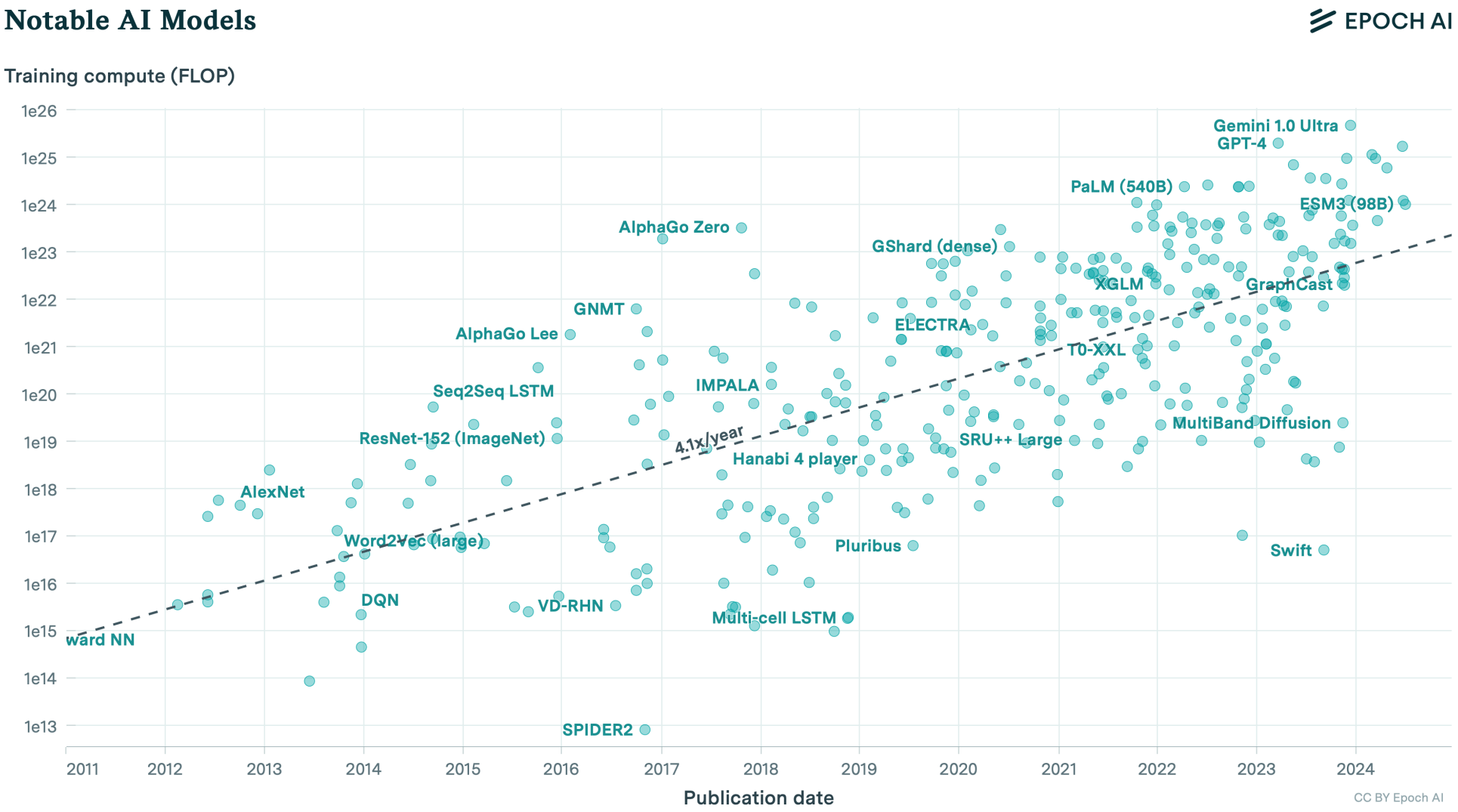
Can Tree-Based Model Improve Performance Prediction for LLMs?

