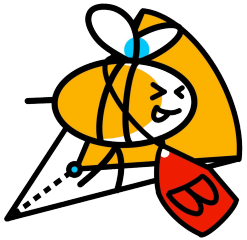
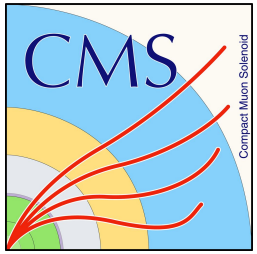


Scaling Laws for the Unified Particle Transformer in Heavy-Flavour Tagging



From model/data scaling to b- and c-tagging performance.

Pavlo Kashko, Alexandre De Moor

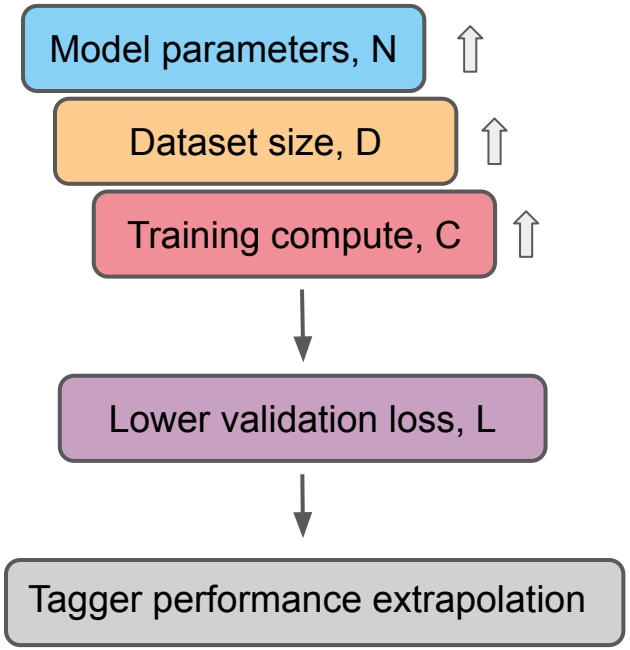
on behalf of the CMS Collaboration

ML4Jets 2026, Vienna

Why scaling laws for jet tagging?

Modern machine-learning systems improve not only through new architectures, but also through scale.

- Scaling laws relate performance to model size N, training data D, and compute C.
- In transformer models, loss L often follows an approximate power-law dependence on these quantities.
- Efficient scaling requires model capacity and training statistics to grow together.



$$L(X) = L_\infty + A \cdot X^{-\alpha}$$

$$X \in \{N, D, C\}$$

Can the same behaviour be observed for detector-level jet flavour tagging?

What do we study?

1. Model scaling
   The capacity of a fixed Particle Transformer model family is varied systematically through its architectural dimensions, providing a controlled scan of the number of trainable parameters.

2. Data scaling
   The number of unique jets used for training is varied up to the full 301 million-jet reweighted dataset.

3. Compute scaling
   Training, validation, and inference performance are studied as functions of the total training compute.

4. Physics-performance scaling
   The evolution of the classification loss is related directly to b-jet, c-jet, and quark/gluon discrimination, using background rejection at fixed signal efficiency as the physics-performance metric.

The goal is to understand how additional ML resources translate into physics performance.

Dataset construction

The full training pool contains approximately 543 million jets.

- Each jet receives a pT-η-dependent sampling weight relative to the b-jet reference distribution.
- Jets are sampled stochastically according to these weights during training.
- The weighted sample corresponds to approximately 301 million effective jets per epoch.

The underlying jet pool is preserved; only the sampling probability changes.

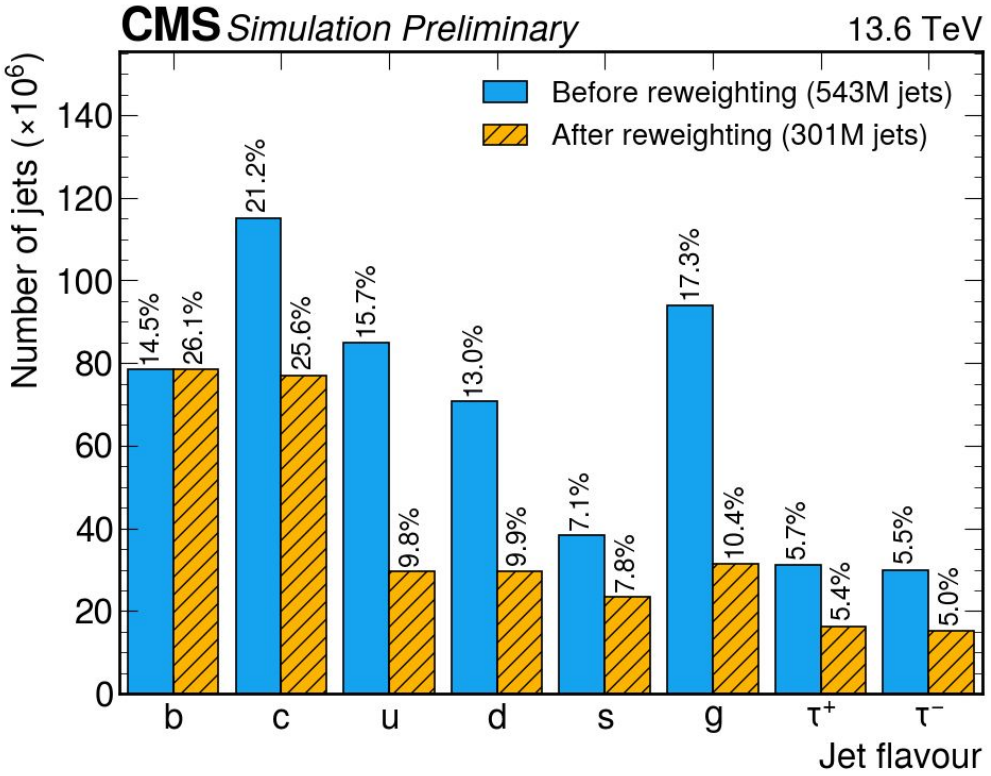


Figure 1: Flavour composition of the jet sample used for training, before (blue) and after (orange, hatched) the pT-η shape reweighting.

