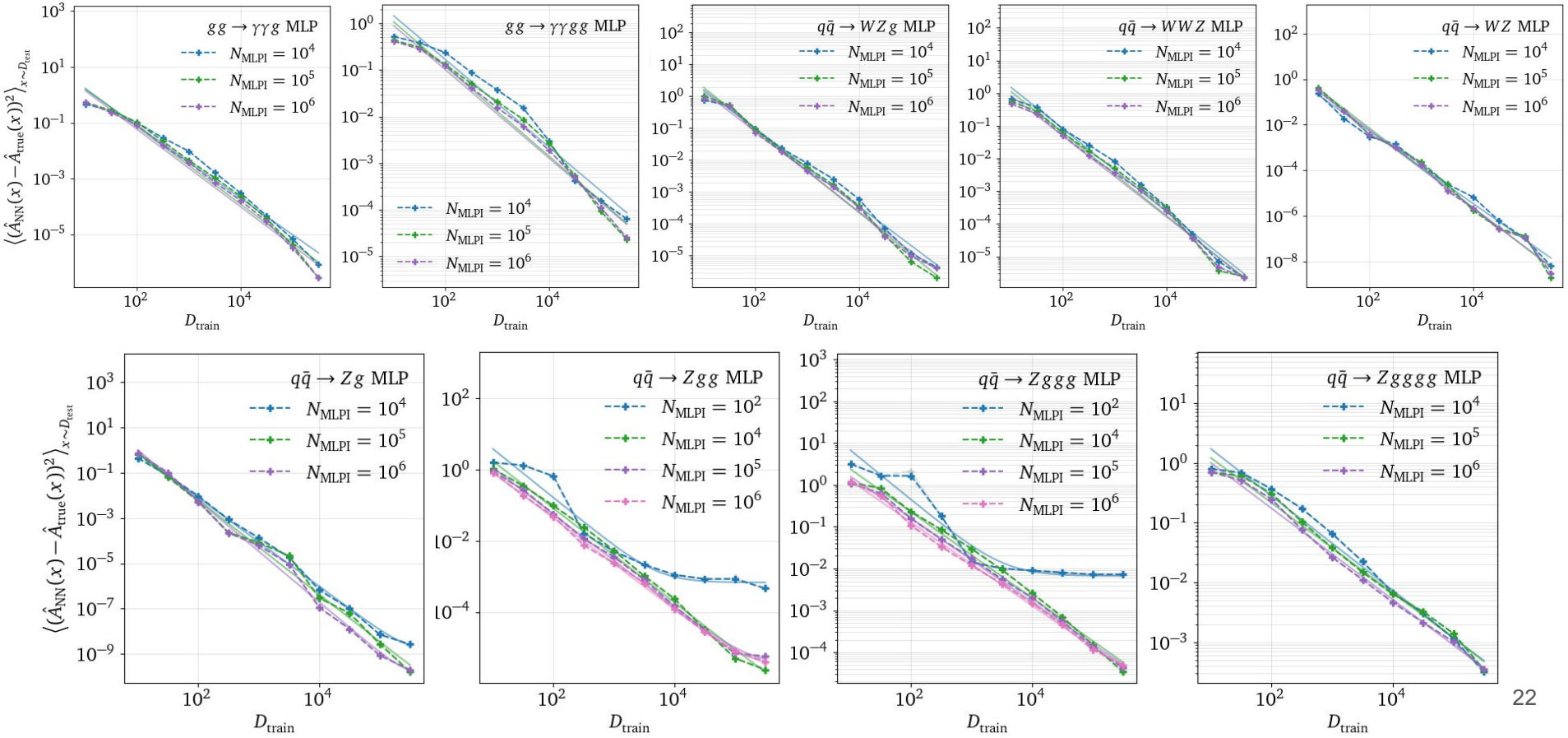
Results scaling on computing resources



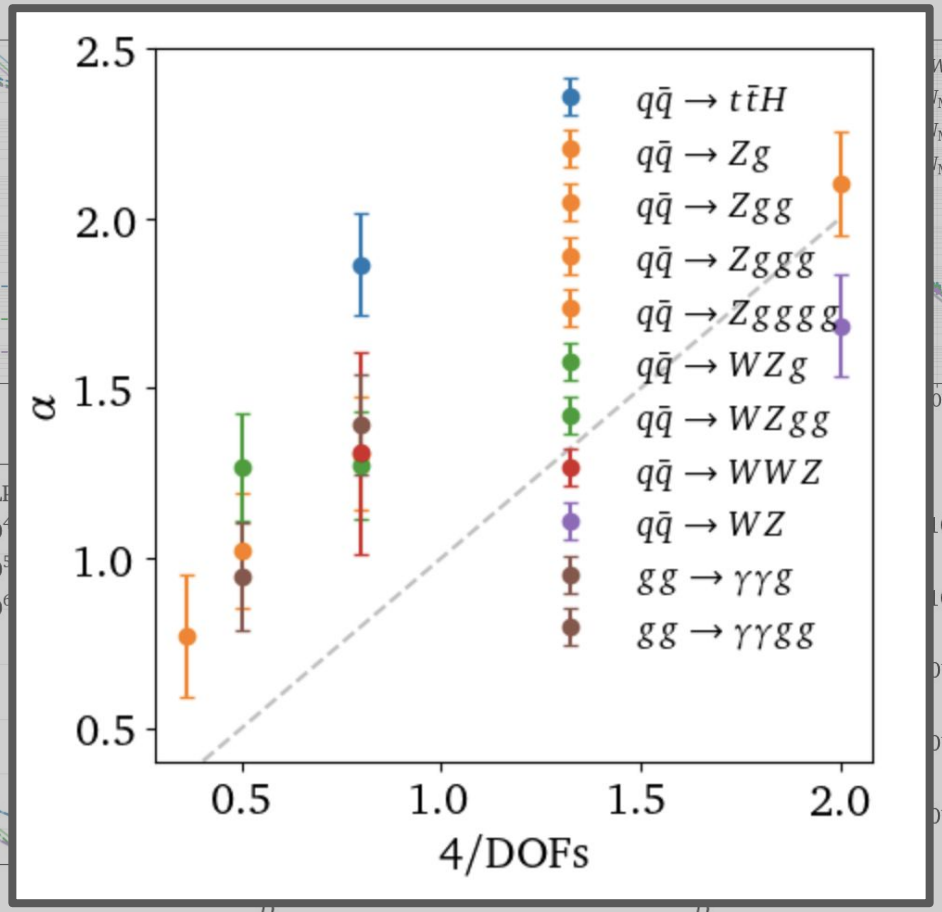
Results scaling on computing resources



Results scaling on dataset size

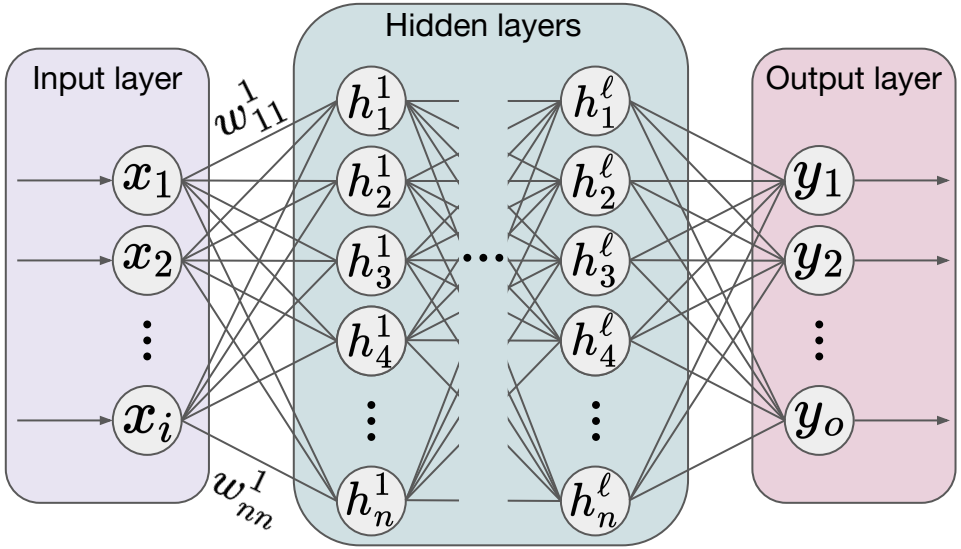


Results scaling on dataset size



Machine learning recap

Example: multilayer perceptron (MLP)