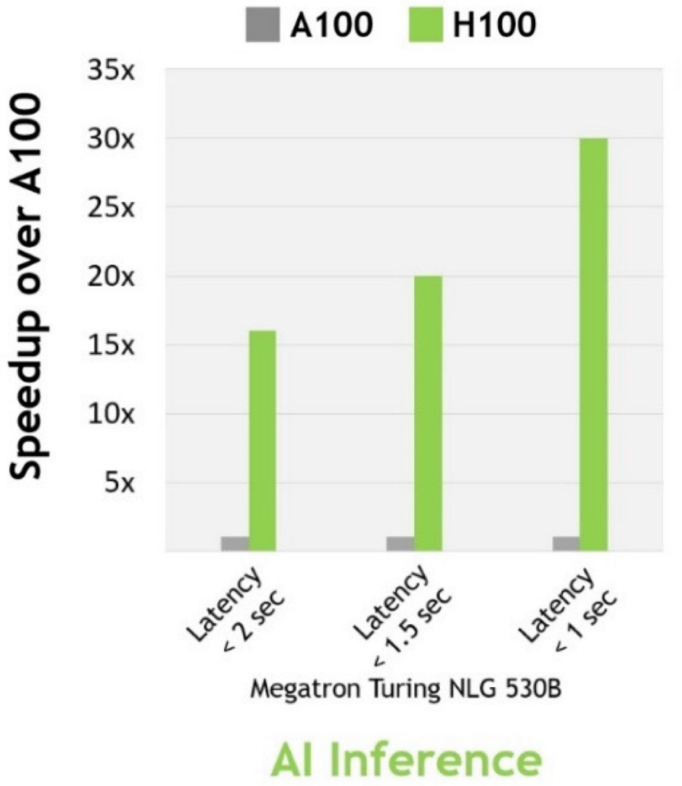
Karthick Panner Selvam, Mats Brorsson
University of Luxembourg

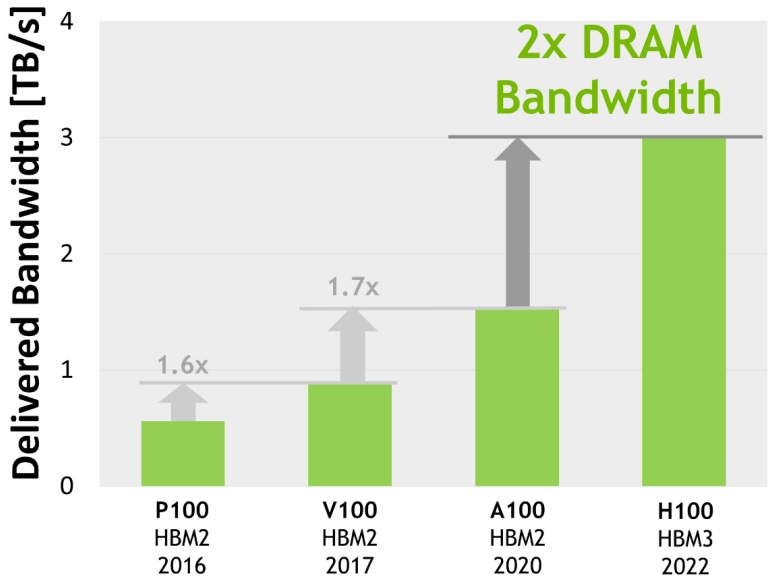
karthick.pannerselvam@uni.lu
mats.brorsson@uni.lu

Model complexity steadily increases



Computing power also increases





AI Deployment


Low cost


Low carbon emission

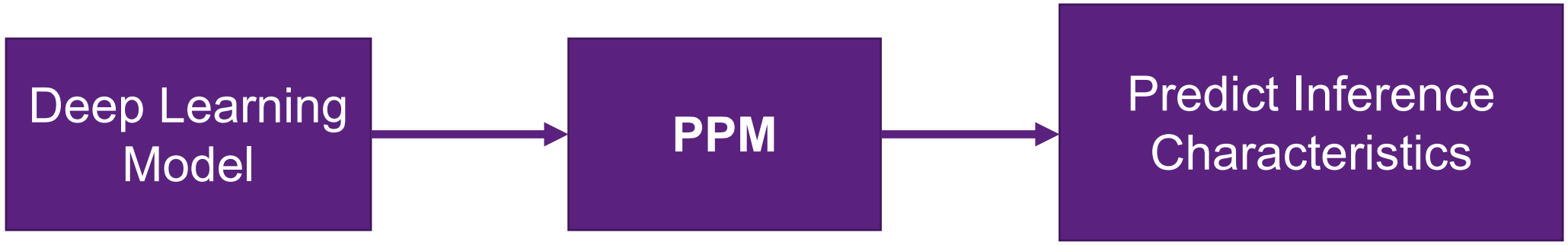

Without sacrificing performance

Why not just directly measure it on GPU ?

 It’s tedious to replicate for multiple models.

 Payment is required to access the GPU(s).