Kinematic coverage after reweighting

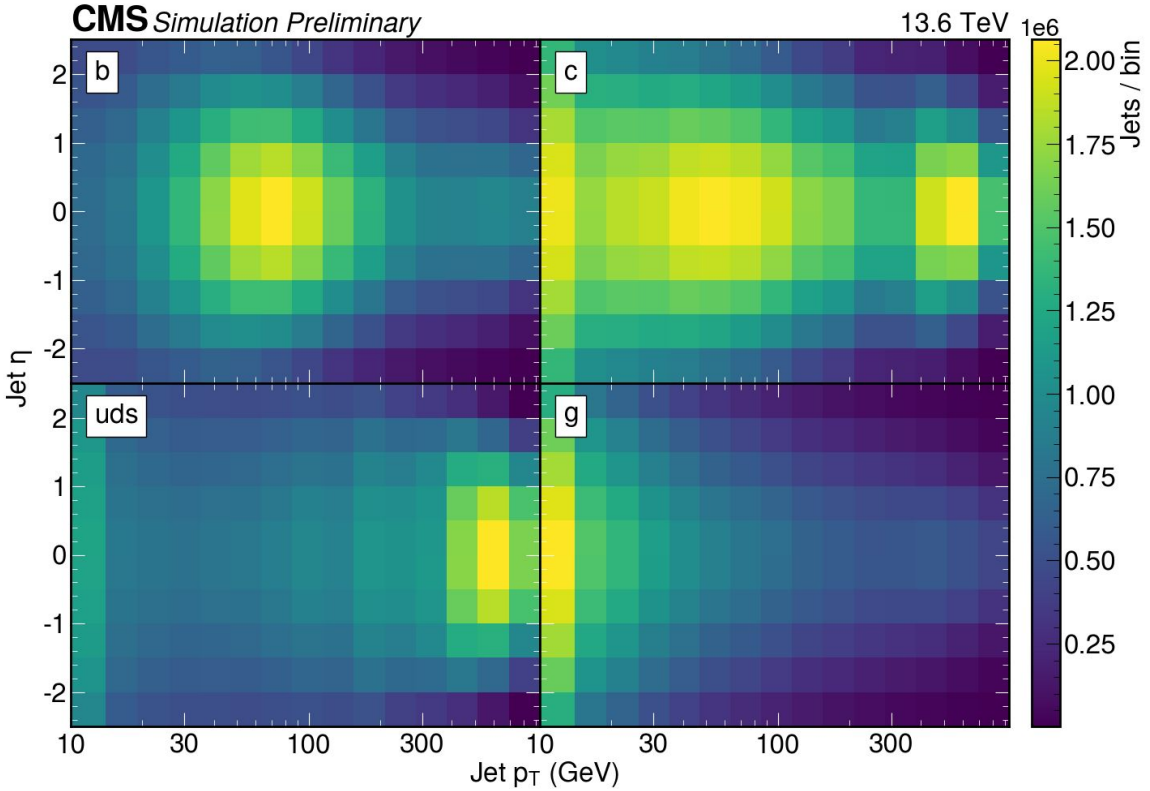


Figure 2: Number of simulated training jets per pT-η bin for b (upper left), c (upper right), uds (lower left), and gluon (lower right) jets.

- Reweighting equalizes the the pT-η shapes across flavors while preserving the broad kinematic phase space of the original samples.
- The scaling study therefore changes the number of training examples without changing the kinematic features learned by different model sizes.

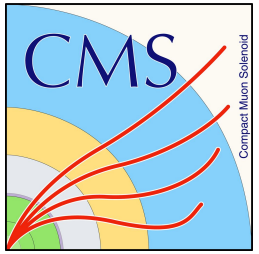
Model definition and hyperparameters

All models use the same Particle Transformer [4,7] architecture family.

- Capacity is varied through the embedding dimension and number of encoder blocks.
- Inputs, flavour labels, loss function, optimizer, and evaluation sample are kept fixed.
- Common training hyperparameters are adopted from previous CMS Particle Transformer optimization.
- All trainings were done using CMS Machine Learning framework b-hive [8, Monday talk by Ulrich].

Only scale changes; the tagging task remains fixed.

Hyperparameter	Value
Input blocks	Charged, neutral particle flow candidates, secondary vertices, V^0 candidates, lost tracks
Learning rate	0.001
Optimizer	AdamW
Learning rate scheduler	cosine annealing
Loss function	cross-entropy
Batch size	2048
Number of epochs	20



Scaling of the loss

Empirical model/data balance

Increasing model size or dataset size independently improves performance, but the two must remain reasonably balanced.

- Small models underuse large datasets.
- Large models become inefficient when trained with insufficient statistics.

A broad region of low validation loss is observed for

$$\frac{N}{D} \sim 5\text{-}10\%$$

The best configurations increase model capacity and training statistics together.

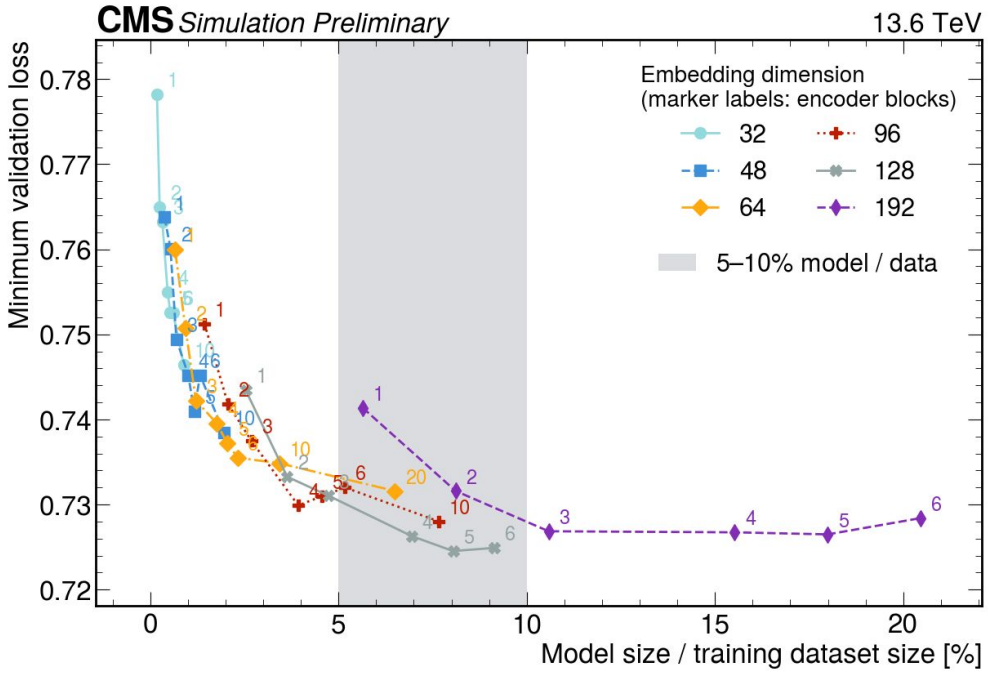


Figure 3: Minimum validation loss as a function of ratio (trainable parameters / number of training jets, in %).

Scaling model capacity: width and depth

Model capacity can be increased through both the transformer width and depth. At fixed training conditions, increasing either axis lowers the minimum validation loss. The best-performing region is reached when both dimensions are increased, indicating that the observed scaling is not associated with a single architectural parameter.