$$h^i_j = a\left(w^{i-1}_{j0} + \sum_{k=1}^{n} w^{i-1}_{jk} h^{i-1}_k\right)$$

With a **non-linear**

Loss function (for ex. Mean Squeared error)

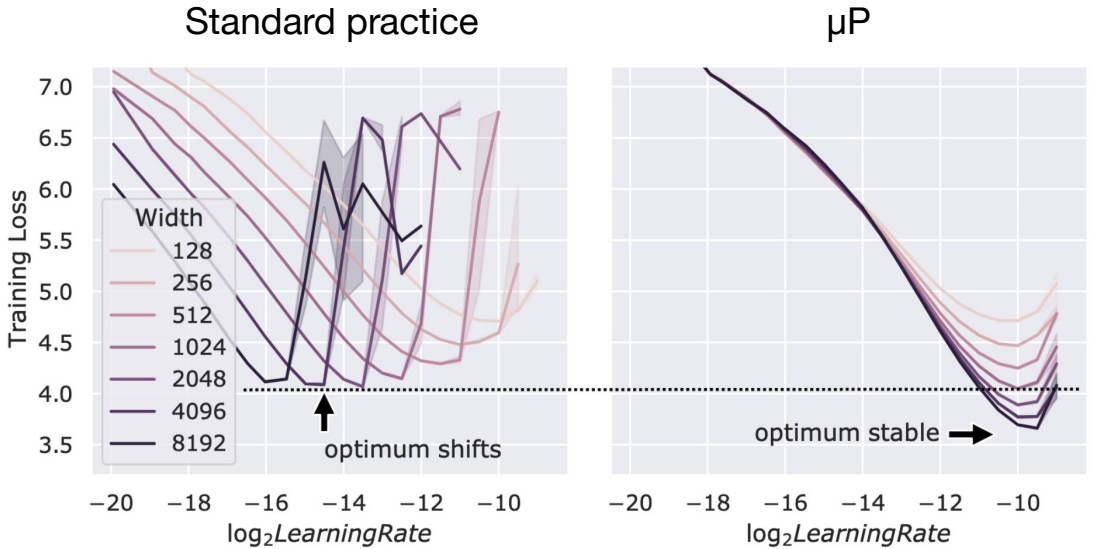
$$\mathcal{L}_{\text{MSE}} = \left\langle (A_{\text{true}}(x) - A_{\text{NN}}(x))^2 \right\rangle_{x \sim D_{\text{train}}}$$

