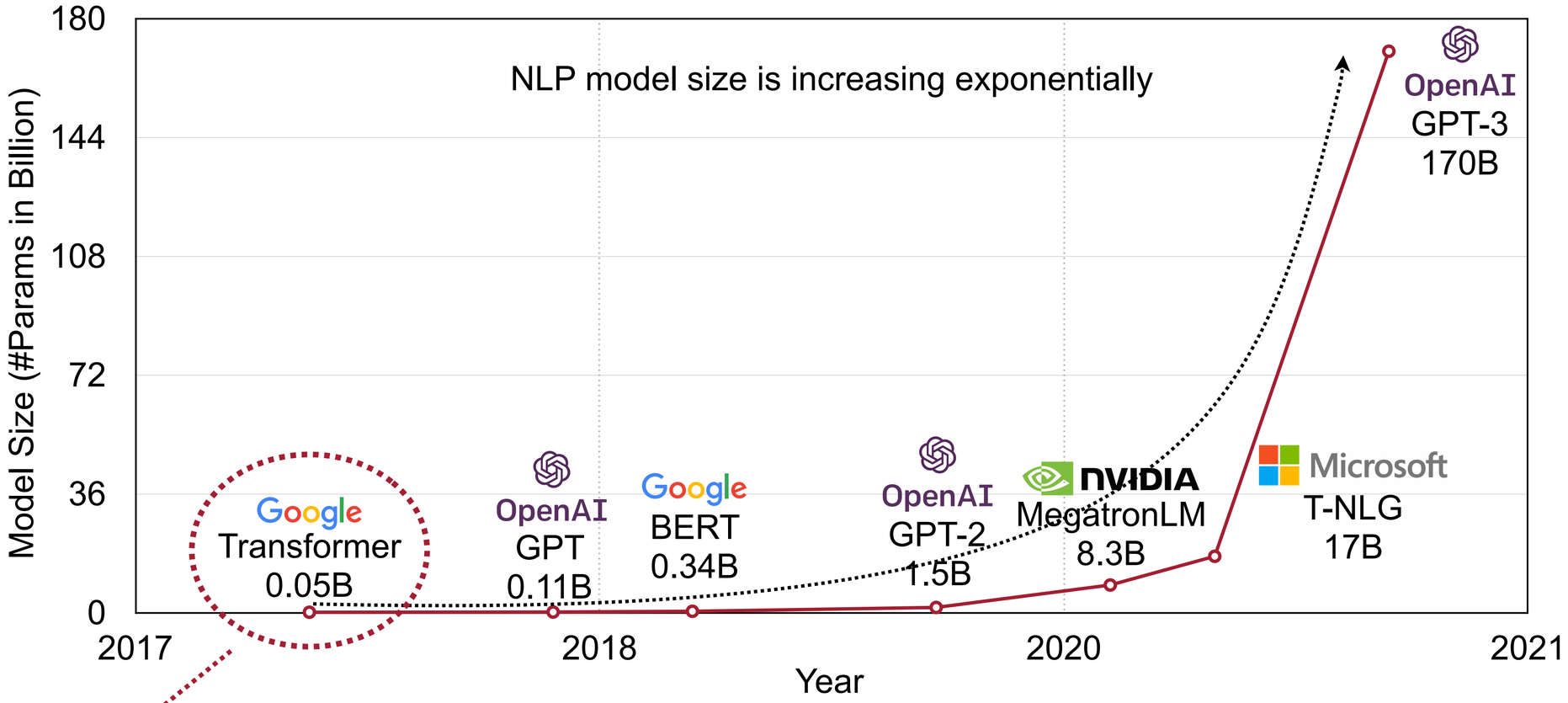
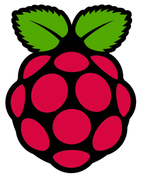
HAN LAB
Hardware, AI and Neural-nets

HAT: Hardware-Aware Transformers for Efficient Natural Language Processing

Hanrui Wang[1], Zhanghao Wu[1], Zhijian Liu[1], Han Cai[1], Ligeng Zhu[1],
Chuang Gan[2] and Song Han[1]
[1]Massachusetts Institute of Technology
[2]MIT-IBM Watson AI Lab

Transformers are Inefficient





- Raspberry Pi takes 20 seconds to translate a 30-token sentence with Transformer-Big model

Searching for an Efficient Transformers is Inefficient

We need Green AI and Tiny AI

MIT Technology Review

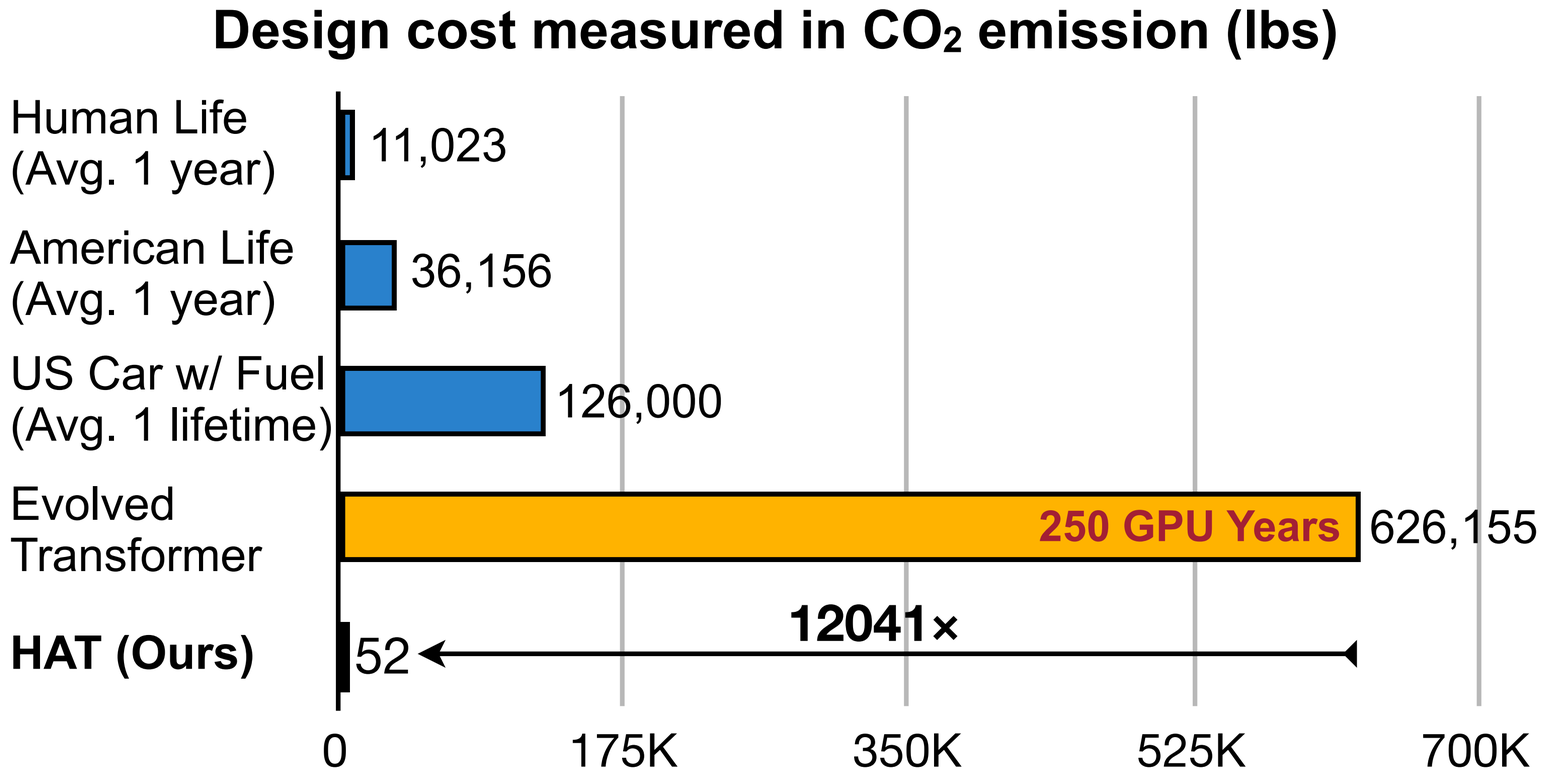
Artificial intelligence / Machine learning

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by **Karen Hao** June 6, 2019

The artificial-intelligence industry is often compared to the oil industry: once mined and refined, data, like oil, can be a highly lucrative commodity. Now it seems the metaphor may extend even further. Like its fossil-fuel counterpart, the process of deep learning has an outsize environmental impact.