Performance Predictive Model

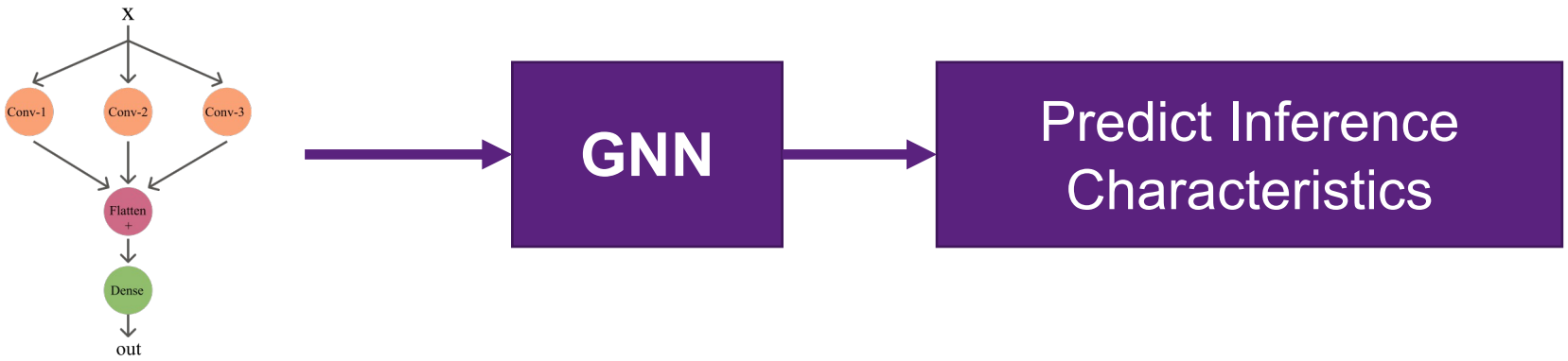


Predicted parameters help to

- Better resource allocation
- Cost saving
- Neural Architecture Search

Problem

- Vast space for performance prediction for LLMs
- Previous work widely used Graph Neural Network



1. *DIPPM: a Deep Learning Inference Performance Predictive Model using Graph Neural Network* – EuroPAR 2023

2. *Can Semi-Supervised Learning Improve Prediction of Deep Learning Model Resource Consumption?* – NeurIPS 2023 MLSys workshop

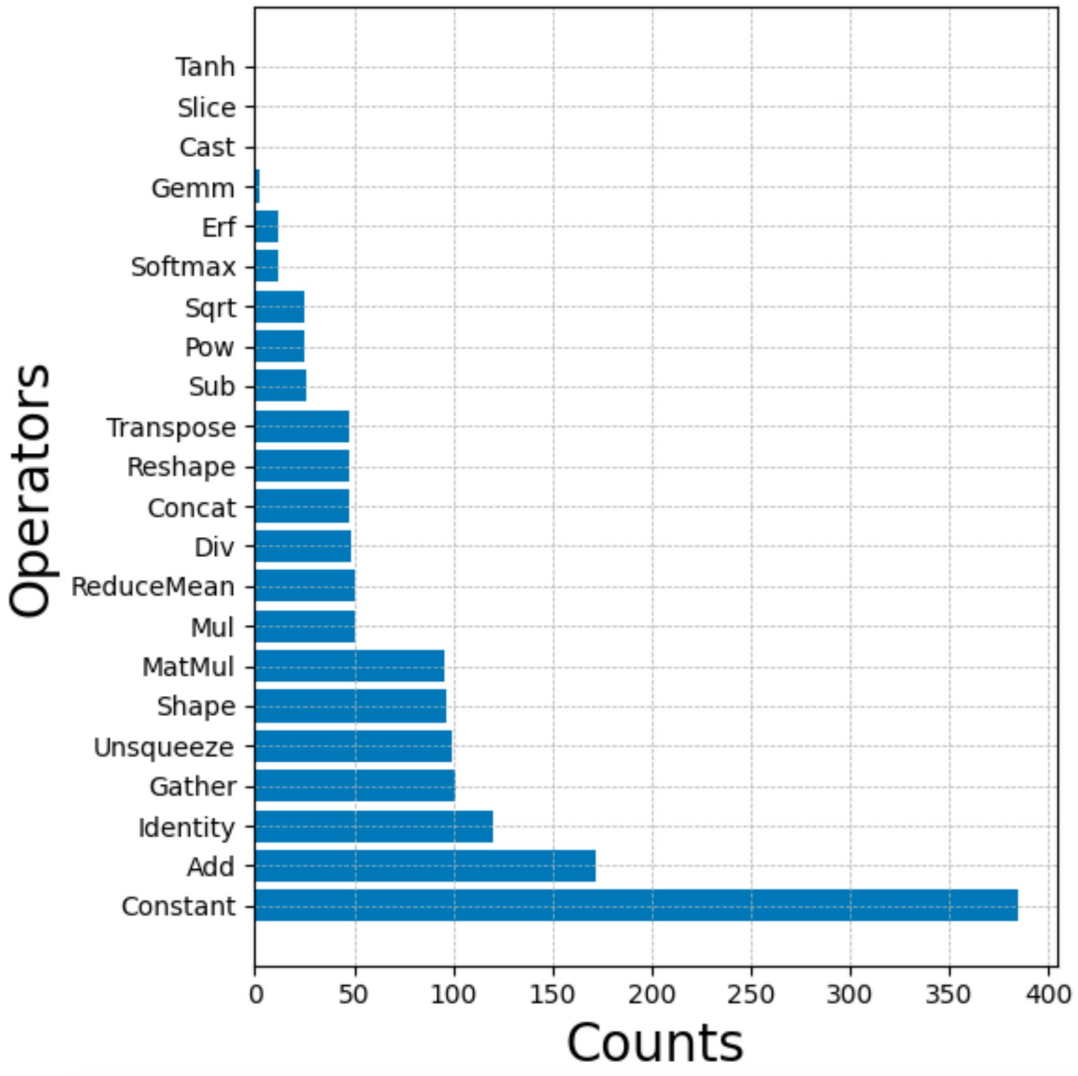
Can we use GNN for LLM performance prediction?