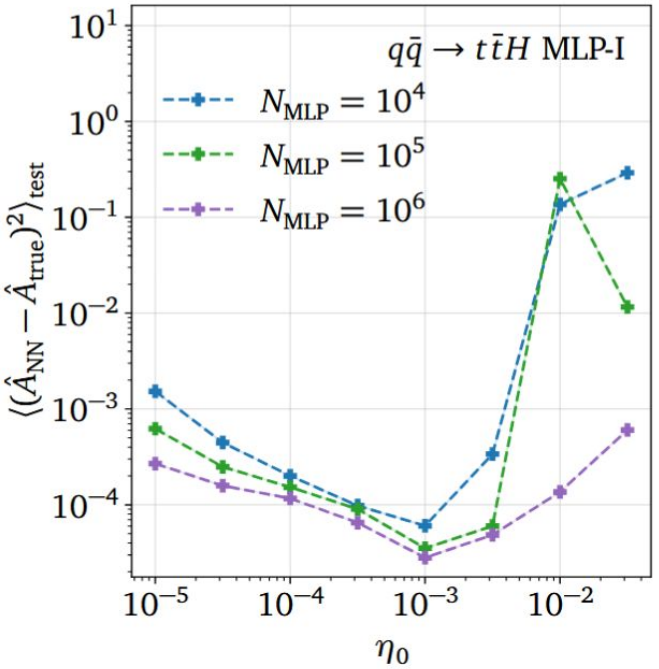
Gradient descent

Weight updates

Finding the right **hyperparameters** for the task at hand remains the central challenge

Hyperparameter Transfer





Allows to optimize HPs on a smaller (cheaper) model **once** and use them for different network sizes

Hyperparameter Transfer[1]

Hyperparameter optimization: the standard problem

Optimizing hyperparameters for large networks is expensive:

- The optimal values **shift** as network width increases
- This forces re-optimization at every new model size
- Large models are costly to train just for tuning

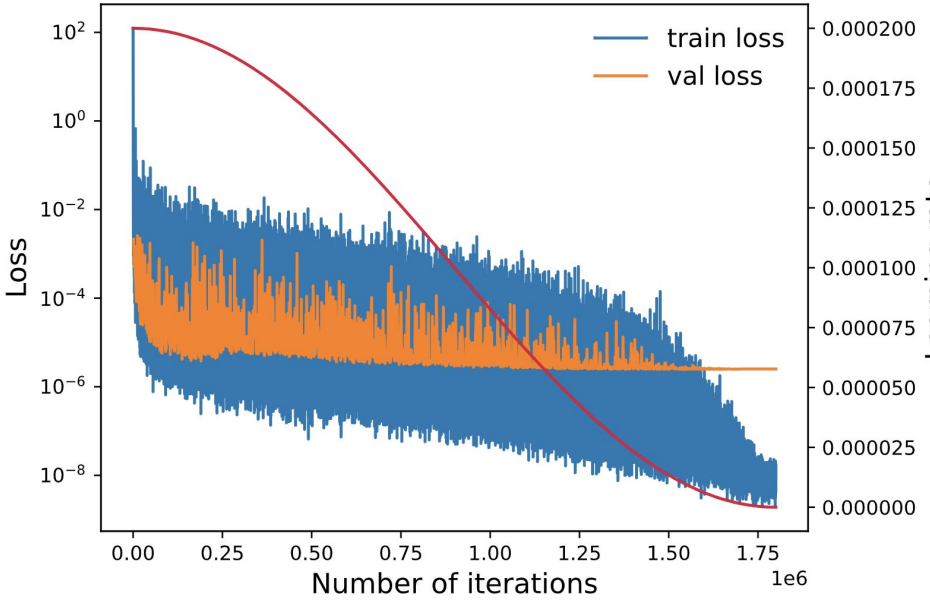
Key insight of μP: instead of the standard initialization, $w \sim \mathcal{N}(0, 1/n)$, treats layers that grow with width and ones that don't differently. Scales learning for hidden and output layers by width as well

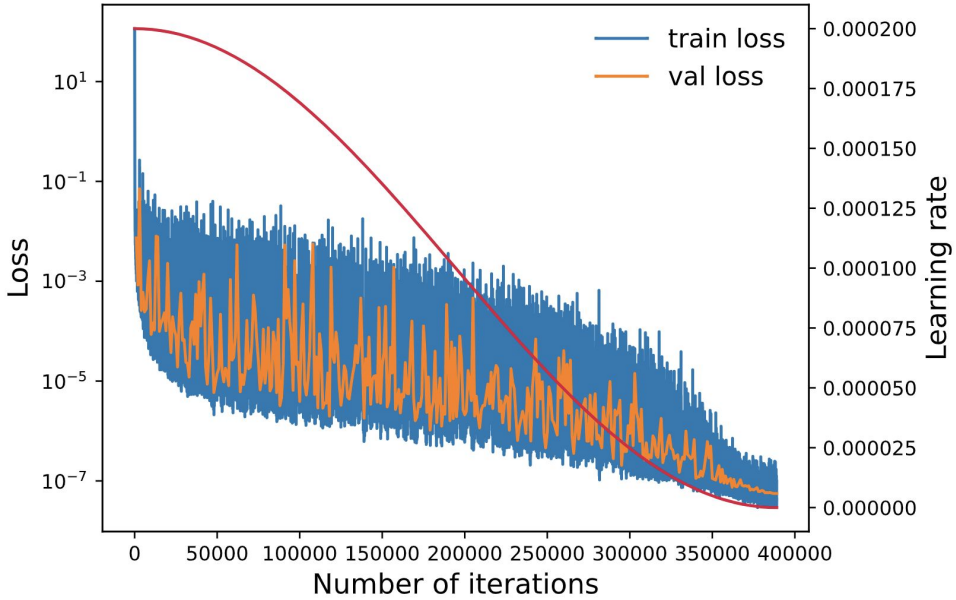
This ensures that the **magnitude of activations and updates remains constant** as width grows, so the loss landscape seen by the optimizer is width-independent