- Previous AutoML has huge searching cost and CO_2 emission*

*Strubell, E., Ganesh, A., & McCallum, A. Energy and policy considerations for deep learning in NLP. ACL 2019

HAT: Hardware-Aware Transformers, ACL 2020

Hardware-Aware Transformers

Efficiently search for efficient Transformer architectures

250 GPU Years
$5,500,000
626,000 lbs CO_2

184 GPU Hours
$456
52 lbs CO_2

Search in a weight-sharing supernet "SuperTransformer"

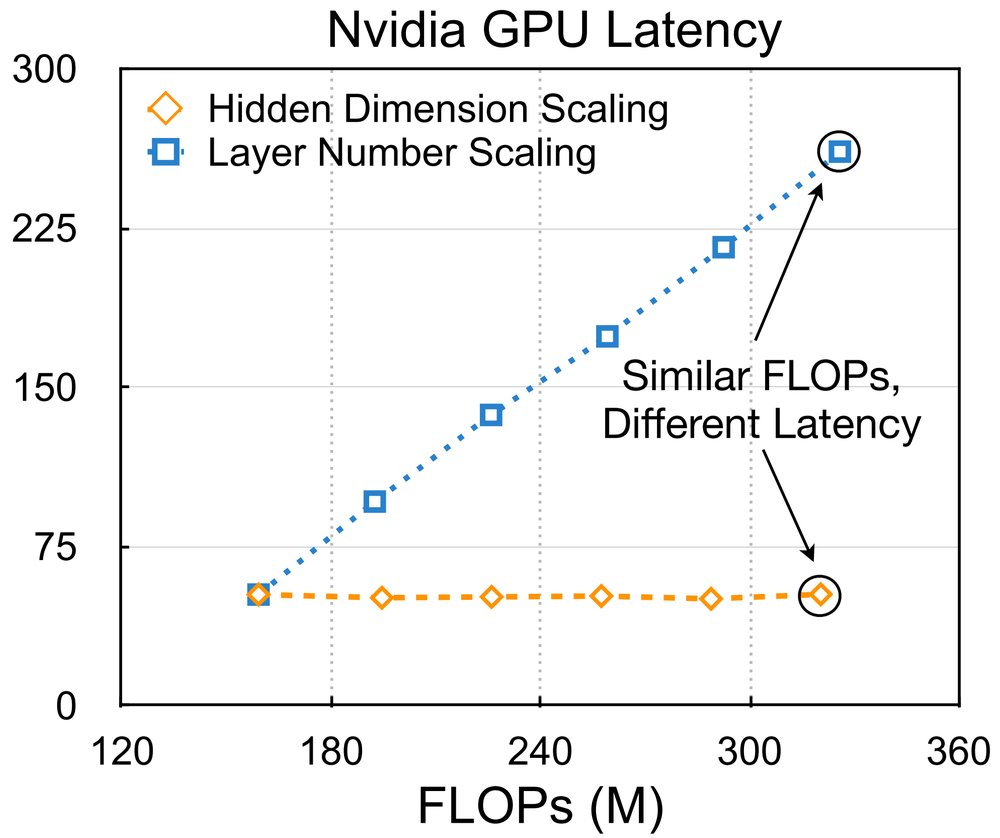
(1) Hardware latency feedback in the search loop

(2) Novel design space

737ms on Xeon
176M model size
10.6 GFLOPs

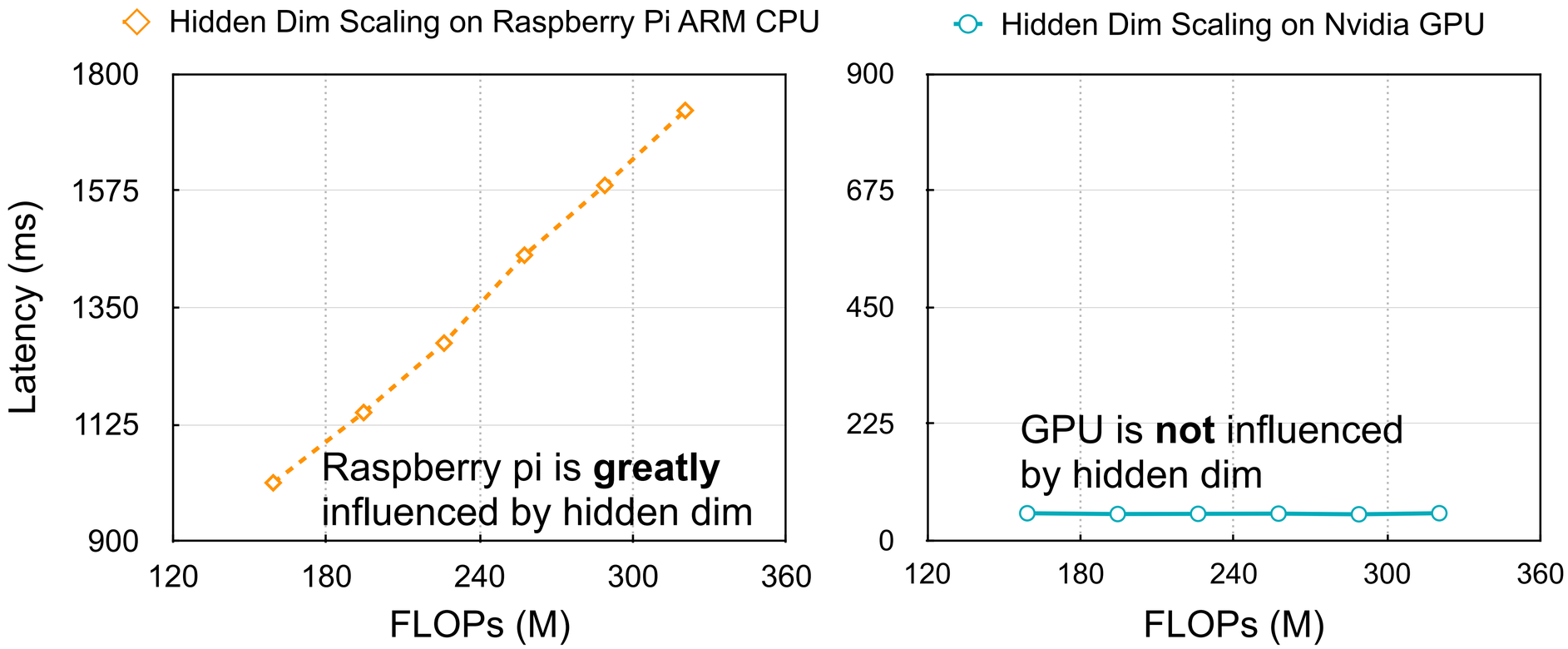
329ms on Xeon
44M model size
2.7 GFLOPs

Two Common Pitfalls in Efficiency Evaluation



- FLOPs cannot reflect real latency

Two Common Pitfalls in Efficiency Evaluation

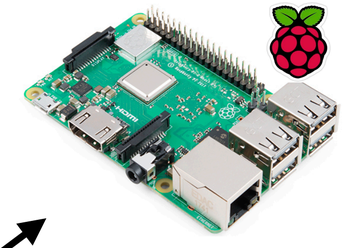


- Latency is influenced by different factors on different hardware
- Different hardware has different strategies for efficient model design