Yes, but it is computationally expensive.

LLMs Graph Analytics

Model	Nodes	Edges
bert-large-uncased [4]	2896	6621
xlm-roberta-base [3]	1495	3417
roberta-large [16]	2923	6681
microsoft-deberta-v3-small [12]	2450	5379
roberta-base [16]	1495	3417
bert-base-uncased [4]	1468	3357
distilbert-base-uncased [19]	685	1579
microsoft-deberta-v3-large [12]	9398	20643
microsoft-deberta-v3-base [12]	4766	10467

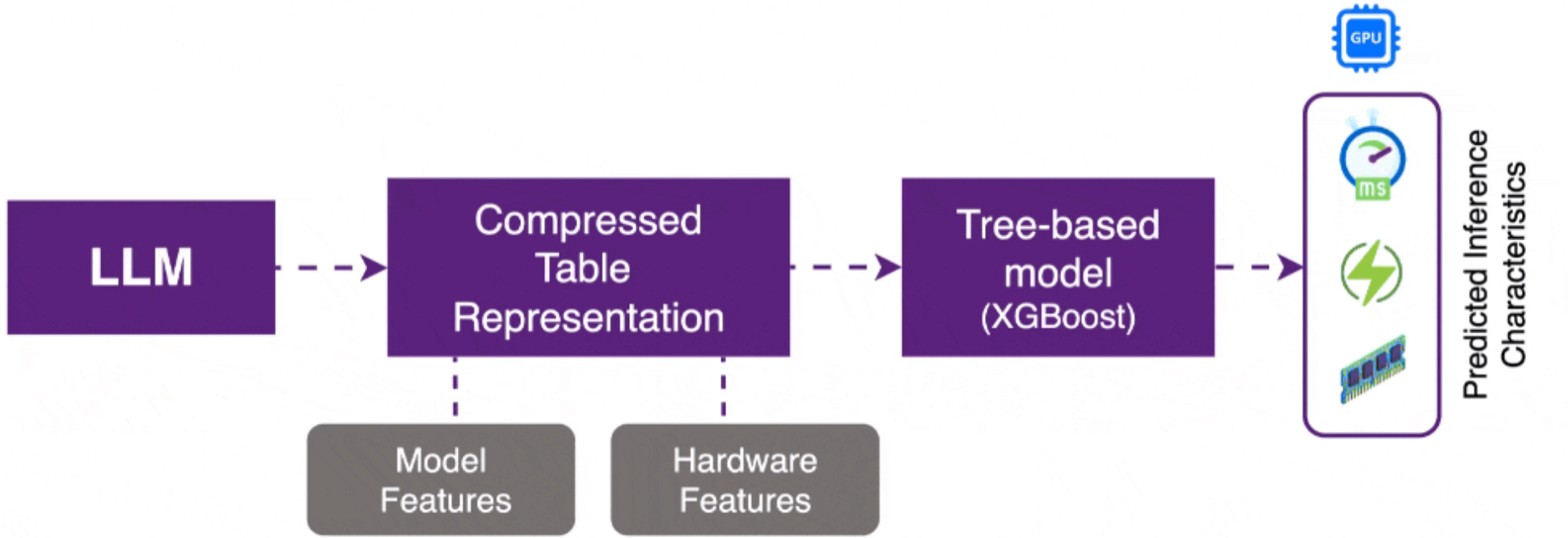
LLMs have many Nodes & Edges