[1] arXiv:2203.03466

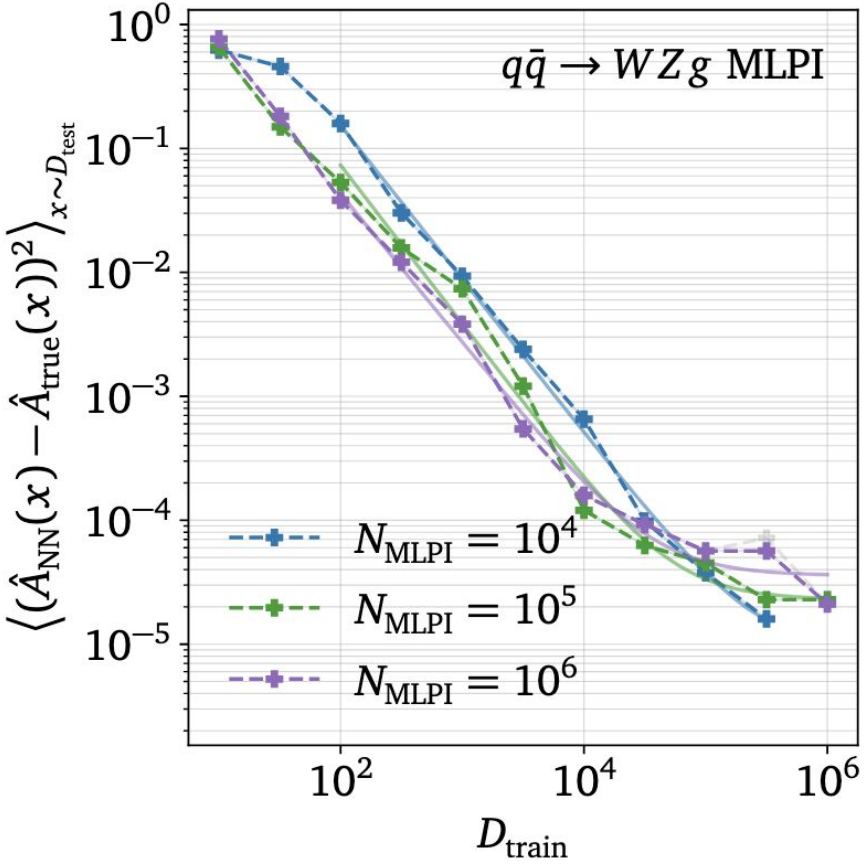
Scheduler

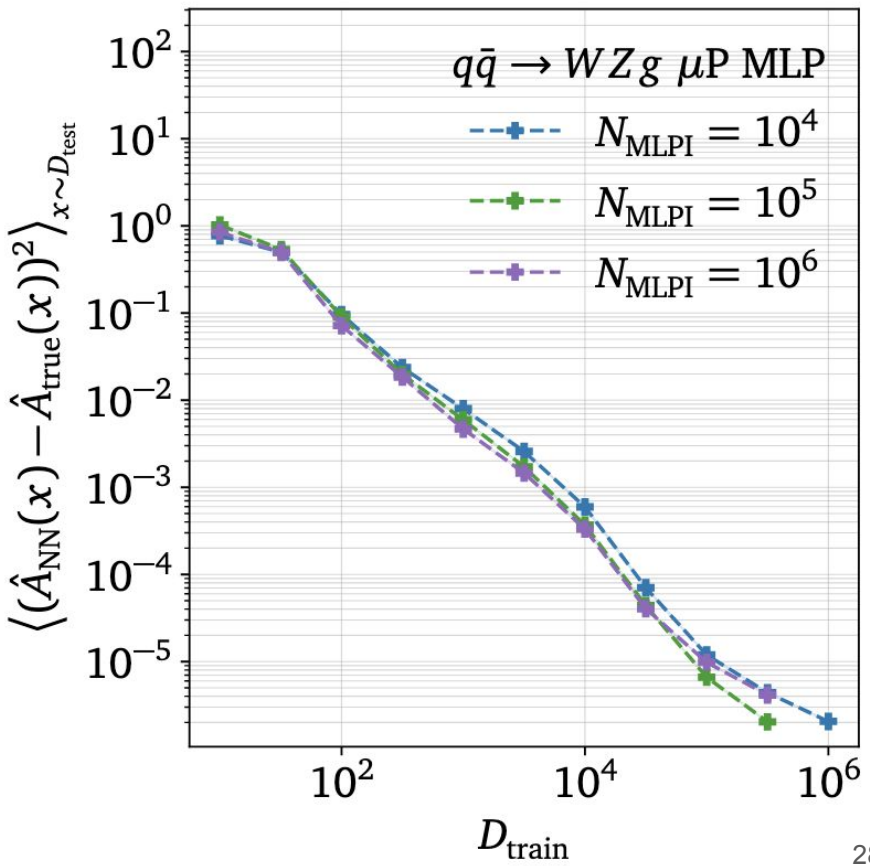
Scheduler has to go to 0 **just at the right time**