From General to Specialized Transformers

Previous Paradigm:

A general Transformer for all hardware

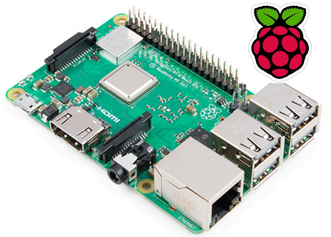


Low Efficiency

HAT:

One **specialized** Transformer model for each hardware

HAT Pi Model



HAT GPU Model

HAT CPU Model

HAT Mobile Model

High Efficiency

Three Steps in HAT

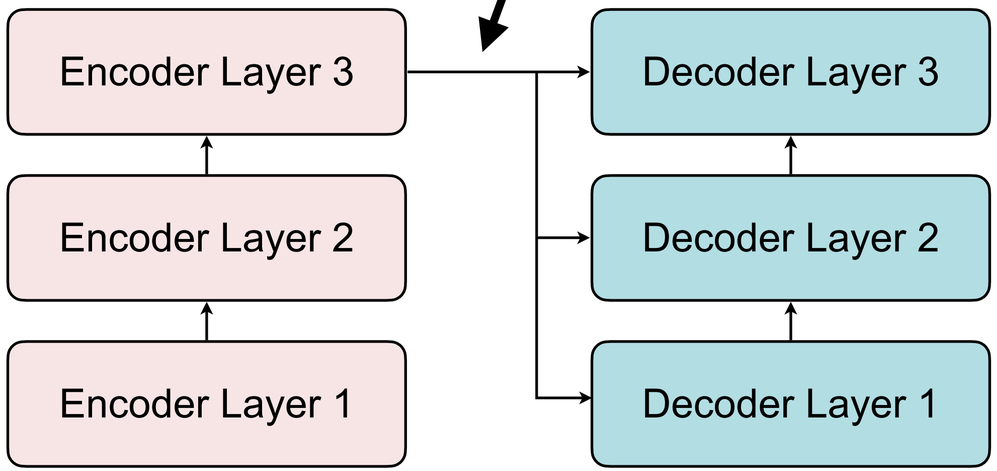
- Train a SuperTransformer
- Evolutionary search with a hardware latency constraint to find a SubTransformer
- Train the SubTransformer from scratch

