HW-GPT-Bench

Hardware-Aware Architecture Benchmark for Language Models

Rhea Sanjay Sukthanker, Arber Zela, Benedikt Staffler, Aaron Klein
Lennart Purucker, Jörg K. H. Franke, Frank Hutter

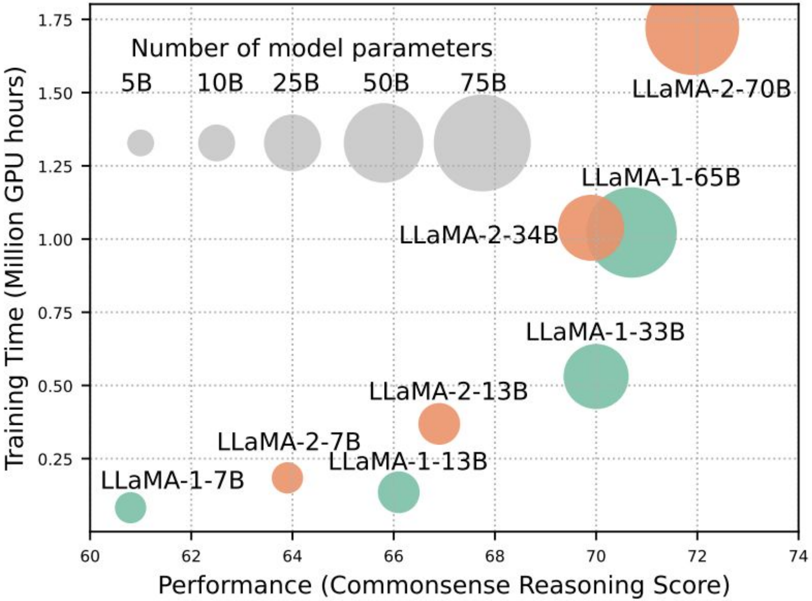
universität freiburg
ellis INSTITUTE TÜBINGEN
small data MORE FROM LESS CRC 1597
Funded by the European Union
BOSCH
SPONSORED BY THE Federal Ministry of Education and Research
ScaDS.AI DRESDEN LEIPZIG

https://arxiv.org/abs/2405.10299

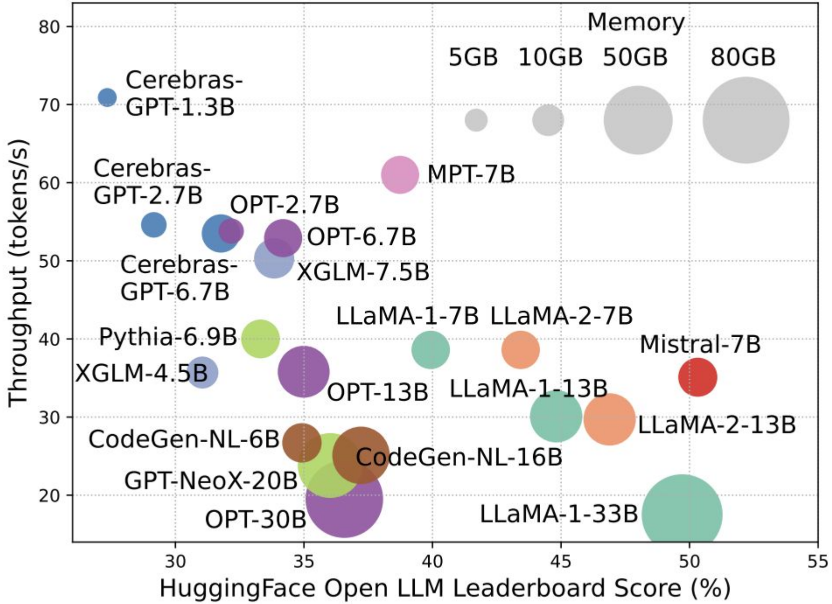
https://github.com/automl/HW-GPT-Bench/

Efficiency in Language Models

Training



Post-Training



Efficiency in Language Models

Training

- Data Selection
- Training Optimizers
- Mixed Precision
- Initialization Techniques
- Weight-Sharing