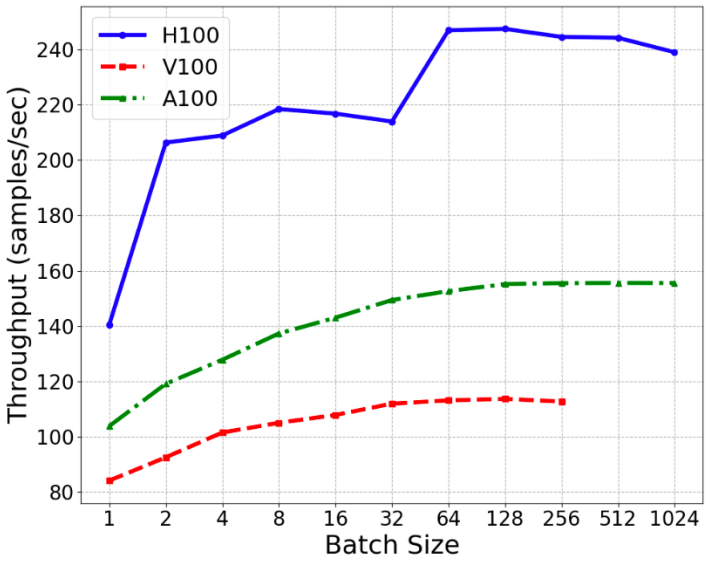
Ops count of Bert base model

LLMs layers are Repetitive

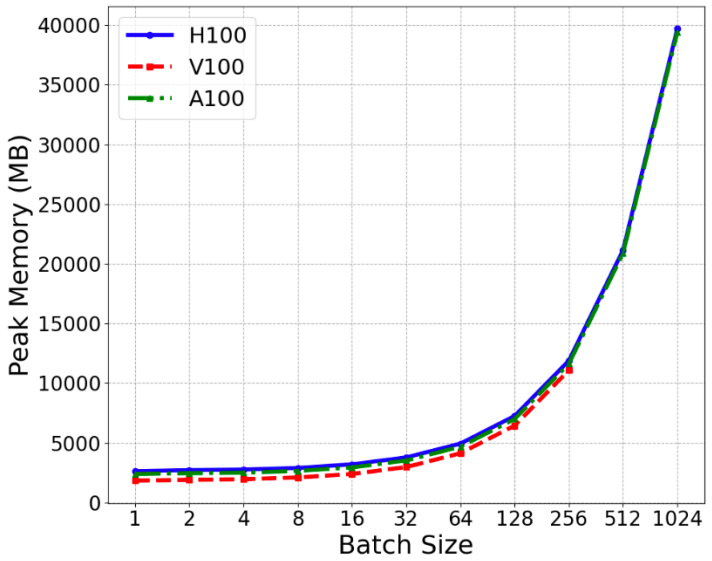
Our Solution



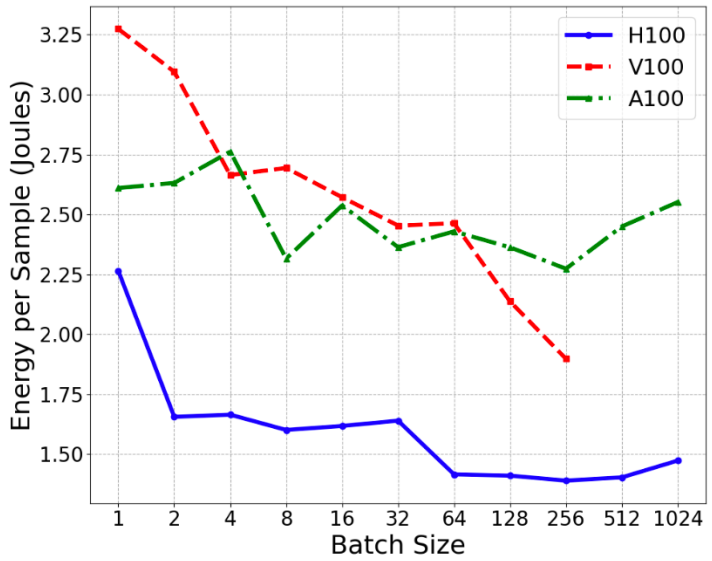
Batch Size



(a) Throughput vs. Batch Size



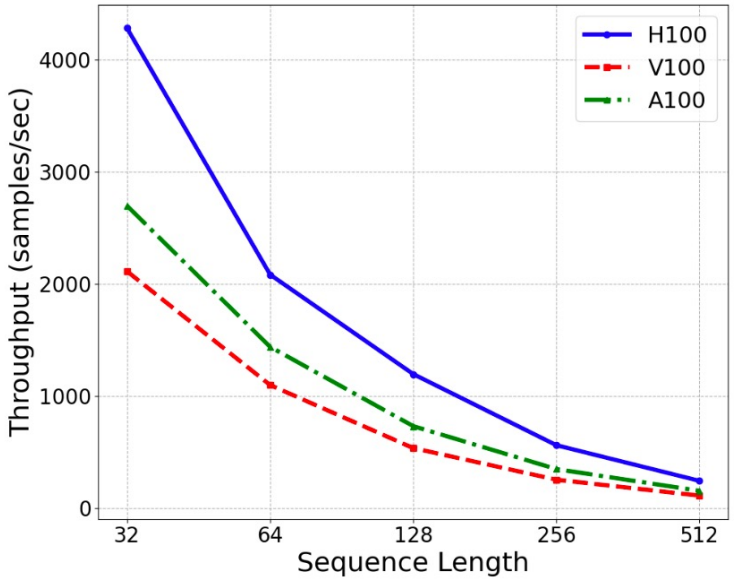
(b) Peak Memory vs. Batch Size



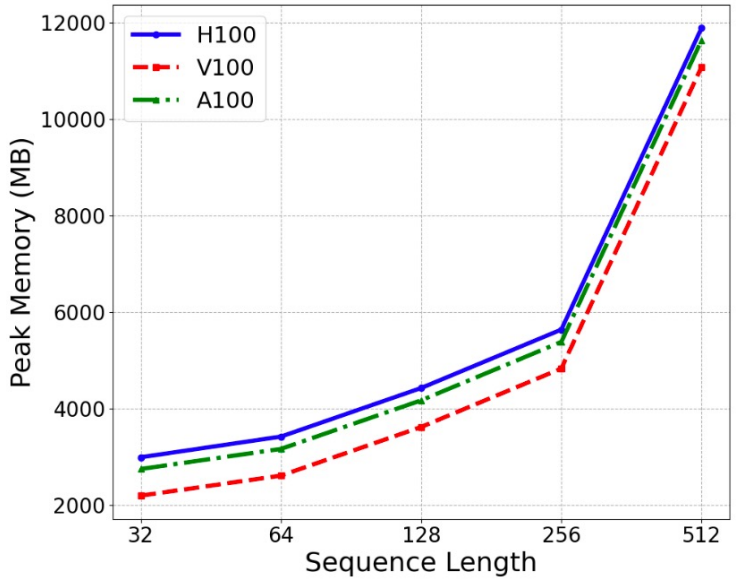
(c) Energy per sample vs. Batch Size

Roberta XLM model with a fixed sequence length of 512