SuperTransformer Design Space

Breaking Two Conventional Design Rules

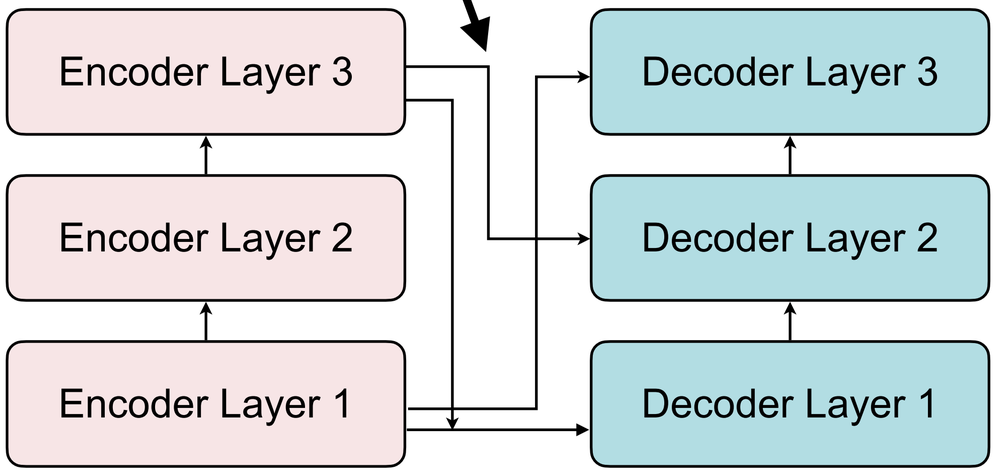
Convention

Information bottleneck



HAT

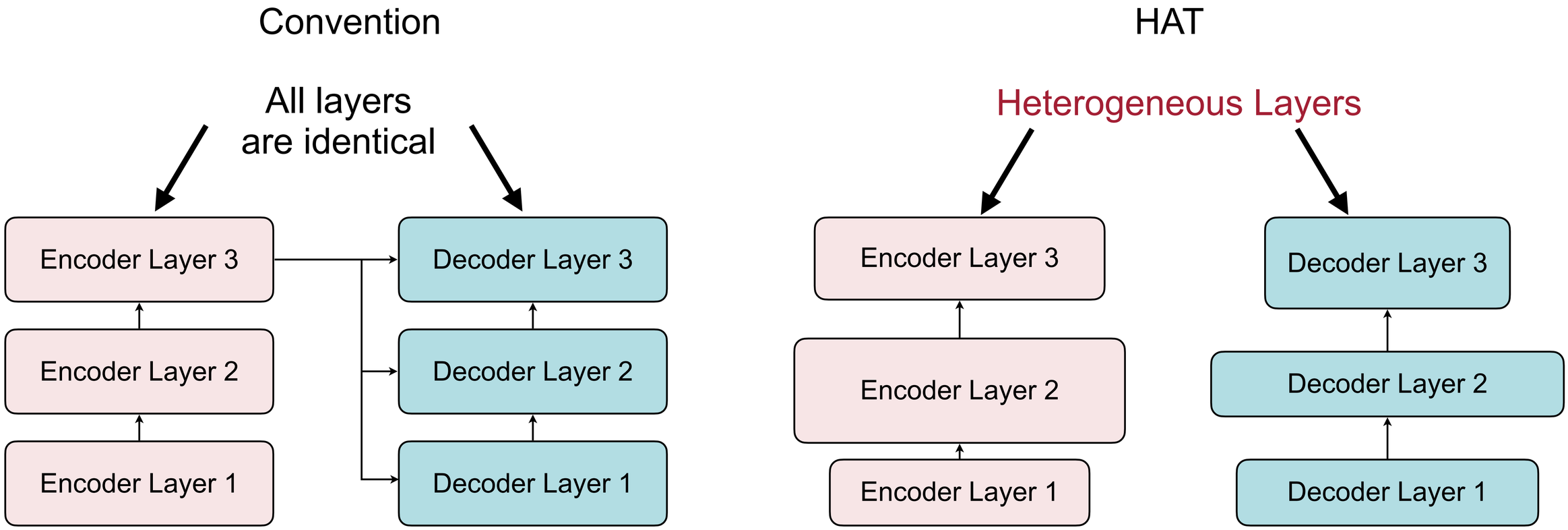
Arbitrary Encode-Decoder Attention



- Each decoder layer can attend to arbitrary encoder layers

SuperTransformer Design Space

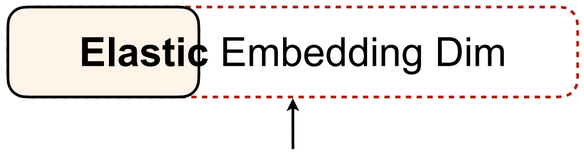
Breaking Two Conventional Design Rules



- Each layer has different architecture

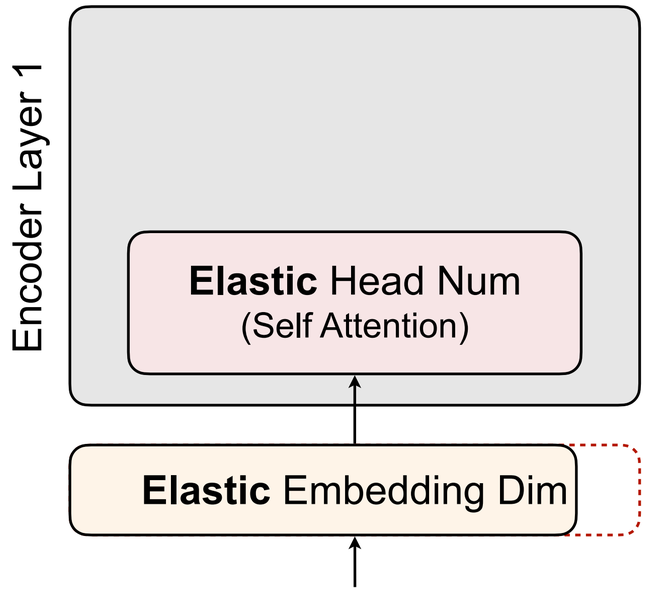
Elastic SuperTransformer with Weight-Sharing

- Elastic embedding dimension, share the front part of embedding



Elastic SuperTransformer with Weight-Sharing

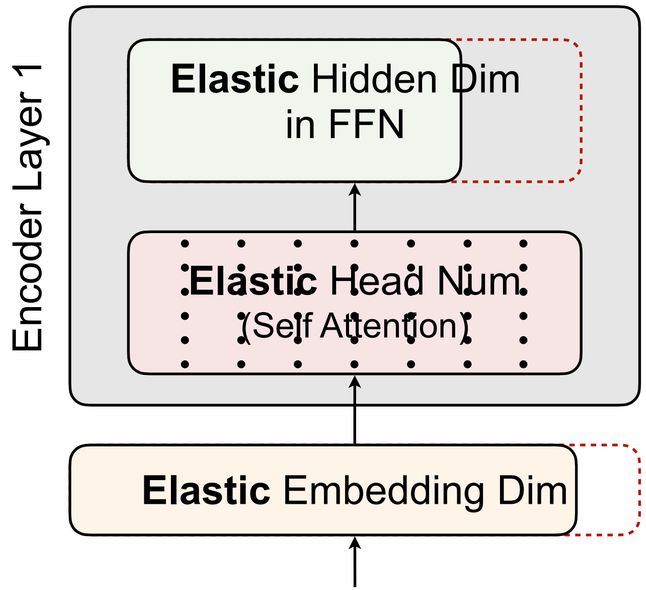
- Elastic head number in all self-attention and cross-attention, share Q,K,V
- Elastic embedding dimension, share the front part of embedding



1 Head

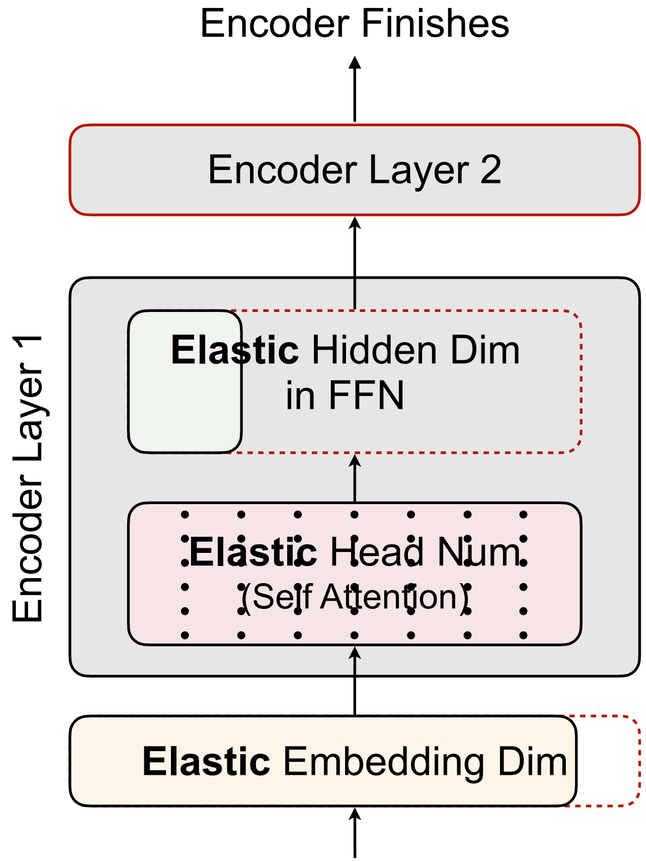
Elastic SuperTransformer with Weight-Sharing

- Elastic hidden dimension, share the front part of weights
- Elastic head number in all self-attention and cross-attention, share Q,K,V
- Elastic embedding dimension, share the front part of embedding



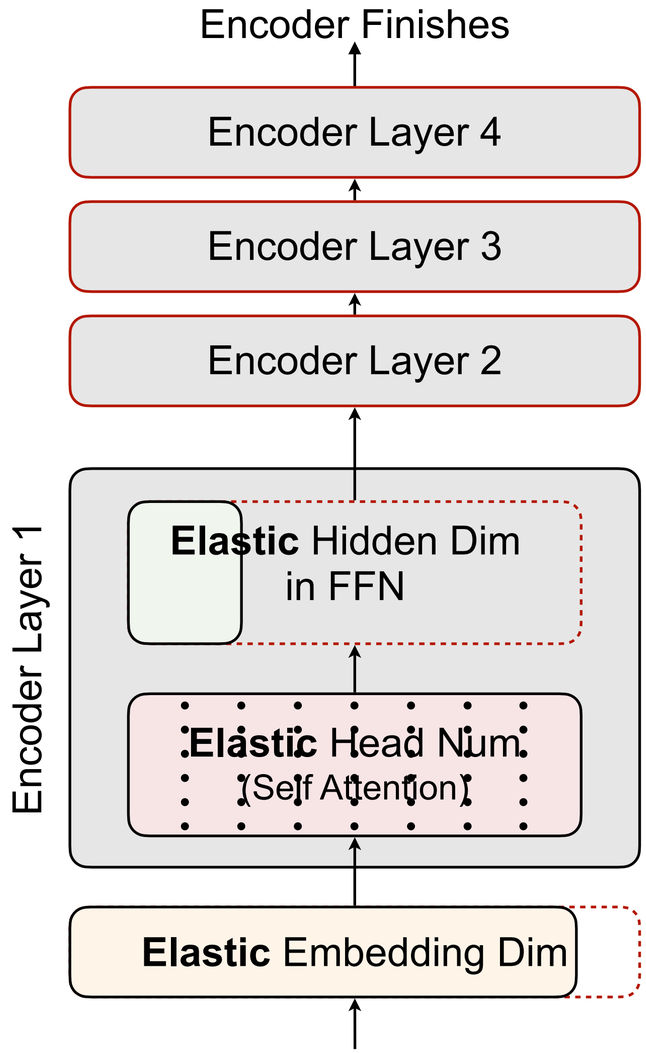
Elastic SuperTransformer with Weight-Sharing

- Elastic number of layers, share the front several layers
- Elastic hidden dimension, share the front part of weights
- Elastic head number in all self-attention and cross-attention, share Q,K,V
- Elastic embedding dimension, share the front part of embedding