Post-Training

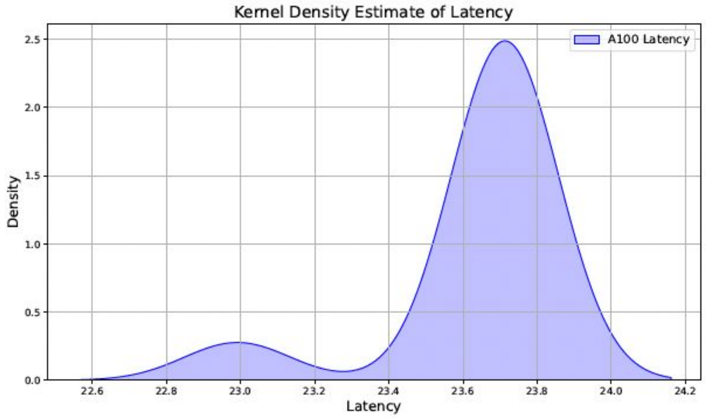
- Post-Training Quantization
- Pruning (Structured)
- Pruning (Unstructured)
- Knowledge Distillation
- Efficient Finetuning

HW-GPT-Bench: Efficient Pretraining with **Weight Sharing** + Search-based **Pruning**

Inference Metrics in Language Models

Efficiency Metrics

- GPU memory consumption
- Latency
- Energy
- Number of parameters
- Floating Point Operations (FLOPS)