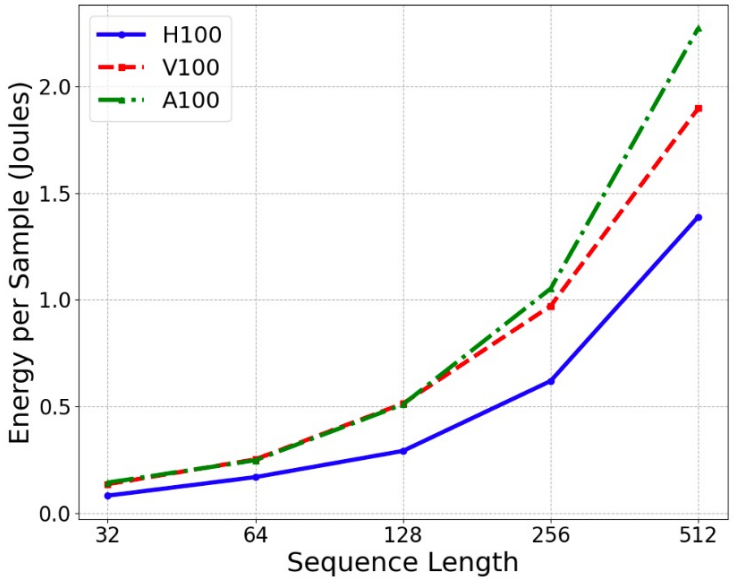
Sequence Length



(a) Throughput vs. Sequence Length



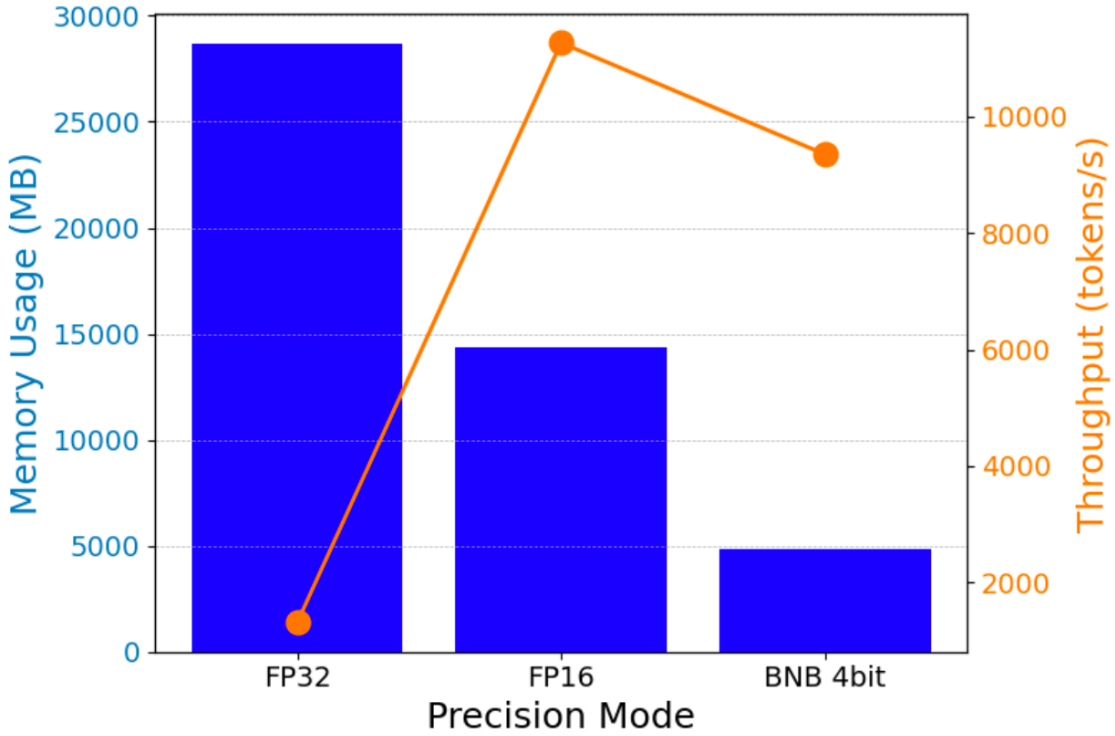
(b) Peak Memory vs. Sequence Length



(c) Energy per sample vs. Sequence Length

Roberta XLM model with a fixed batch size of 256

Quantization Strategies



Quantization Strategies for the **Llama 7B** model on the A100 device.

How does XGBoost work?

Objective Function:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