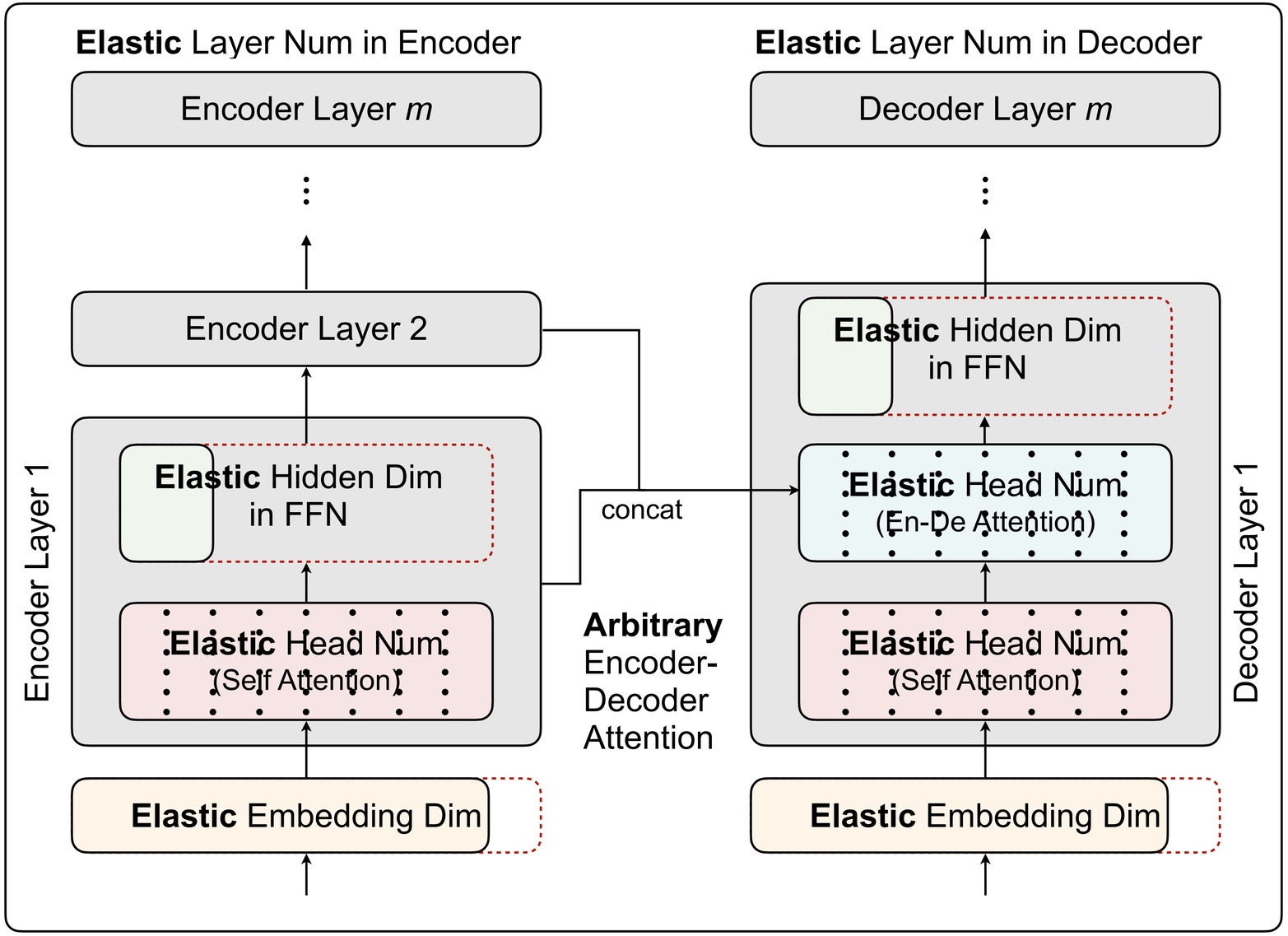


Elastic SuperTransformer with Weight-Sharing

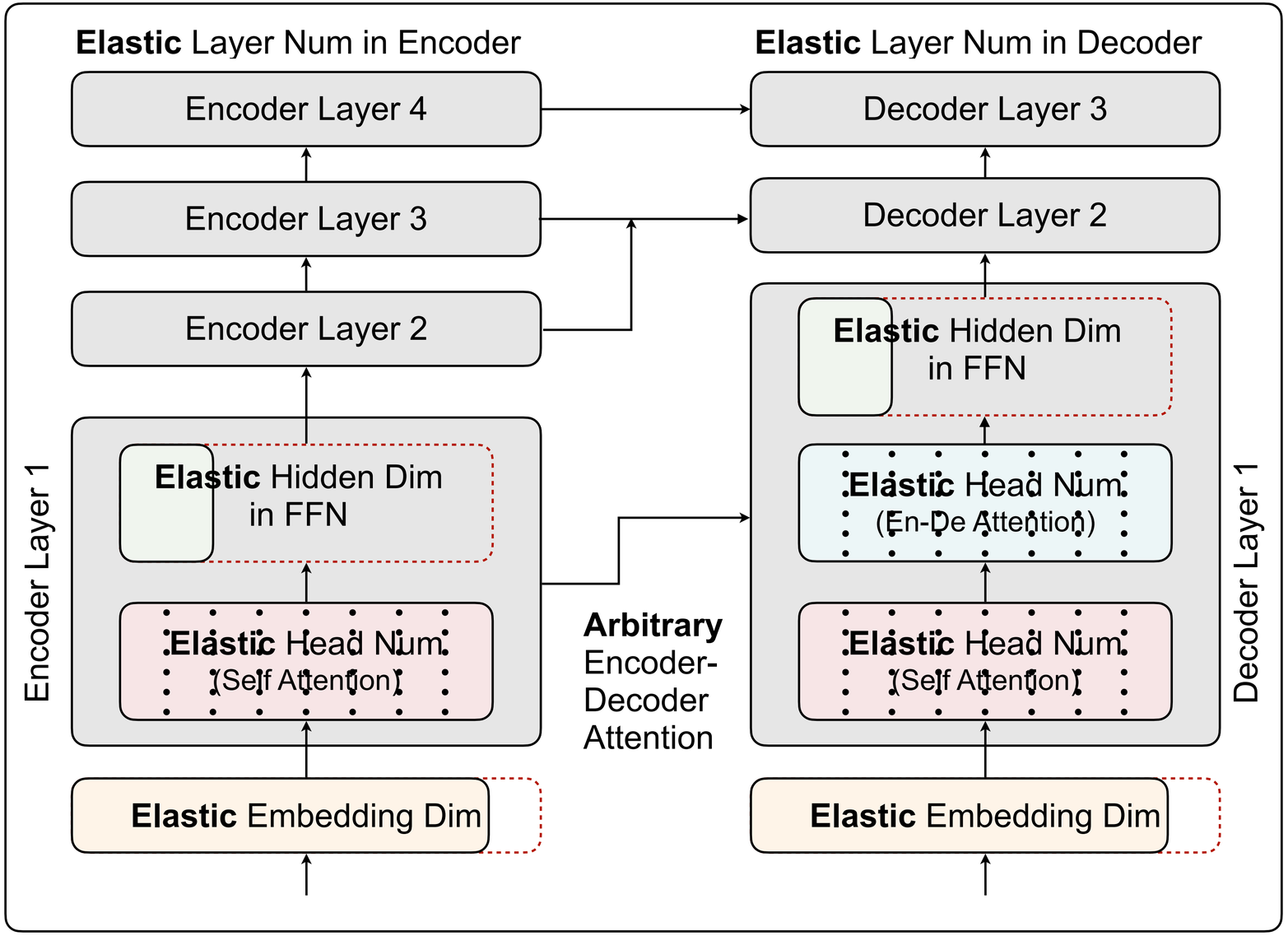
- Elastic number of layers, share the front several layers
- Elastic hidden dimension, share the front part of weights
- Elastic head number in all self-attention and cross-attention, share Q,K,V
- Elastic embedding dimension, share the front part of embedding