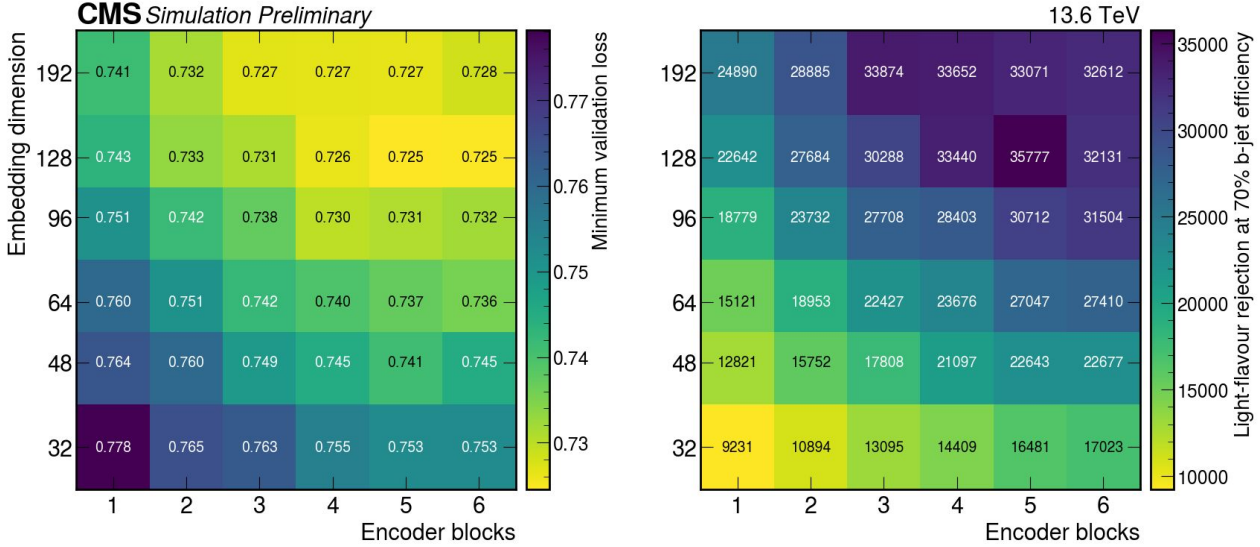
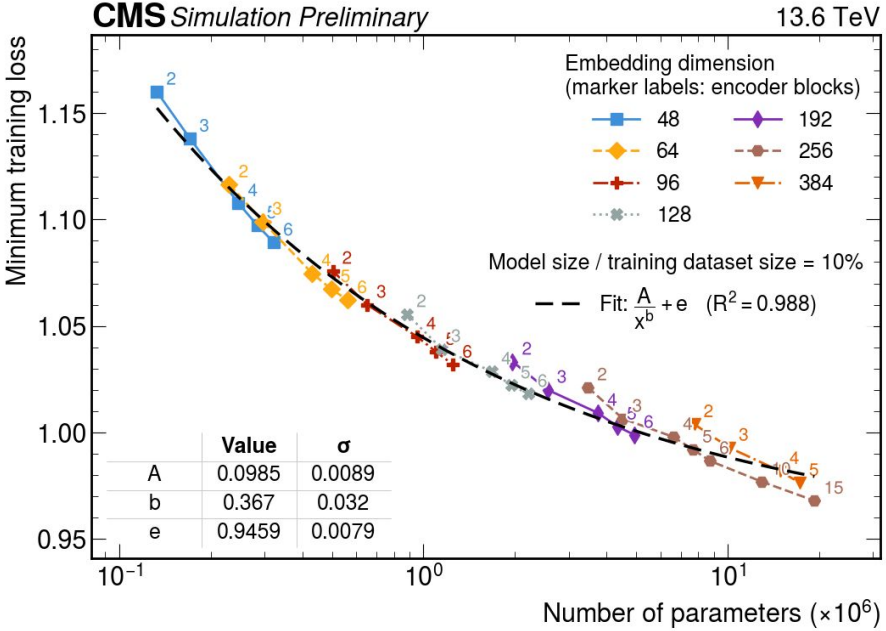


Figure 4: Dependence of the minimum validation loss and b-versus-light rejection on transformer depth and embedding dimension.

Joint model-data scaling at fixed N/D

Training and validation losses decrease smoothly across more than two orders of magnitude in model size. Both are well described by the power-law fit function.



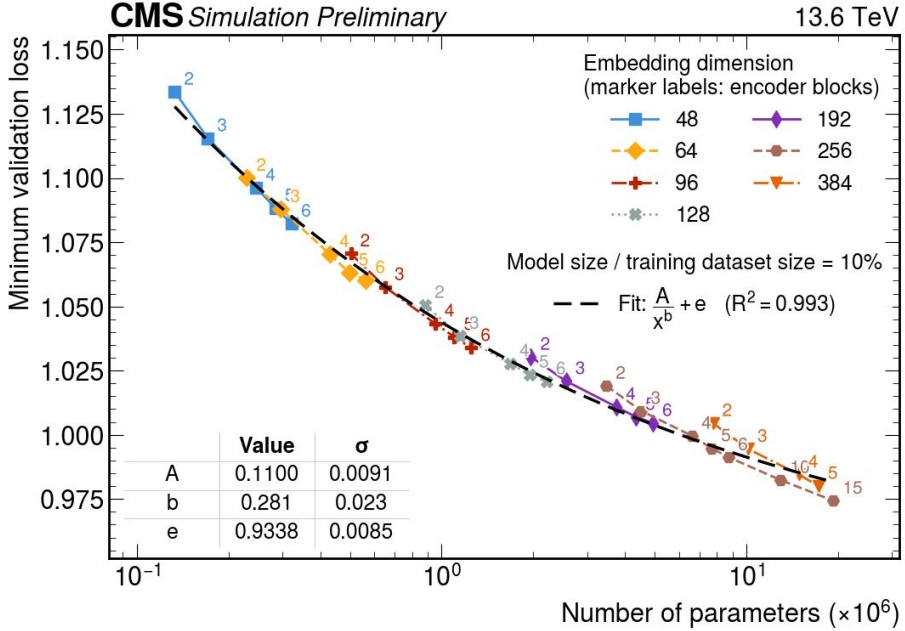


Figure 5: Minimum training loss (left) and validation loss (right) as functions of the number of trainable model parameters N.