with Mean Squared Error (MSE) Loss for regression

The regularization penalize the model, helping to prevent overfitting

Experiment Setup

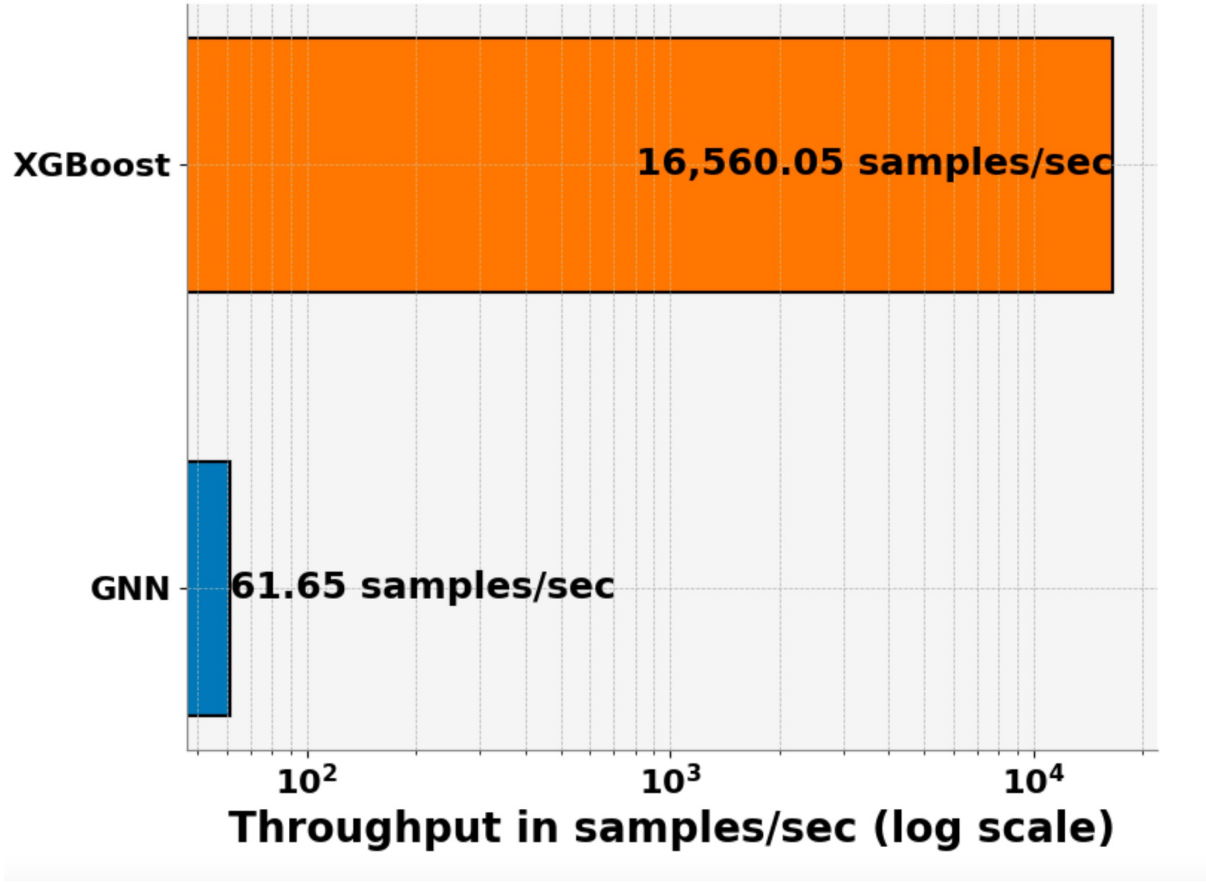
- We used NVIDIA H100, A100 and V100 GPUs to collect the dataset, a total of 1364 LLM model variations were collected.
- We used NVML and CUDA API to measure *Inference time*, *Memory,* and *Energy.*
- We also constructed graph dataset (Node Features Matrix & Adjacency Matrix) for SOA GNN baseline comparison