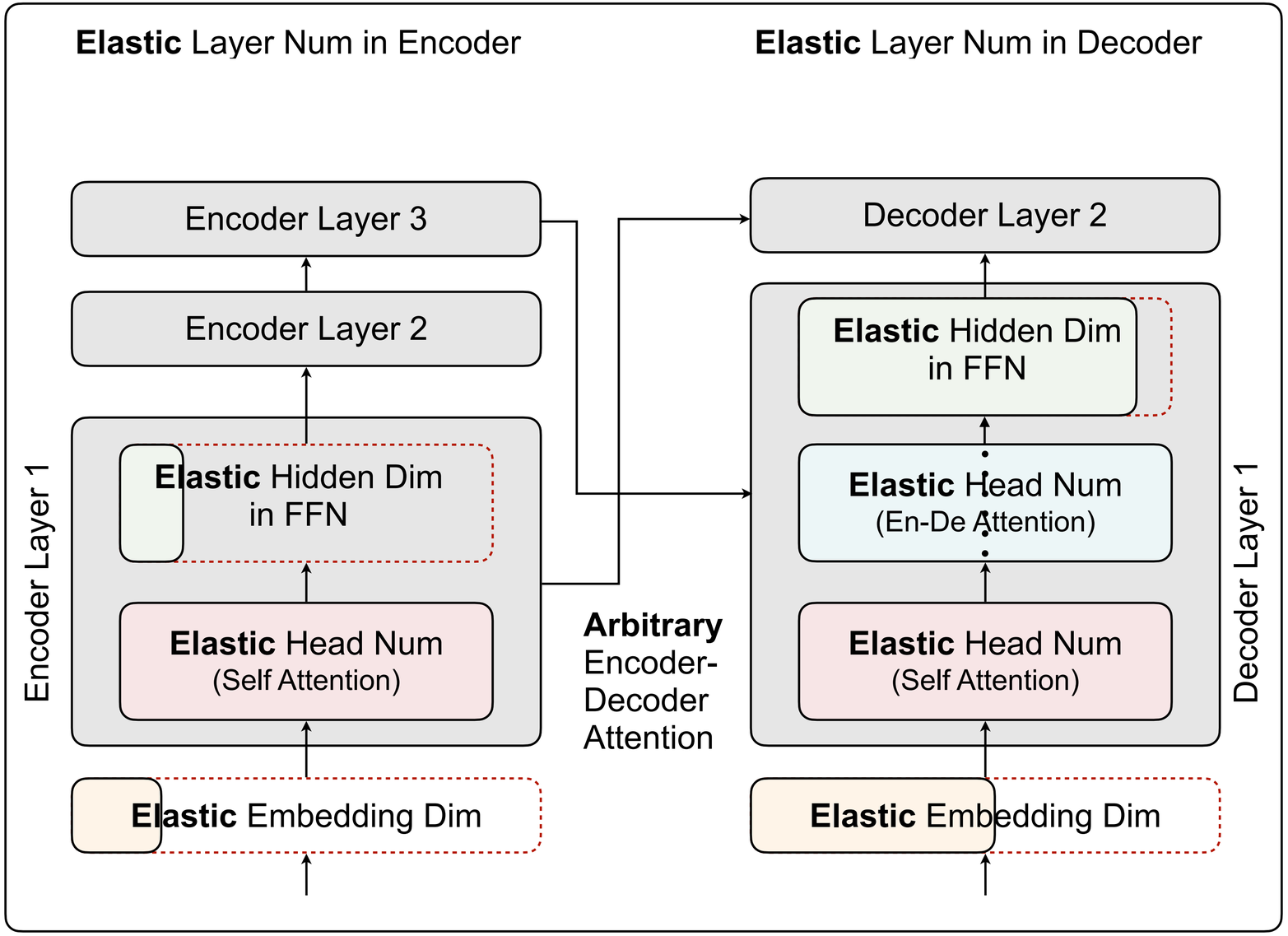
Train SuperTransformer by Uniform Sampling



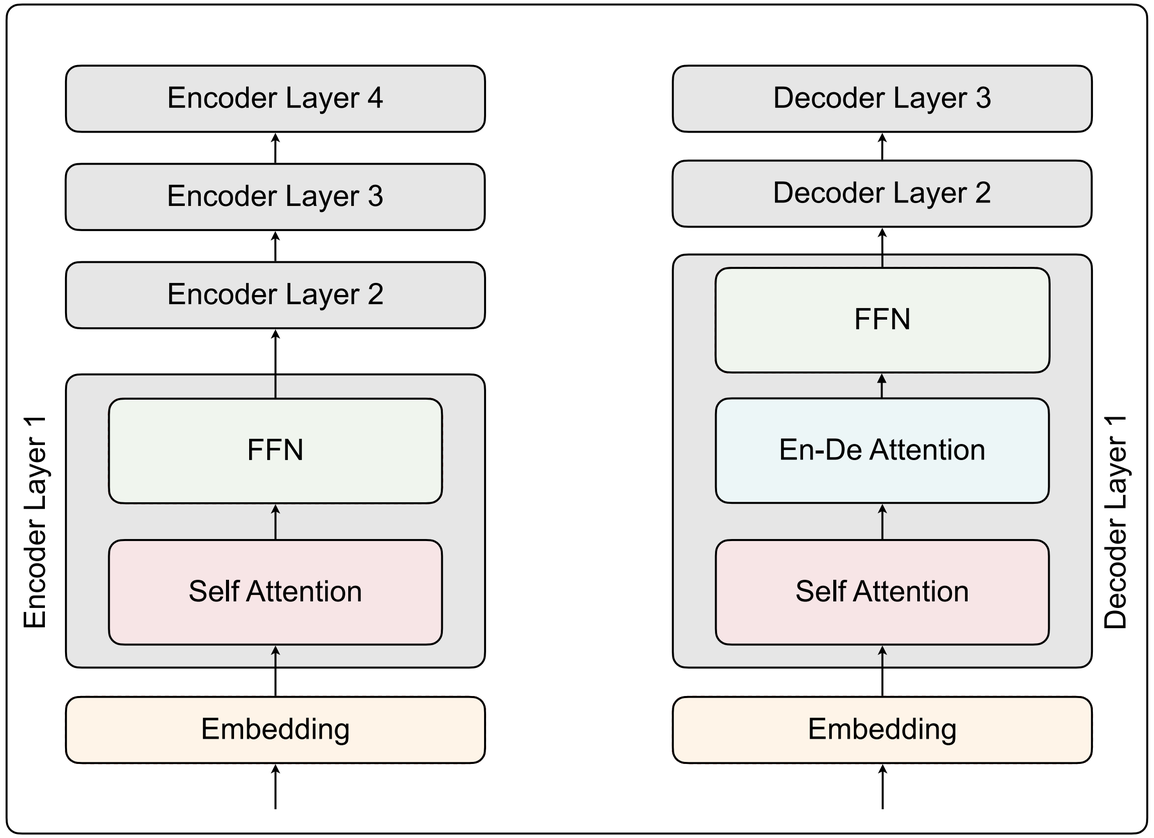
Train SuperTransformer by Uniform Sampling