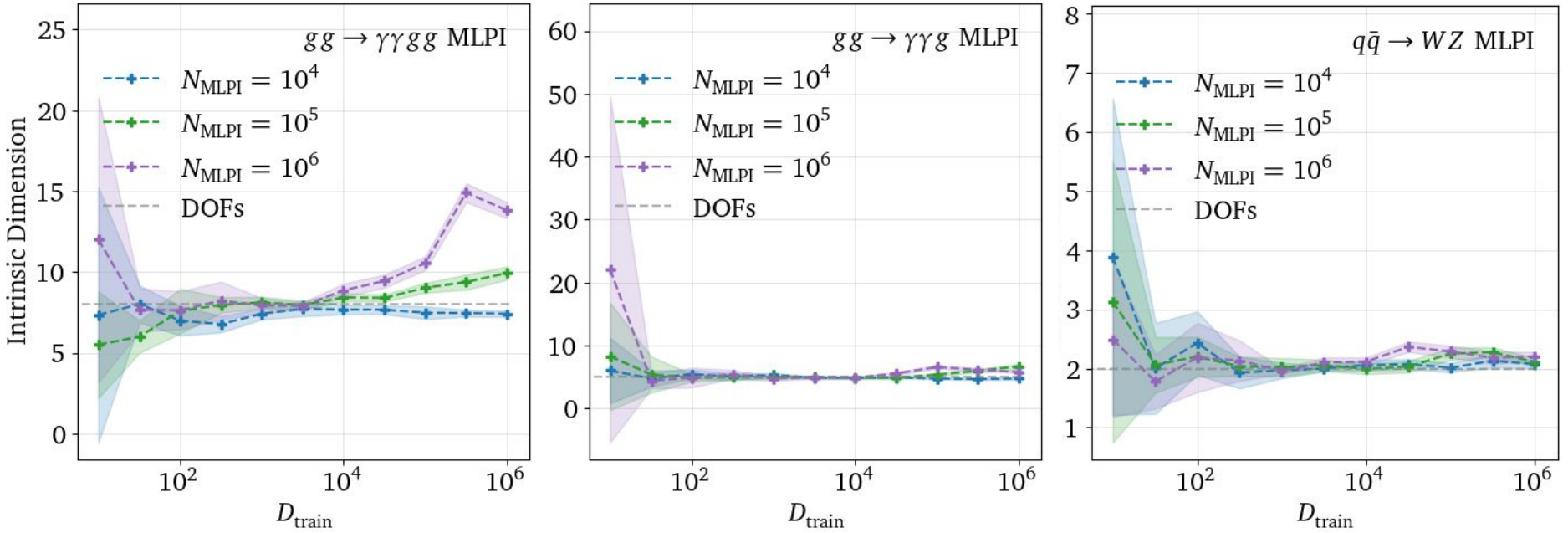


muP vs normal MLP





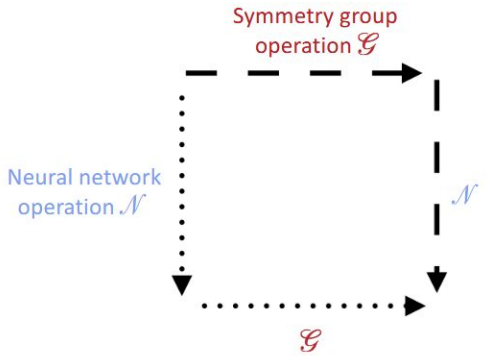
Internal Intrinsic Dimension

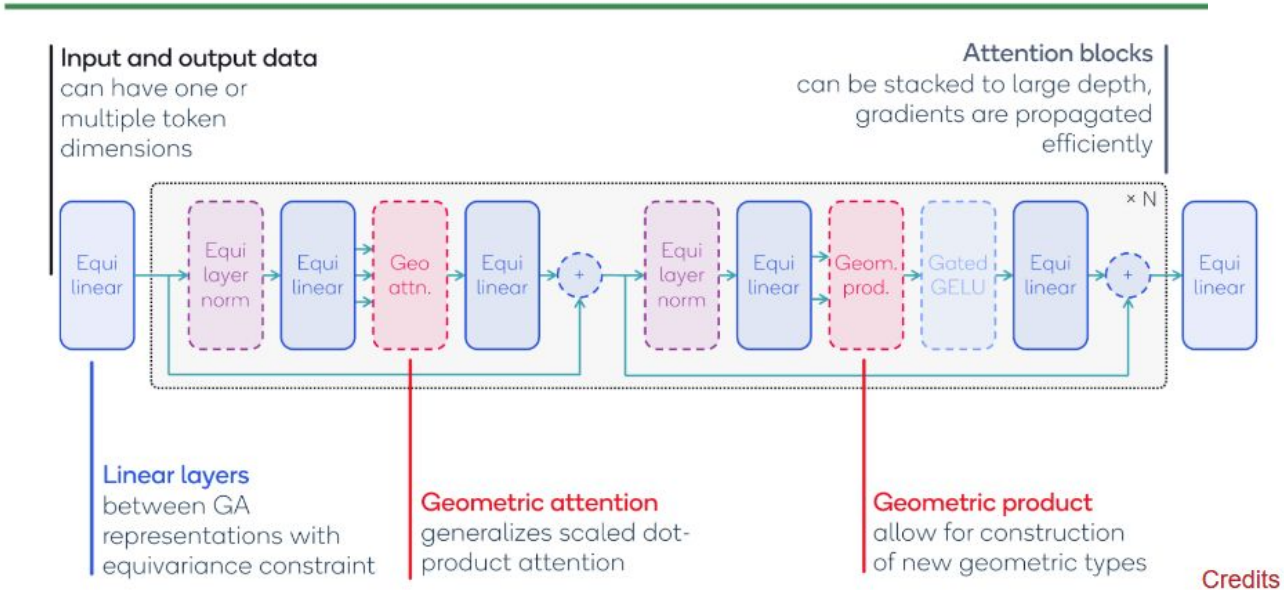


L-GATr

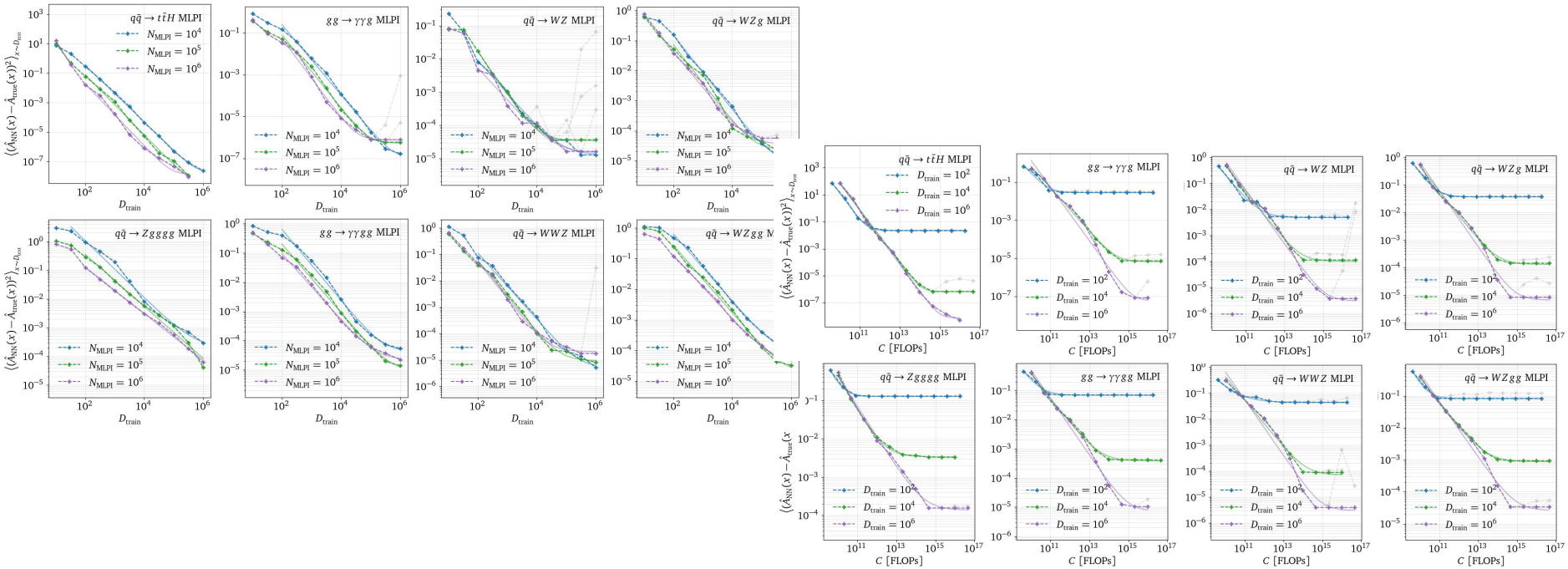
L-GATr = Equivariance + Transformer

$$\mathcal{G}(\mathcal{N}(x)) = \mathcal{N}(\mathcal{G}(x))$$

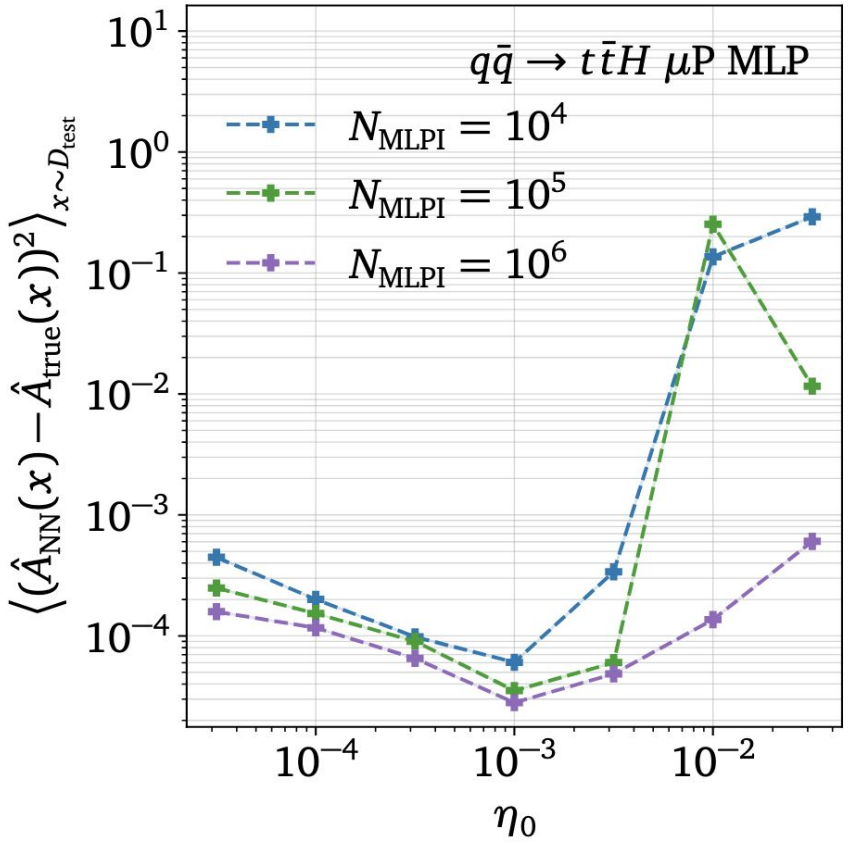




Credits to Johann Brehmer

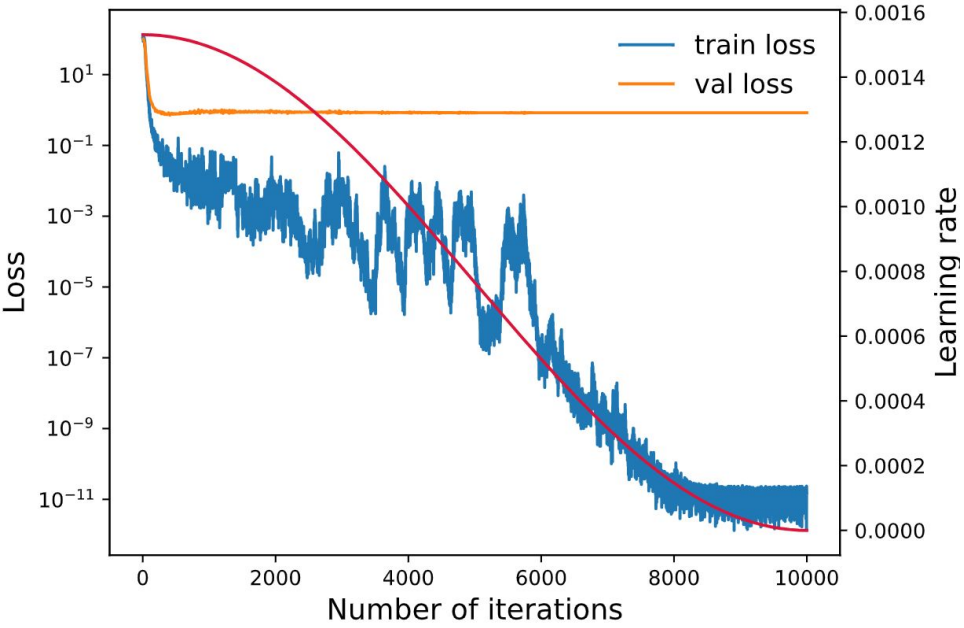