Results: Accuracy

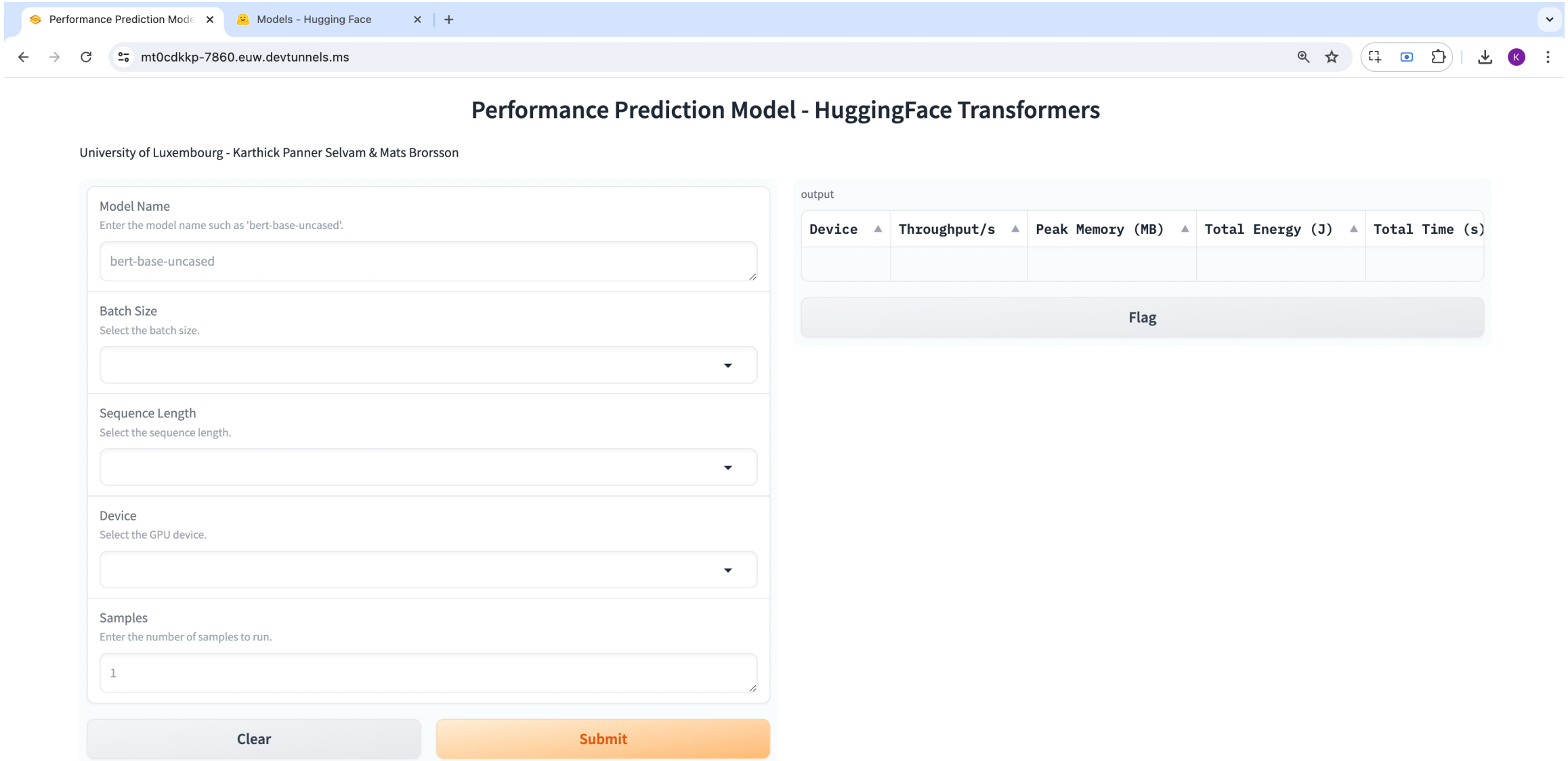
Target	GNN			XGBoost		
	MAPE (%) ↓	RMSE ↓	τ ↑	MAPE (%) ↓	RMSE ↓	τ ↑
Throughput	17.60	481.82	0.852	5.49	109.73	0.96
Memory	9.45	2657.45	0.874	1.81	1209.22	0.98
Energy	53.18	1.94	0.41	6.27	0.20	0.96

XGBoost accuracy is better than the GNN baseline.

Results: Speed



XGBoost is **268x** faster than the GNN baseline.



Performance Transformer – Simple Demo

Summary

Our study demonstrates the superiority of tree-based model over the GNN in predicting LLM performance across diverse hardware configurations, excelling in both **accuracy** and **speed**.

Thank You