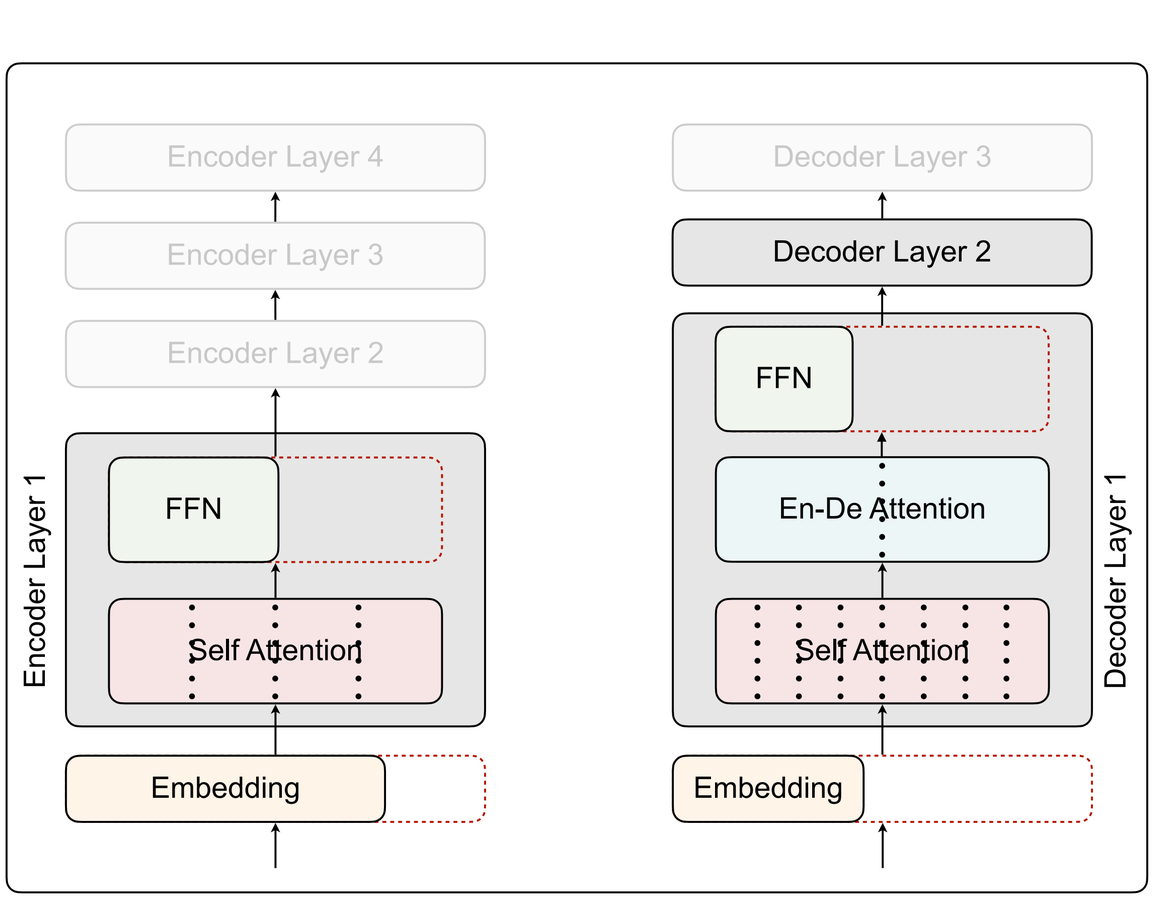
Train SuperTransformer by Uniform Sampling

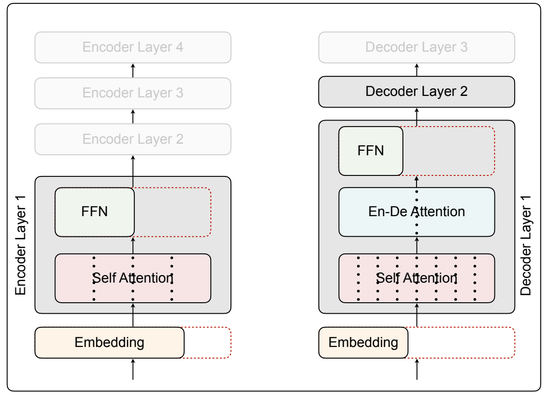


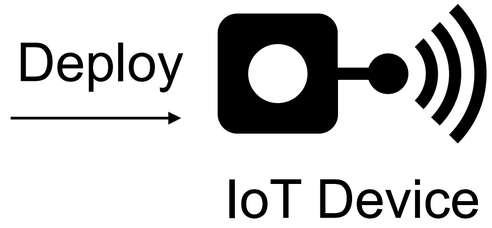
Specialized SubTransformer for Different Hardware



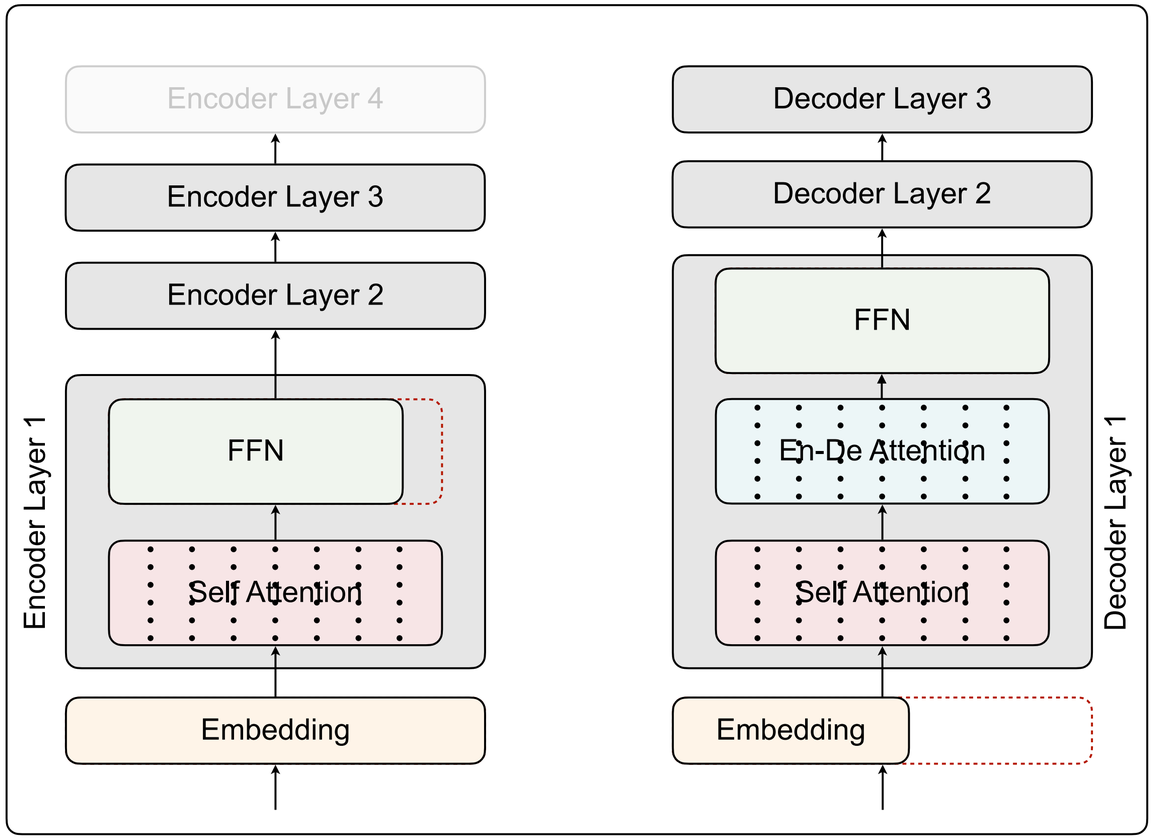
Specialized SubTransformer for Different Hardware