$q\bar{q} \to t\bar{t}H$ MLPI
$gg \to \gamma\gamma g$ MLPI
$q\bar{q} \to WZ$ MLPI
$q\bar{q} \to WZg$ MLPI
$q\bar{q} \to Zgggg$ MLPI
$gg \to \gamma\gamma gg$ MLPI
$q\bar{q} \to WWZ$ MLPI
$q\bar{q} \to WZgg$ MLPI
$N_{\mathrm{MLPI}} = 10^4$
$N_{\mathrm{MLPI}} = 10^5$
$N_{\mathrm{MLPI}} = 10^6$
D_{train}
$\langle(\hat{A}_{\mathrm{NN}}(x) - \hat{A}_{\mathrm{true}}(x))^2\rangle_{x\sim\mathcal{D}_{\mathrm{test}}}$
$D_{\mathrm{train}} = 10^2$
$D_{\mathrm{train}} = 10^4$
$D_{\mathrm{train}} = 10^6$
C [FLOPs]

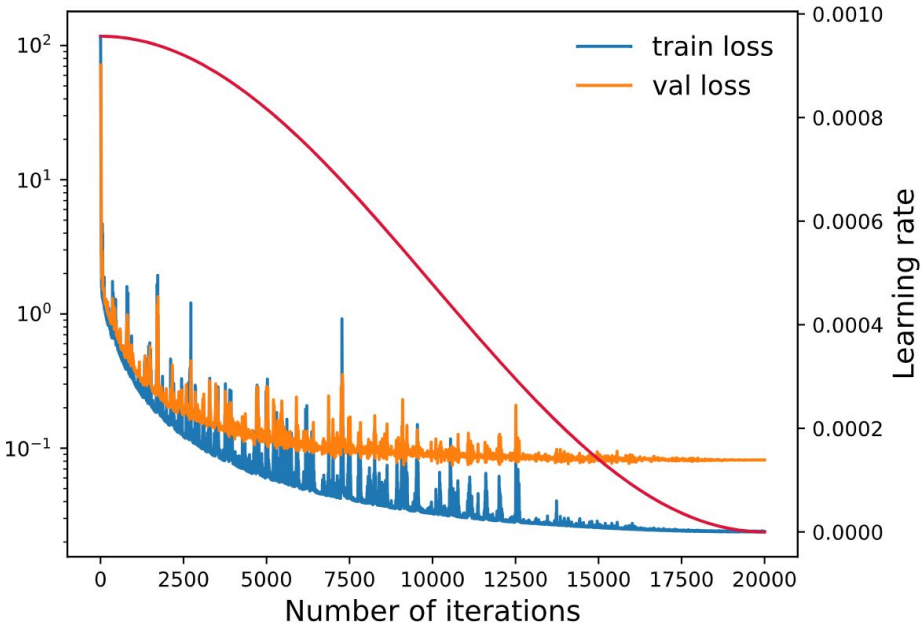
muP


$q\bar{q} \to t\bar{t}H$ μP MLP
$N_{\mathrm{MLPI}} = 10^4$
$N_{\mathrm{MLPI}} = 10^5$
$N_{\mathrm{MLPI}} = 10^6$
$\left\langle (\hat{A}_{\mathrm{NN}}(x) - \hat{A}_{\mathrm{true}}(x))^2 \right\rangle_{x \sim D_{\mathrm{test}}}$
η_0
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}