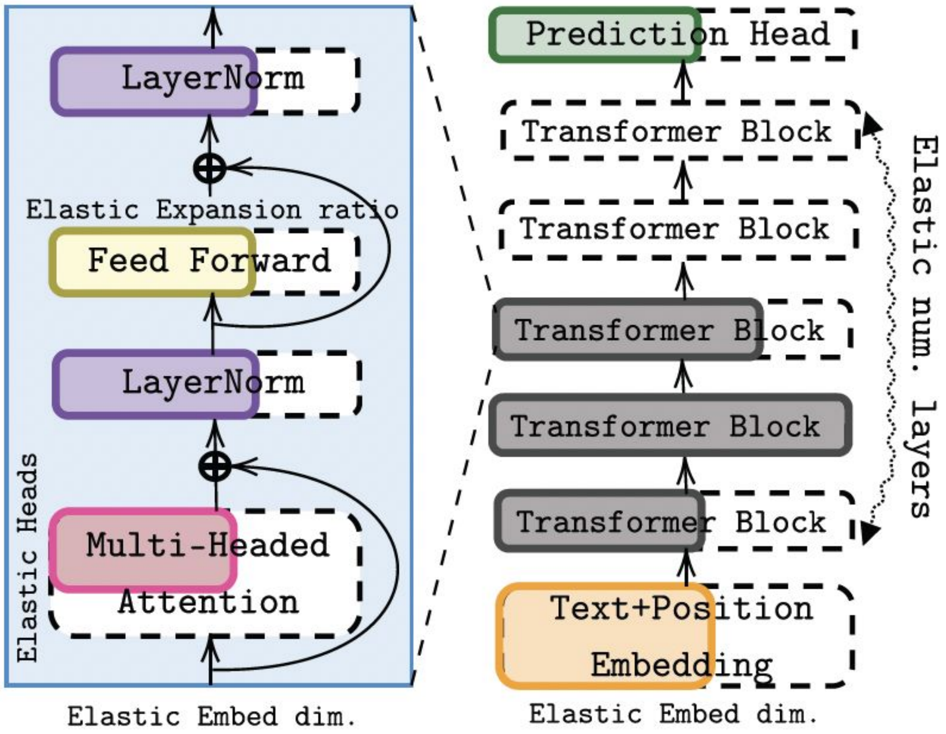
Deterministic

Noisy and device dependent

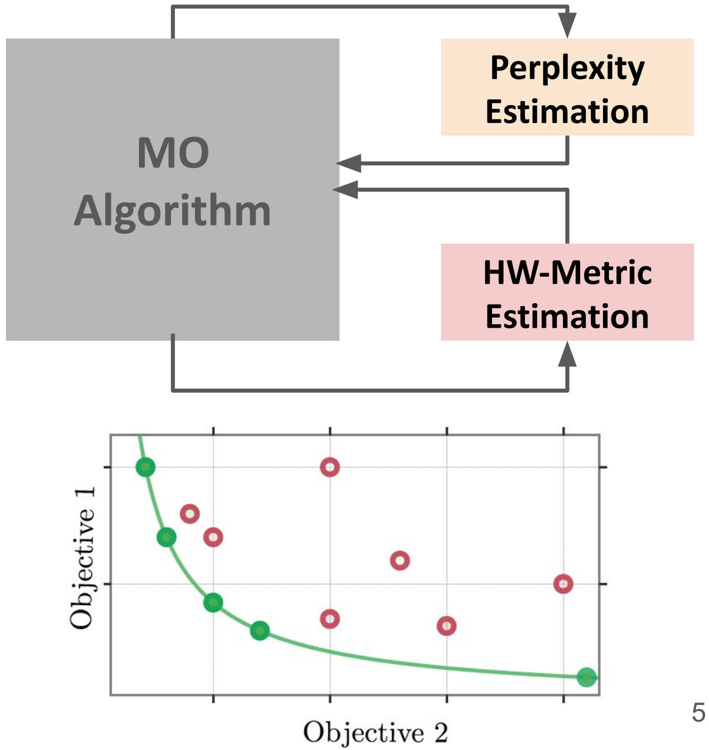
HW-GPT-Bench: Calibrate Latency/Energy Prediction + Memory + Parameters + FLOPS