Compute is the common currency

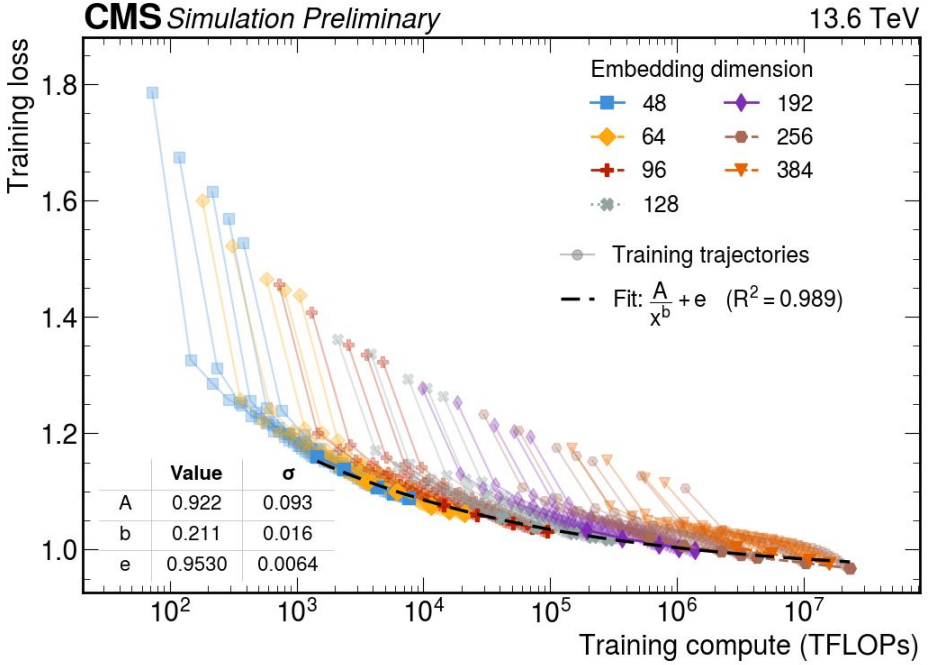


Figure 6: Training loss as a function of the cumulative training compute for the studied model configurations and training epochs. The faint trajectories show the evolution of individual models during training. The dashed curve represents a fit of the smallest loss.

Different architectures can be compared using the total computational cost of training.

- We define C as a cumulative floating-point operations up to a training checkpoint
- Loss decreases smoothly over several orders of magnitude in C.
- Models with different widths and depths populate the same overall compute-scaling trend.

The dependence of the loss is described by

$$L(C) = L_\infty + A \cdot C^{-\alpha}$$

Training compute provides a common scaling variable across architectures.

Scaling is flavour dependent

The inclusive loss hides substantially different behaviour among jet flavours.

- b jets: clear reduction in loss with increasing scale.
- c jets: improvement continues, but with a smaller scaling exponent.
- uds jets: a strong decrease of the remaining reducible loss.
- gluon jets: no clear monotonic scaling behaviour is resolved in the explored range.

Different classification tasks **benefit differently** from additional scale. The exponent determines how quickly the reducible component decreases; the absolute gain also depends on its amplitude.

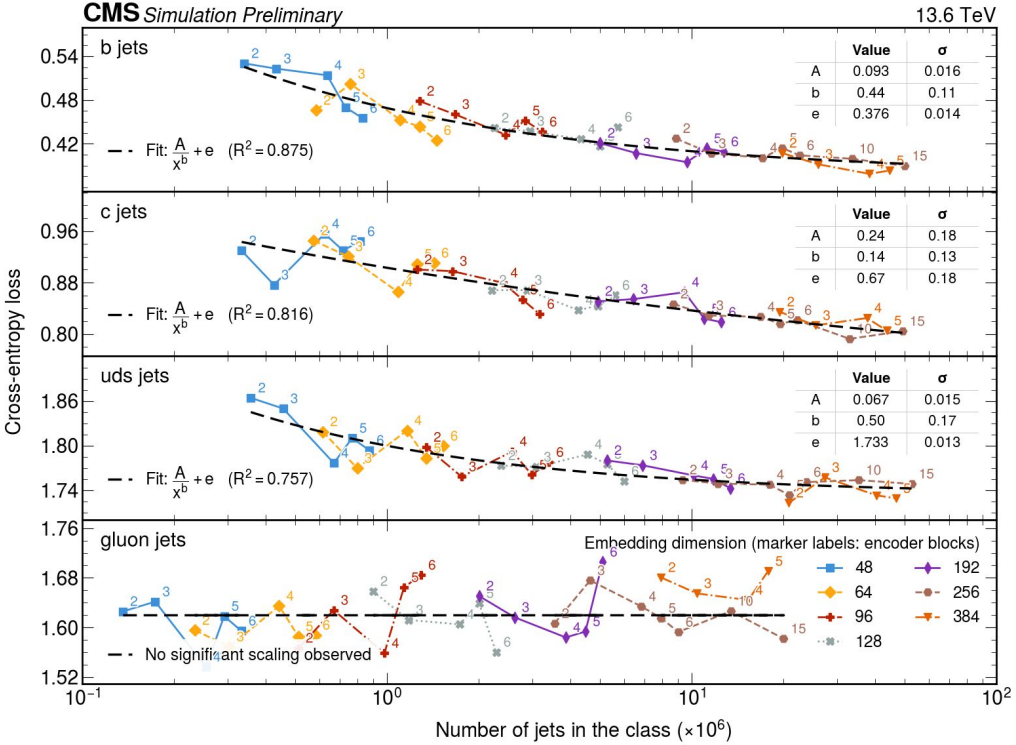
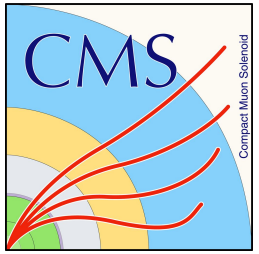


Figure 7: Cross-entropy loss for b, c, uds, and gluon jets as a function of the number of training jets in the corresponding flavour class.



Scaling of tagging performance

Does lower loss mean better tagging?