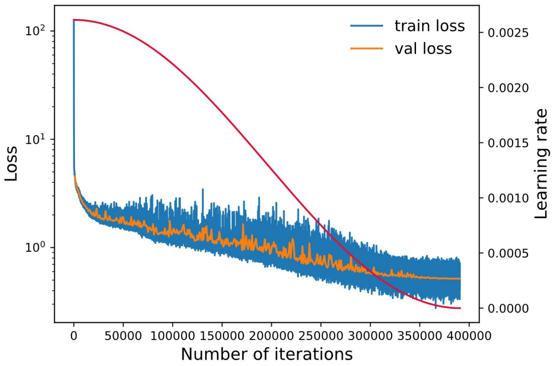
muP overfitting for small datasets

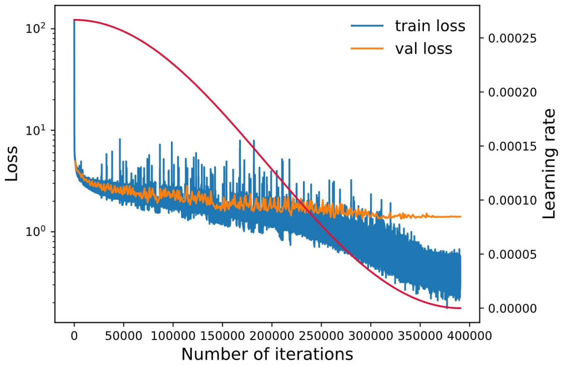
100 training points

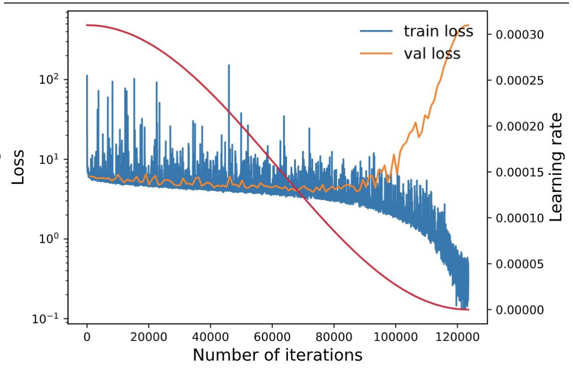


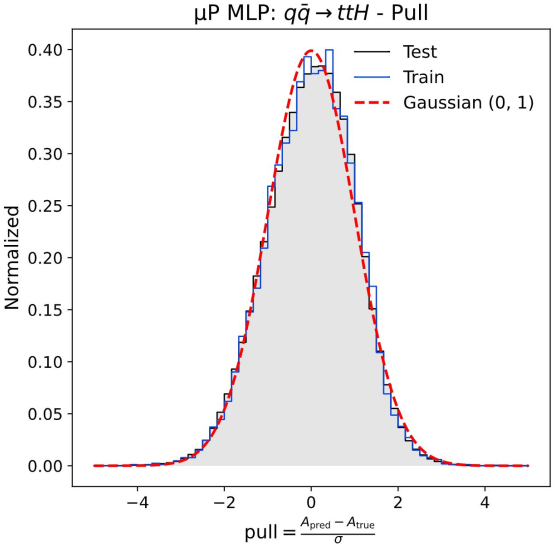


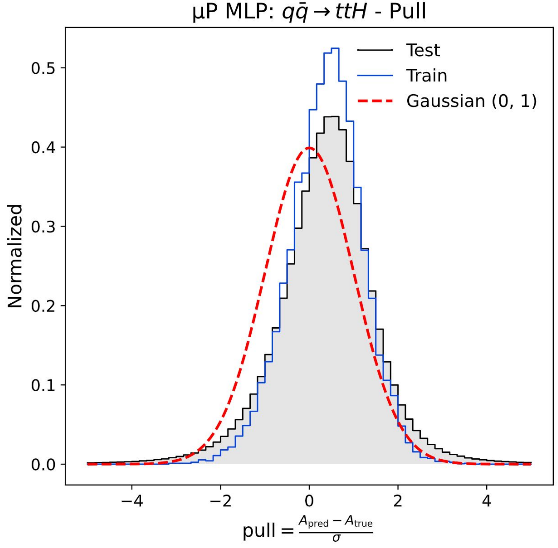
Overfitted Het loss

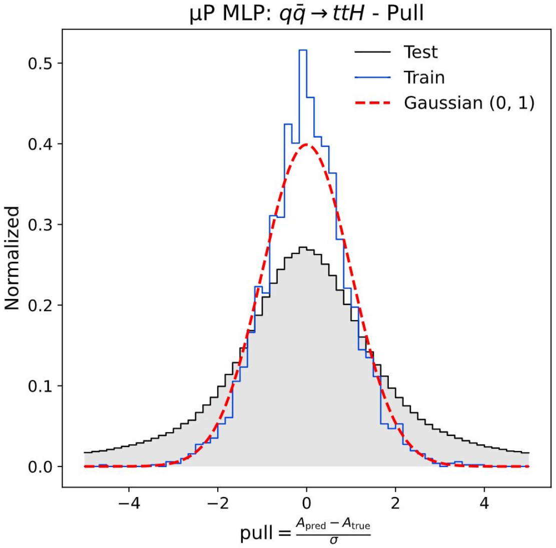












Can we determine scaling laws for scattering amplitude surrogates?

Compare

- Many different processes:

  $$q\bar{q} \to t\bar{t}H,\ q\bar{q} \to Z + ng,\ q\bar{q} \to WZ + ng,\ q\bar{q} \to WWZ,\ gg \to \gamma\gamma + ng$$

- Scalings in D, C and N
- Different loss functions: **MSE** vs **Heteroscedastic Loss** for uncertainties estimation
- Different architectures: **MLP** vs **LLoCa-Transformer**

If scaling laws are universal ⟶ **We predict desired accuracy for given resources**

The physical **degrees of freedom** are known a priori: test relation between scaling and intrinsic dimensionality