Two-Stage Neural Architecture Search

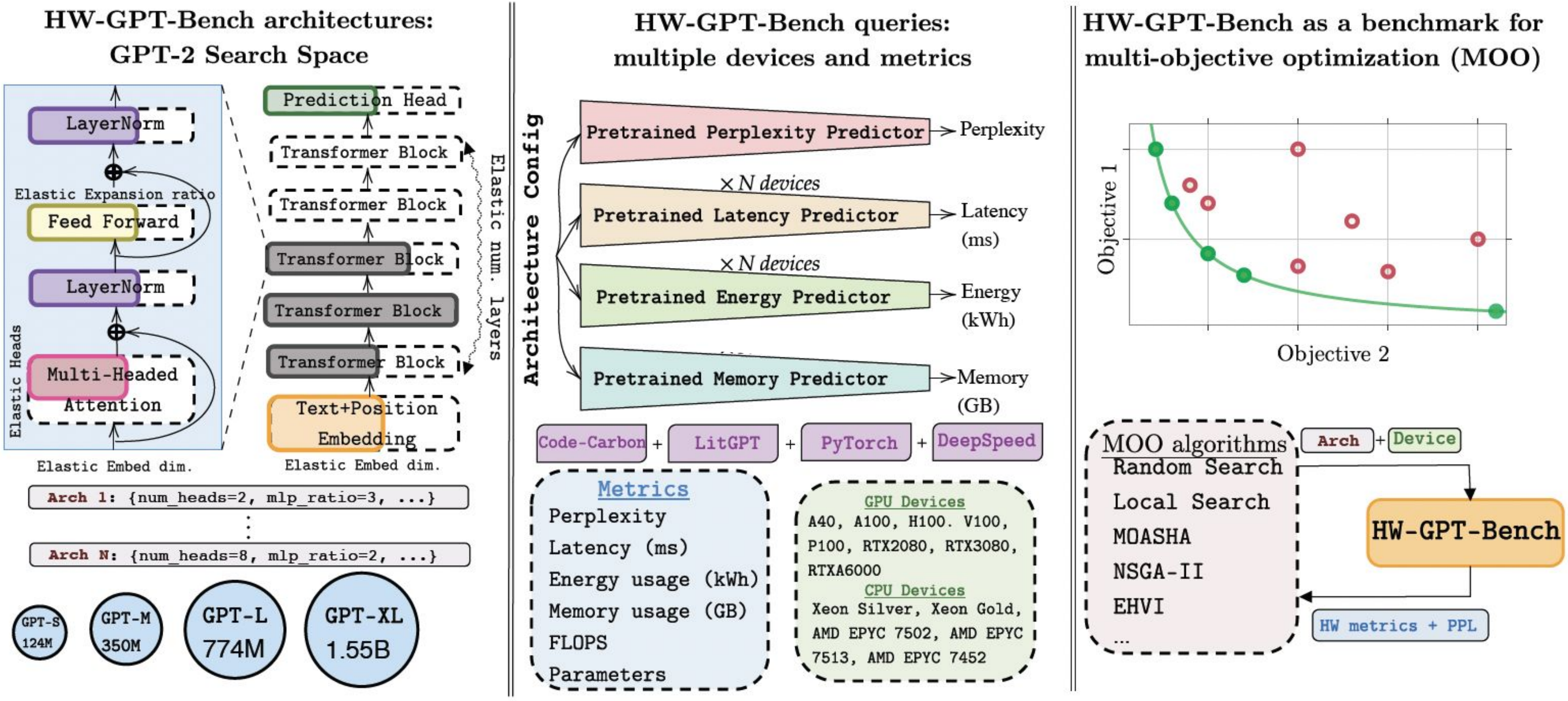
Stage 1: Train with Weight Sharing



Stage 2: Multi-Objective Search



HW-GPT-Bench: Overview



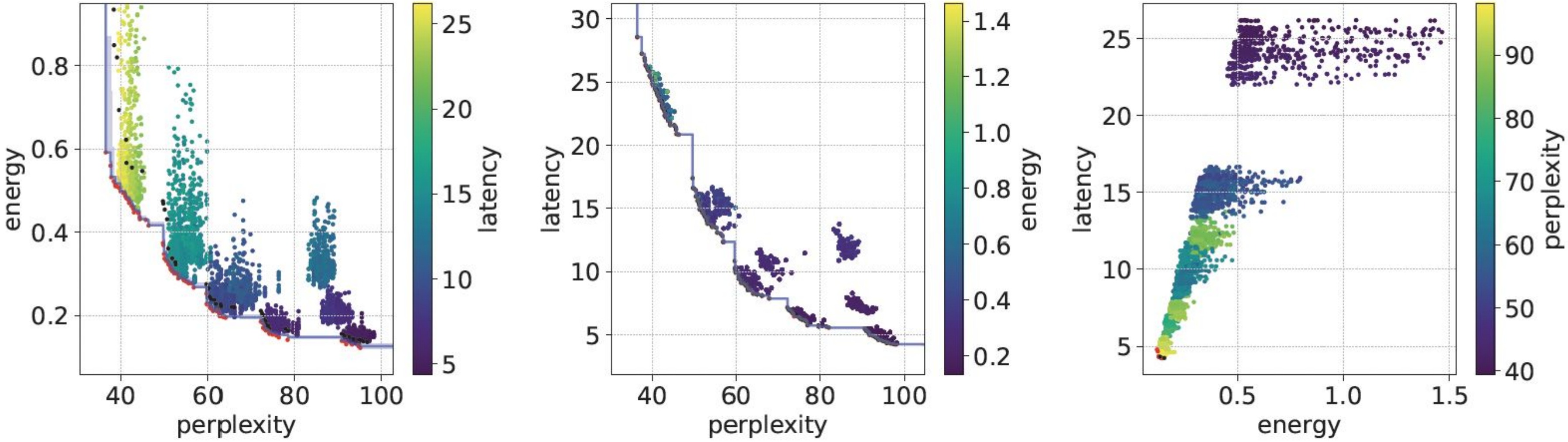
HW-GPT-Bench: Search Space Design

Supernet Type	Embedding Dim.	Layer No.	Head No.	MLP Ratio	Bias	No. of Archs	Supernet Size
GPT-S	[192, 384, 768]	[10, 11, 12]	[4, 8, 12]	[2, 3, 4]	[On, Off]	$\sim 10^{12}$	124M
GPT-M	[256, 512, 1024]	[22, 23, 24]	[8, 12, 16]	[2, 3, 4]	[On, Off]	$\sim 10^{24}$	350M
GPT-L	[320, 640, 1280]	[34, 35, 36]	[8, 16, 20]	[2, 3, 4]	[On, Off]	$\sim 10^{36}$	774M
GPT-S-wide	[192, 384, 768]	[3, 6, 12]	[3, 6, 12]	[1, 2, 4]	[On, Off]	$\sim 10^{12}$	124M
GPT-M-wide	[256, 512, 1024]	[6, 12, 24]	[4, 8, 16]	[1, 2, 4]	[On, Off]	$\sim 10^{24}$	350M
GPT-L-wide	[320, 640, 1280]	[9, 18, 36]	[5, 10, 20]	[1, 2, 4]	[On, Off]	$\sim 10^{36}$	774M