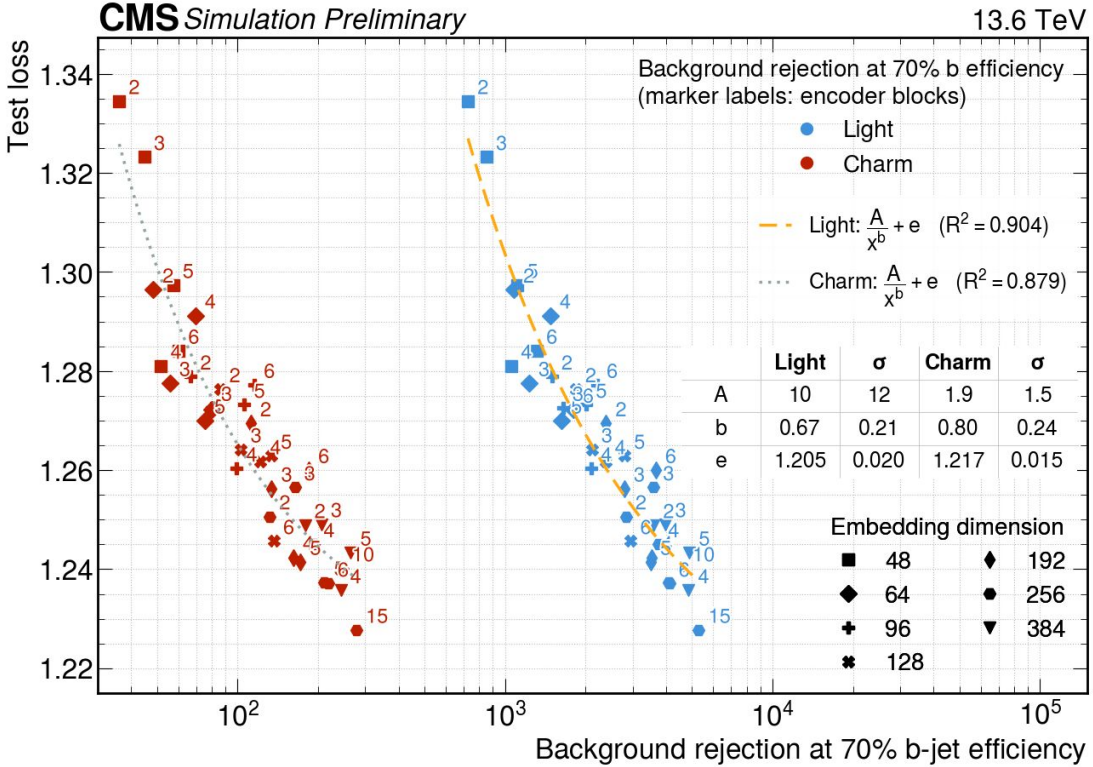


Figure 8: Test loss versus background rejection at 70% b-jet efficiency for models trained with 10% ratio of model size to training-dataset size.

- Improvements in the classification loss translate into improved flavour-tagging performance.
- For b-jet identification, lower inference loss is associated with higher background rejection at fixed signal efficiency.
- This relation provides the connection between the metric and the physics quantity relevant for CMS analyses.

b tagging scales with compute (overview)

- b-tagging performance improves continuously with increasing training compute over more than four orders of magnitude.
- Light- and charm-jet rejection follow similar power-law trends, with fitted exponents of approximately 0.15.
- Models with different embedding dimensions populate the same overall scaling trend, indicating that training compute provides a useful common scaling variable across the studied architectures.
- The observed behaviour shows no clear saturation within the explored compute range.

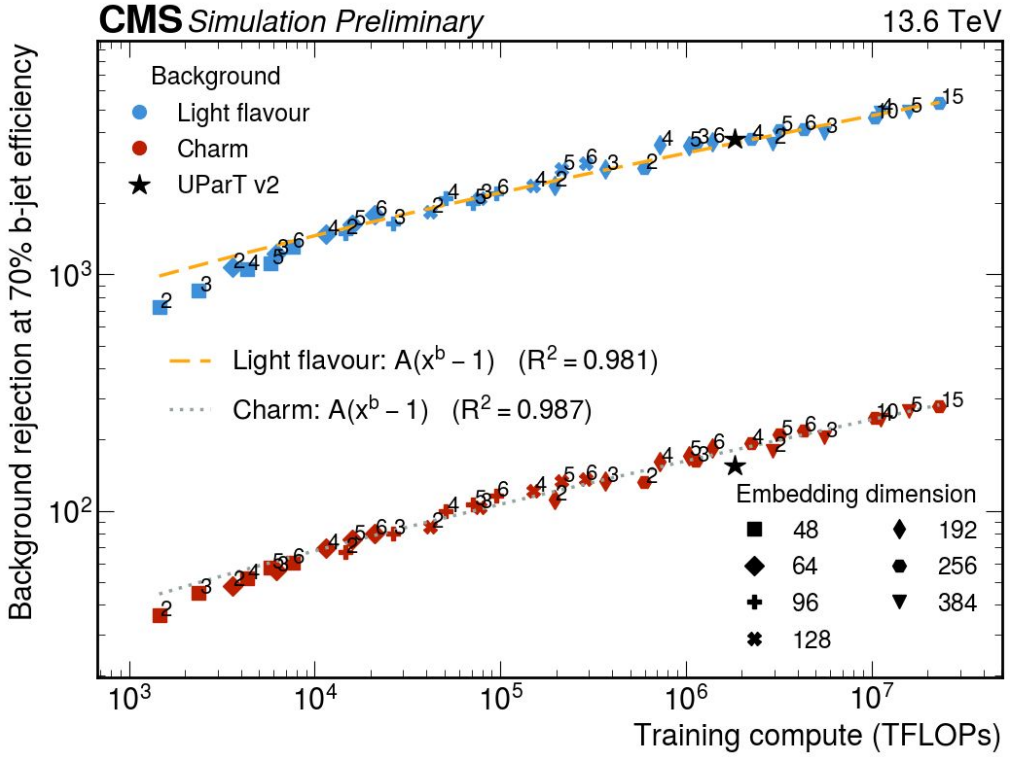
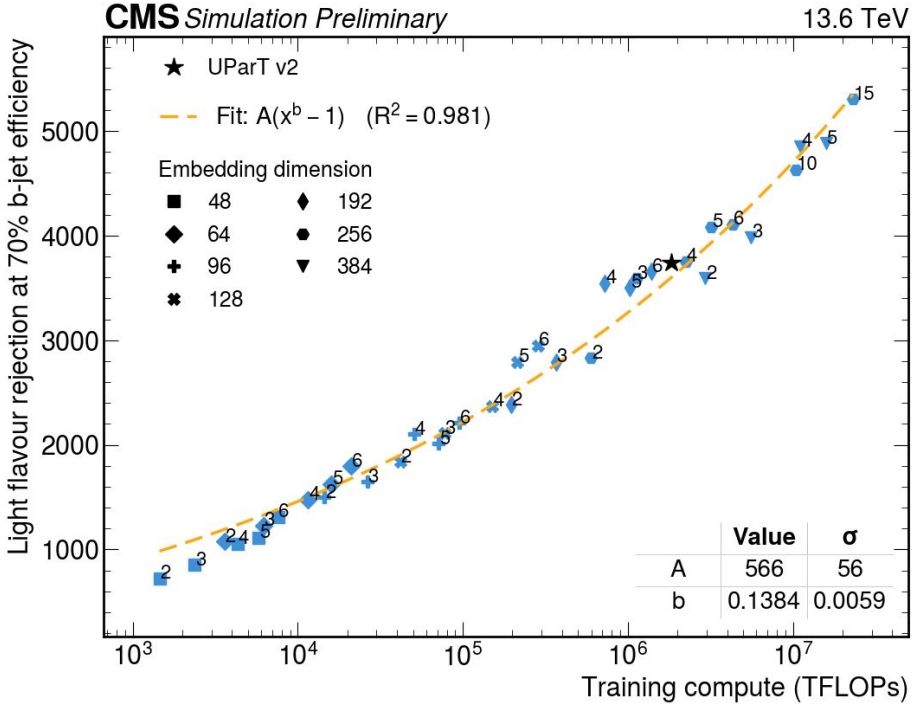


Figure 9: Background rejection for light-flavour and charm jets at a fixed b-jet efficiency of 70% as a function of the total training compute.

b tagging scales with compute (linear scale)