GPT-XL-wide	[400,800,1600]	[12,24,48]	[6, 12, 25]	[1, 2, 4]	[On, Off]	$\sim 10^{48}$	1.55B

- RoPE
- Parallel Residual
- Weight tying (Embedding and Head)
- GPT-S, -M, -L, -XL
- Two Search Space Variants

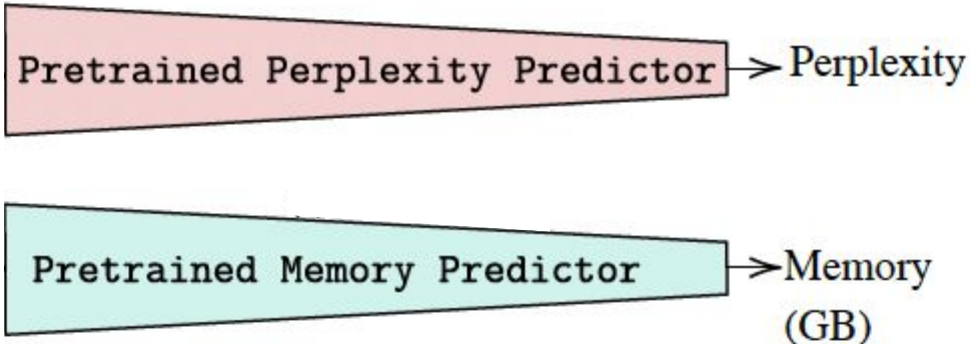
HW-GPT-Bench: Dataset Collection

- Pretrain supernet on openwebtext
- Sample 10000 unique architectures per search space
- Perplexity computed by inheriting subnetworks (openwebtext validation set)
- 10 observations for latency (8 GPUS, 5 CPUS): **PyTorch Profiler**
- 50 observations for energy (8 GPUS, 5 CPUS): **Codecarbon**, **NVIDIA profiling**



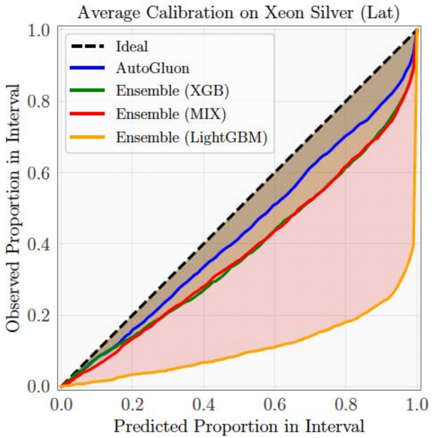
HW-GPT-Bench: Modeling the Perplexity and Memory Surrogate

- Simple MLP, 4 linear layers, 128 hidden, ReLU activation
- Kendall-Tau Correlation of **> 0.9** across search spaces



HW-GPT-Bench: Modeling the Latency/Energy Surrogate

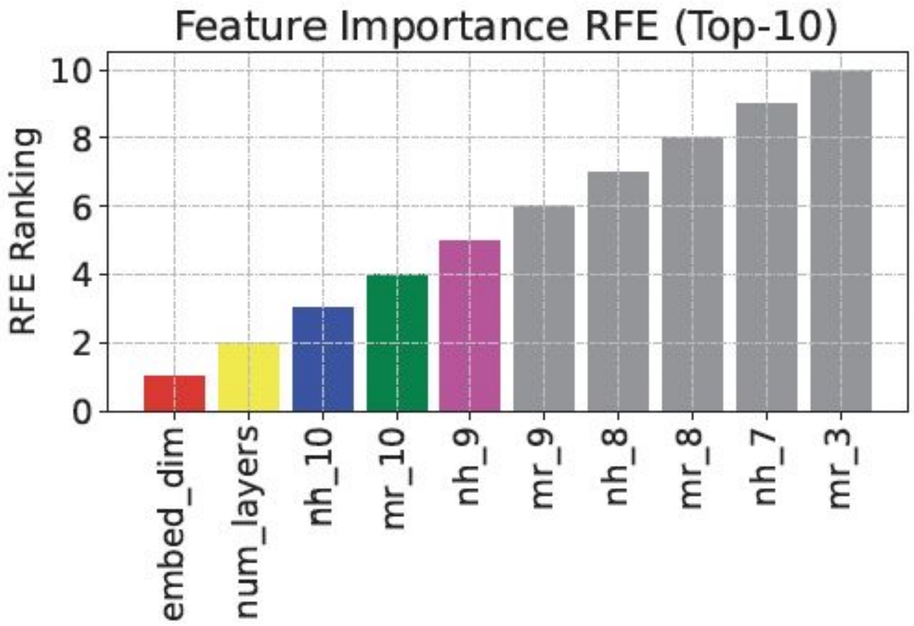
Surrogate	Accuracy												Calibration					
	MAE ↓		RMSE ↓		MDAE ↓		MARPD ↓		R^2 ↑		Corr. ↑		RMS Cal. ↓		MA Cal. ↓		Miscal. Area ↓	
	H100	2080	H100	2080	H100	2080	H100	2080	H100	2080	H100	2080	H100	2080	H100	2080	H100	2080
AutoGluon	**0.153**	**1.80**	**0.211**	**2.576**	**0.111**	1.121	**0.153**	**10.677**	**0.999**	**0.904**	**0.999**	**0.950**	**0.223**	**0.244**	**0.198**	**0.217**	**0.199**	**0.220**
Ensemble (Mix)	0.569	1.830	0.785	2.621	0.413	**1.092**	0.565	10.920	0.999	0.900	0.999	0.949	0.472	0.298	0.411	0.264	0.415	0.267
Ensemble (XGB)	0.620	1.832	0.827	2.628	0.475	1.154	0.629	10.919	0.990	0.899	0.990	0.948	0.481	0.286	0.417	0.251	0.421	0.254
Ensemble (LGB)	0.361	2.094	0.411	2.922	0.379	1.415	0.384	13.140	0.970	0.875	0.999	0.947	0.559	0.347	0.481	0.304	0.486	0.308



- Model mean and variance of the distribution
- AutoGluon (stacked ensemble) outperforms other ensembling methods
- Sample from the gaussian with predicted mean & variance

HW-GPT-Bench: Analysis and Interpretability

Embedding dim and layer number quite important!



GPT-S: $y = 646.234 \cdot l^{-0.226} \cdot e^{-0.371} \cdot m^{-0.100} \cdot h^{-0.076} \cdot b^{-0.001}$

GPT-M: $y = 404.456 \cdot l^{-0.104} \cdot e^{-0.343} \cdot m^{-0.091} \cdot h^{-0.049} \cdot b^{-0.005}$