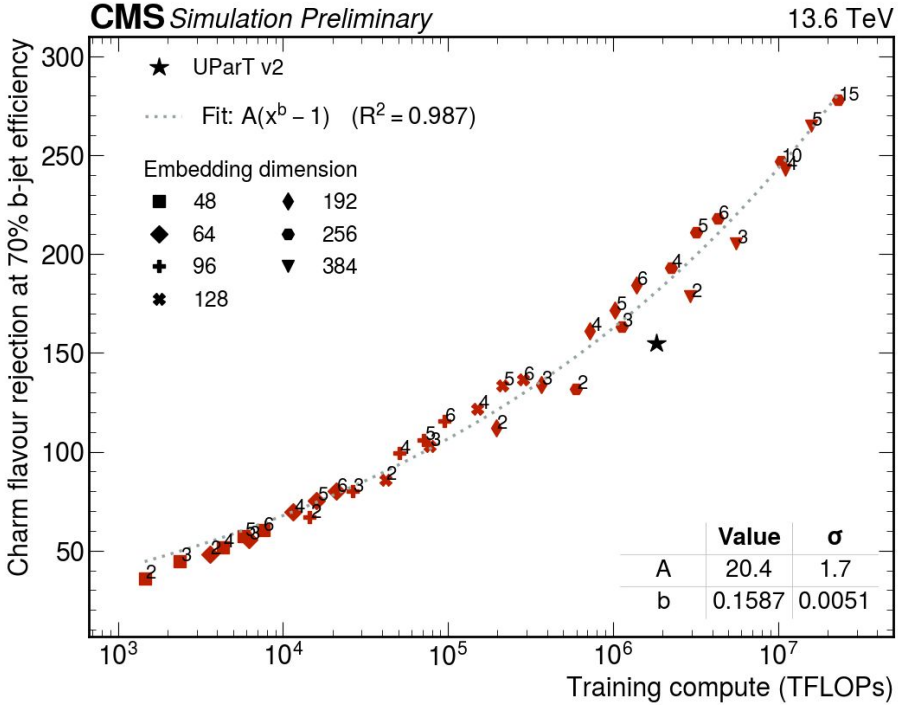
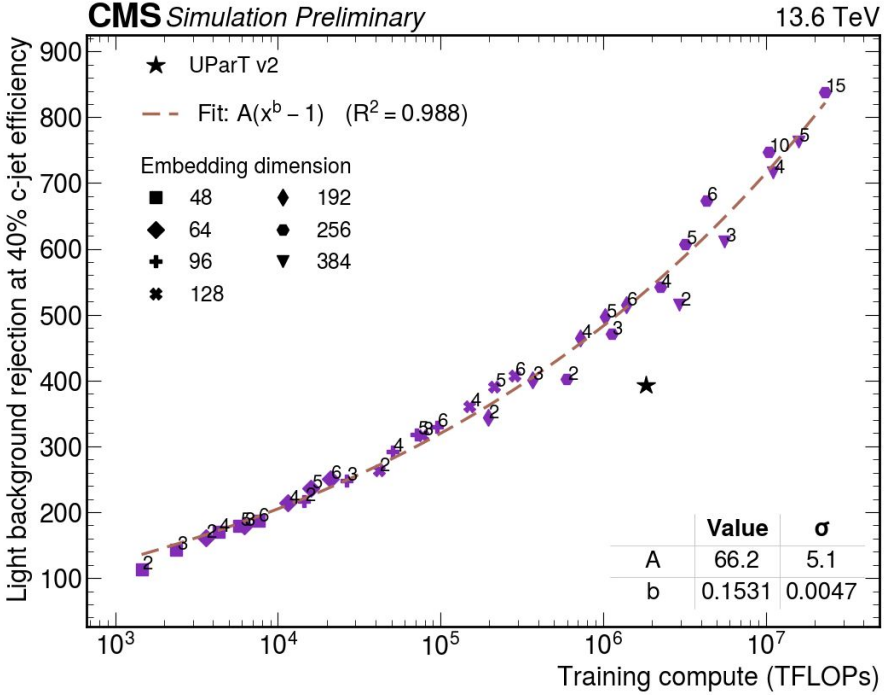


Figure 10: Light-flavour (left) and charm-jet (right) background rejection at 70% b-jet efficiency as a function of training compute. The performance follows an approximate power-law dependence across the studied model configurations; UParT v2 is shown for reference.

c tagging and quark/gluon tagging



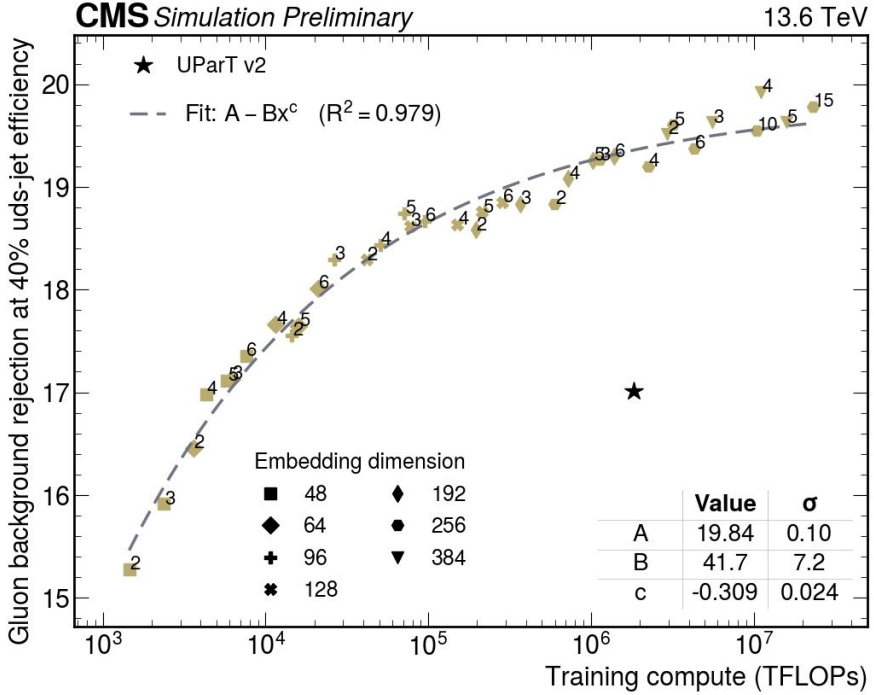


Figure 11: Background rejection as a function of training compute for c-jet tagging (left) and quark/gluon discrimination (right). The c-tagging performance shows a clear power-law improvement with increasing compute, while quark/gluon discrimination exhibits a weaker trend approaching saturation. UParT v2 is shown for reference.

Scaling is powerful – and expensive

- The computational cost increases rapidly as model capacity and training statistics are scaled together.
- Over the explored range, the total training time follows approximately

$$t_{\text{train}} \sim N^{1.12}$$

  with *N* the number of trainable parameters.
- The largest configurations require orders of magnitude more compute than the smallest models, emphasizing the increasing resource cost of further performance gains.
- The whole study was done using a single GPU H100 with all models taking ~380h of pure calculations.