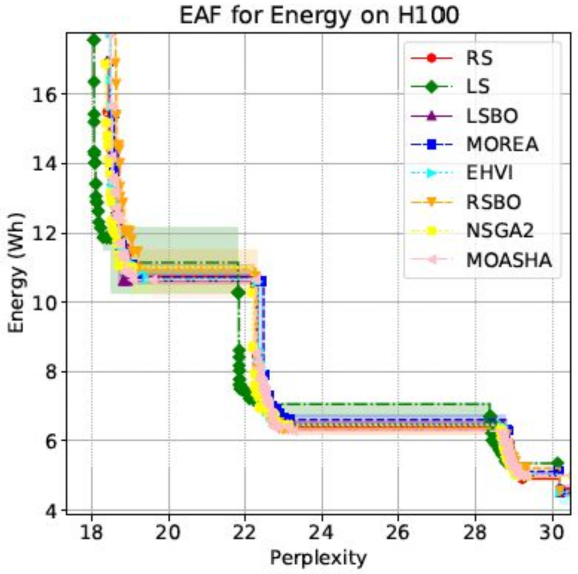
GPT-L: $y = 280.757 \cdot l^{-0.073} \cdot e^{-0.309} \cdot m^{-0.088} \cdot h^{-0.051} \cdot b^{-0.005}$

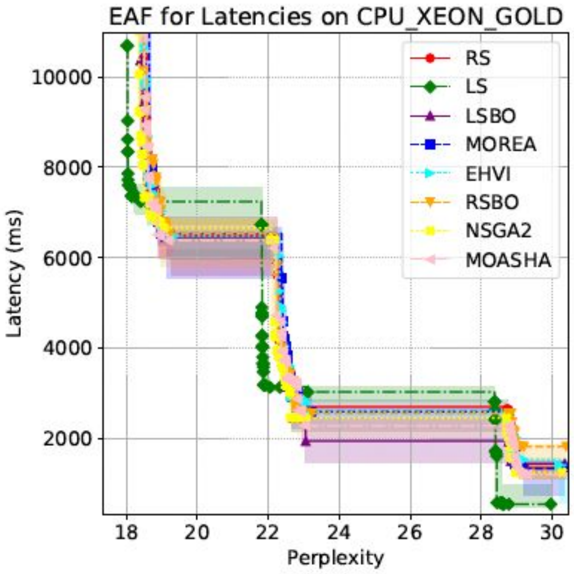
HW-GPT-Bench as a Benchmark for Multi-objective Optimization

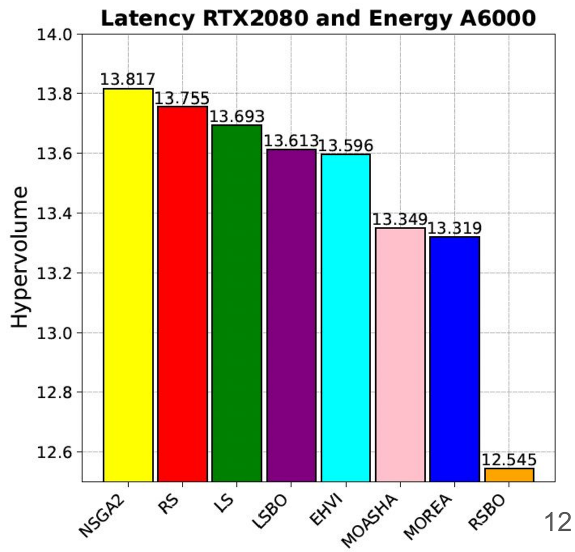
Use a variety of optimizers from syne-tune

2-objectives

3 objectives







HW-GPT-Bench API

```
from hwgpt.api import HWGPT
api = HWGPT(search_space="s") # initialize API
random_arch = api.sample_arch() # sample random arch
api.set_arch(random_arch) # set  arch
results = api.query() # query all
energy = api.query(metric="energy") # query energy
rtx2080 = api.query(device="rtx2080") # query device
# query perplexity based on mlp predictor
perplexity_mlp = api.query(metric="perplexity",predictor="mlp")
# query perplexity based on supernet
perplexity_supernet = api.query(metric="perplexity",predictor="supernet")
# run baseline and plot EAF
nsga2_results = api.run_baseline(method="nsga2", device="rtx2080", metrics=["energy","perplexity"],
     ppl_predictor="mlp")
# plot Pareto-front
api.plot_eaf(nsga2_results)
```

Easy to use API for a variety of devices and model scales

Takeaway:

A new and calibrated hw-aware benchmark for Language Model Architectures

https://arxiv.org/abs/2405.10299

https://github.com/automl/HW-GPT-Bench/