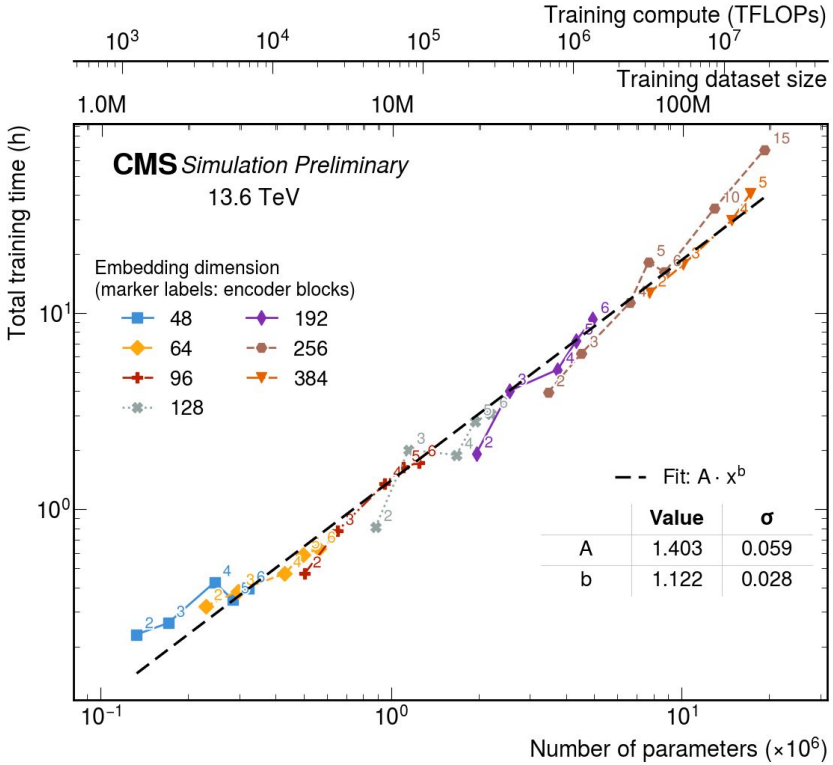


Figure 12: Total training time as a function of the number of model parameters. The corresponding training-dataset size and total training compute are shown on the upper axes.

Takeaways

Scaling laws appear in detector-level jet flavour tagging

1. Model and data should scale together.
   A broad low-loss region appears when model capacity is matched to available training statistics.

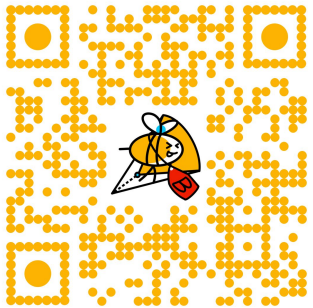
2. Loss follows smooth empirical scaling relations.
   Model size, dataset size, and compute provide quantitative scaling variables.

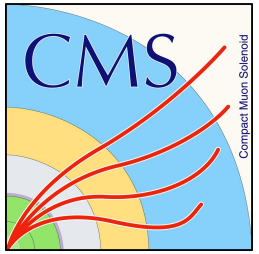
3. Better loss means better physics performance.
   b- and c-tagging background rejection increase systematically with scale.

4. Scaling is task dependent.
   Heavy-flavour tagging continues to improve, while quark/gluon discrimination shows much weaker gains.

5. Compute becomes the limiting resource.
   Further improvements remain possible, but at rapidly increasing computational cost.

Scaling gives us a quantitative way to decide
how large the next generation of CMS jet taggers should be.





Back Up

References

[1] Jared Kaplan et al., “Scaling Laws for Neural Language Models”, 2020, arXiv:2001.08361.
[2] Jordan Hoffmann et al., “Training Compute-Optimal Large Language Models”, 2022, arxXv:2203.15556.
[3] Joshua Batson and Yonatan Kahn, “Scaling Laws in Jet Classification”, 2023, arXiv:2312.02264.
[4] CMS Collaboration, “Flavour tagging performance of the updated Unified Particle Transformer algorithm with the CMS experiment at √s=13.6 TeV”, CMS Detector Performance Summary CMS-DP-2025-081.
[5] ATLAS Collaboration, “Carpe Datum: Scaling behavior of transformers for heavy hadron flavor identification”, 2026, ATL-SOFT-PUB-2026-002.
[6] Matthias Vigl et al., “Neural Scaling Laws for Boosted Jet Tagging”, 2026, arXiv:2602.15781.
[7] H. Qu, C. Li, S. Qian, “Particle Transformer for Jet Tagging,” PMLR 162:18281-18292, 2022.
[8] CMS Collaboration, “b-hive: a modular training framework for state-of-the-art object-tagging within the Python ecosystem at the CMS experiment”, CMS Detector Performance Summary CMS-DP-2024-020.

Glossary

- **Particle-level jet**: A jet clustered from stable ($c\tau > 1$cm) particles, excluding neutrinos, before any detector simulation.
- **Reconstruction-level jet**: A jet clustered from reconstructed particle flow (PF) candidate [1].
- **Small-cone jets (AK4 jets)**: Jets are clustered from PF candidates using the anti-kT algorithm [2] with parameter R= 0.4. The Pileup Per Particle Identification (**PUPPI**) algorithm [3,4] is used for pileup (PU) mitigation.
- **Heavy-flavour jets**: reconstruction-level jets matched with a particle-level jet associated with a b/c ghost hadron
- **Light-flavour jets**: reconstruction-level jets matched with a particle-level jet associated with a light parton. The flavour is defined by the hardest parton associated.
- **Tau jets**: Jets originating from tau leptons decaying to hadrons
- **Secondary Vertex (SVs):** The point from where the b or c hadron decays. The vertex reconstruction is performed using the adaptive vertex fitter (AVF) and inclusive vertex finding (IVF) algorithm [5].
- **UParT v1/2:** A ParticleTransformer [8] model designed for AK4 jet tasks, performing in an inclusive way heavy flavour and hadronic τ identification [10] combined with a flavour-aware jet energy regression and jet energy resolution estimation [11]. It introduces pairwise "interaction" features between jet constituents and SVs, enhancing internal jet relations.

Introduction

Neural scaling laws describe empirical relationships between machine-learning performance and the resources used for training, including model capacity, dataset size, and computational cost. Such behaviour has been extensively studied for large language models.

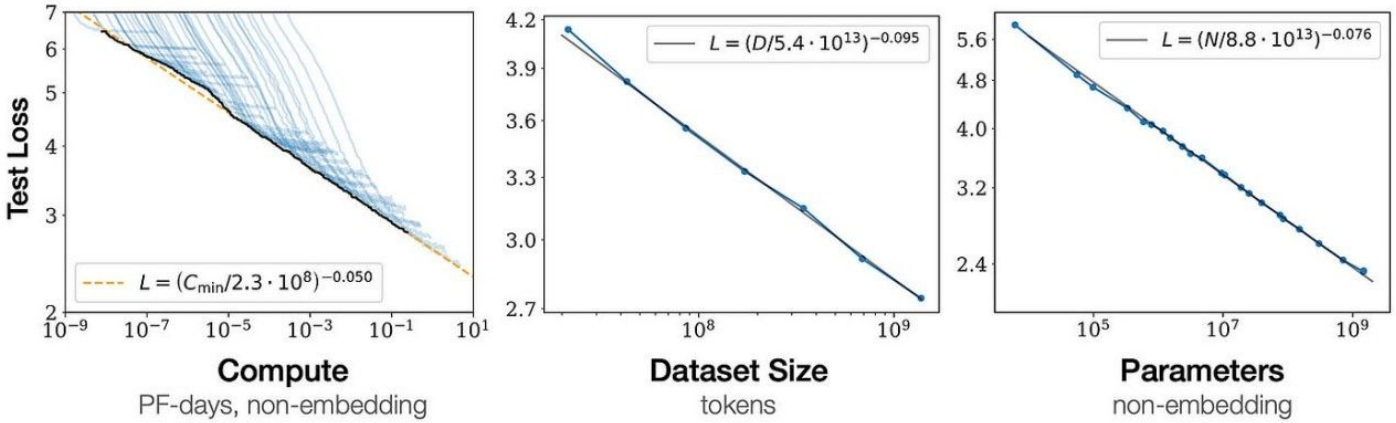


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

2001.08361

Previous scaling studies

Year	Study	Main result
2020	Kaplan et al. [1]	Power-law scaling of language-model loss with model size, dataset size, and training compute
2022	Hoffmann et al. [2]	Compute-optimal scaling requires coordinated increases of model capacity and training data
2023	Batson & Kahn [3]	Power-law dataset scaling in jet classification; different classifiers exhibit different scaling exponents
2025	CMS UParT v2 [4]	First CMS observations of improved flavour-tagging performance with increasing training statistics and model capacity
2026	ATLAS Carpe Datum [5]	Joint model-data-compute scaling for heavy-flavour transformers and validation at substantially increased training scale
2026	Vigl et al. [6]	Compute-optimal scaling for boosted-jet tagging; studies of asymptotic loss, repeated data, and input information

Training/testing data and fitting procedure

- Training datasets are constructed from different processes using mutually exclusive events of AK4 PUPPI jets, 10 GeV < pT < 937 GeV and |η|<2.5. Increasing D corresponds to adding unique jets; repeated presentation through additional epochs contributes to training compute.

- Performance is evaluated on an independent fixed test sample with 30 GeV < pT < 250 GeV that is not used during training or hyperparameter selection.

- Up to 30 charged candidates, 20 neutral candidates, 6 SVs, 4 V^0 candidates, and 435 pairwise interactions are retained per jet (to include ~97% of jets, similarly to the UParT v2 setup [4]). Flavour classification includes b, c, u, d, s, g, and 10 hadronic-τ decay-mode classes; electron-, muon-, and pileup-jet categories are excluded from these flavour classes.

- Fits throughout this note are obtained using non-linear least-squares minimization using the Trust Region Reflective algorithm for bounded parameters. Fit quality is quantified by the coefficient of determination

$$R^2 = 1 - \sum_i^N (y_i - \hat{y}_i)^2 / \sum_i^N (y_i - \bar{y})^2$$

where y_i are the measured values, $\hat{y}_i$ the fitted values, and $\bar{y}$ their mean. Values of R^2closer to 1 indicate that a larger fraction of the observed variance is described by the fitted scaling relation.

Training throughput and computational efficiency

- b-hive optimizations reduce CPU data-loading overhead and improve GPU utilization.
- CPU throughput increases from 104 to 1106 it/s.
- Nominal training throughput improves from 16 to 205 it/s.
- The resulting speedup makes the large scaling scans computationally feasible.

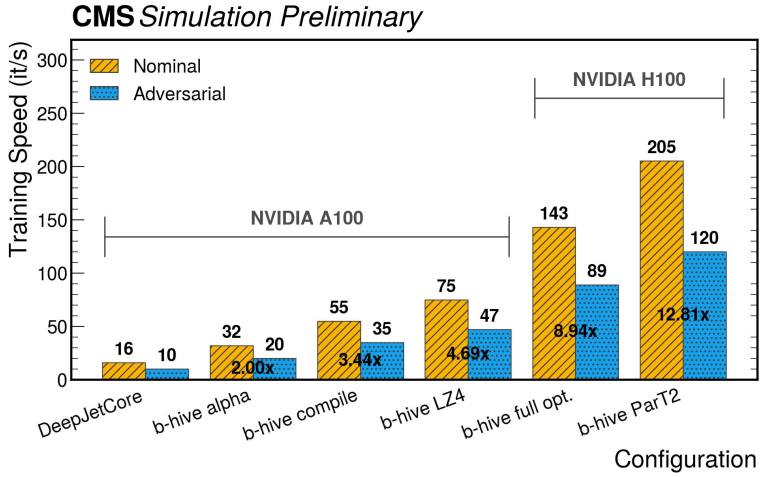


Figure 14: Evolution of the training throughput for successive b-hive configurations. Nominal and adversarial-training rates are shown in iterations per second. The transition from A100- to H100-based configurations is indicated.

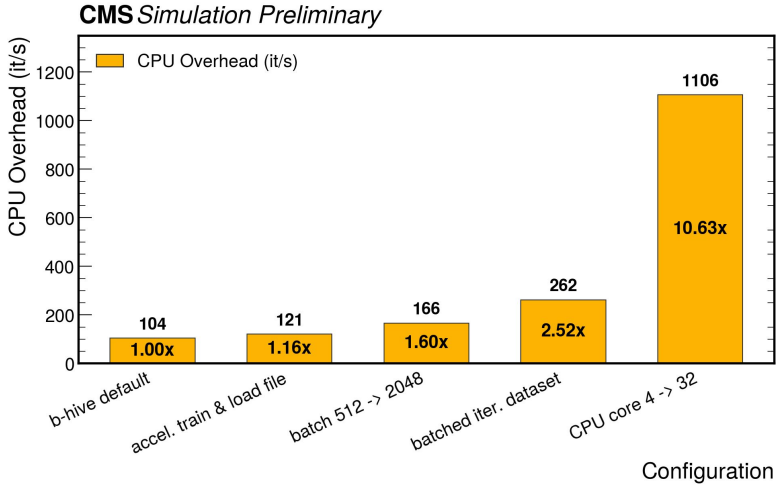


Figure 15: CPU-side data-processing throughput for successive input-pipeline optimizations. Memory and data-loading improvements increase the available CPU throughput by more than an order of magnitude, reducing the CPU bottleneck during GPU training.