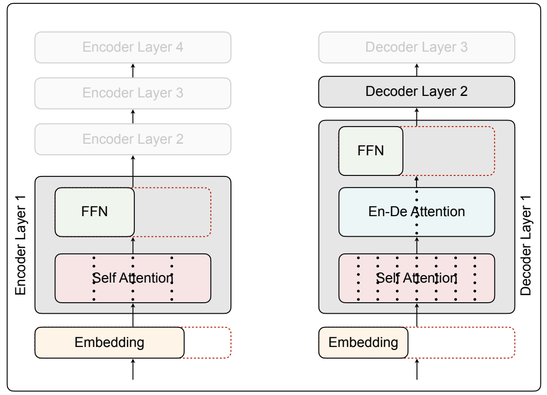




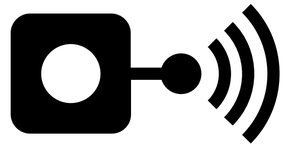


Specialized SubTransformer for Different Hardware

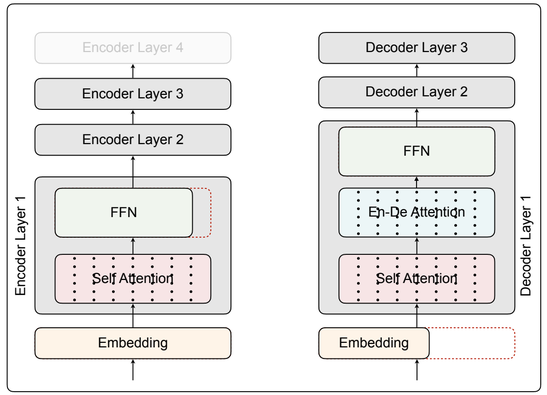




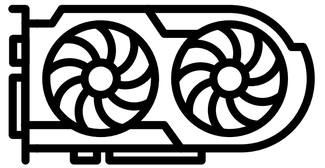
Deploy



IoT Device



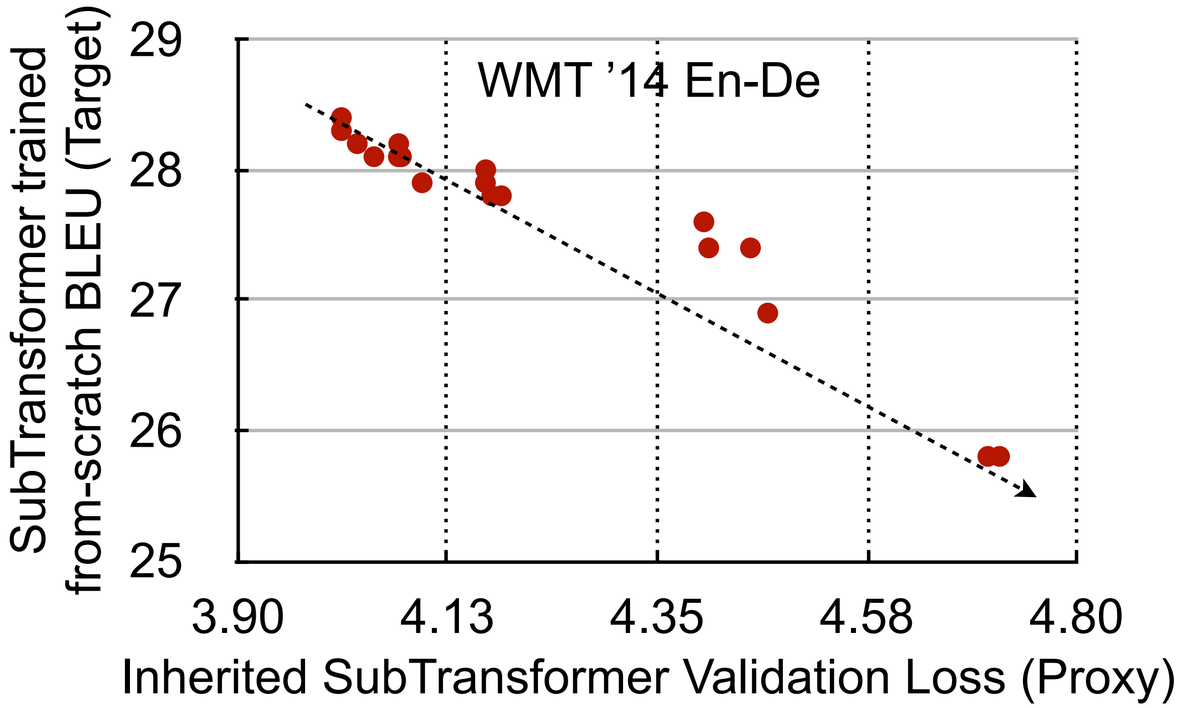
Deploy



GPU

SuperTransformer Provides Accurate Performance Proxy

- SuperTransformer provides fast and accurate proxy of SubTransformer performance
  - The smaller the validation loss of a inherited-weight SubTransformers, the better the final performance trained from-scratch



Three Steps in HAT

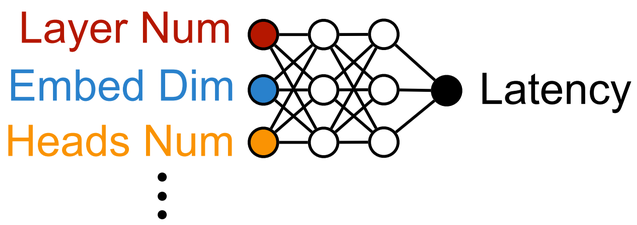
- Train a SuperTransformer
- Evolutionary search with a hardware latency constraint to find a SubTransformer
- Finally, train the searched SubTransformer from scratch

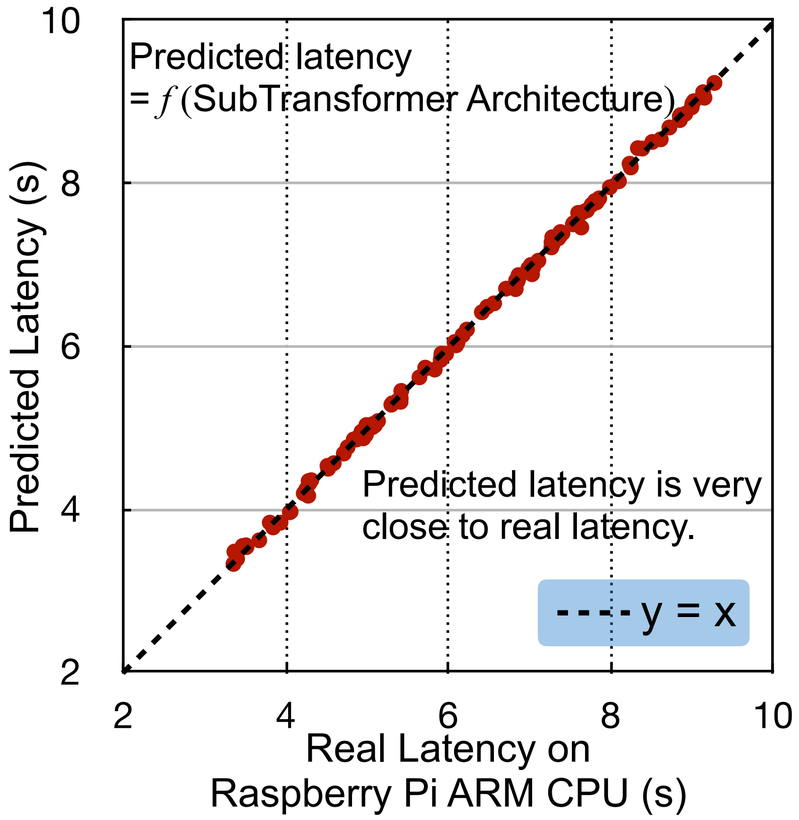
Evolutionary Search with Hardware Latency Feedback

- Search for a model with low loss and satisfies latency constraint

Latency Predictor

- Train a latency predictor to provide fast latency feedback
- With a dataset of [SubTransformer architecture, measured latency]
- Accurate: RMSR is 0.1s



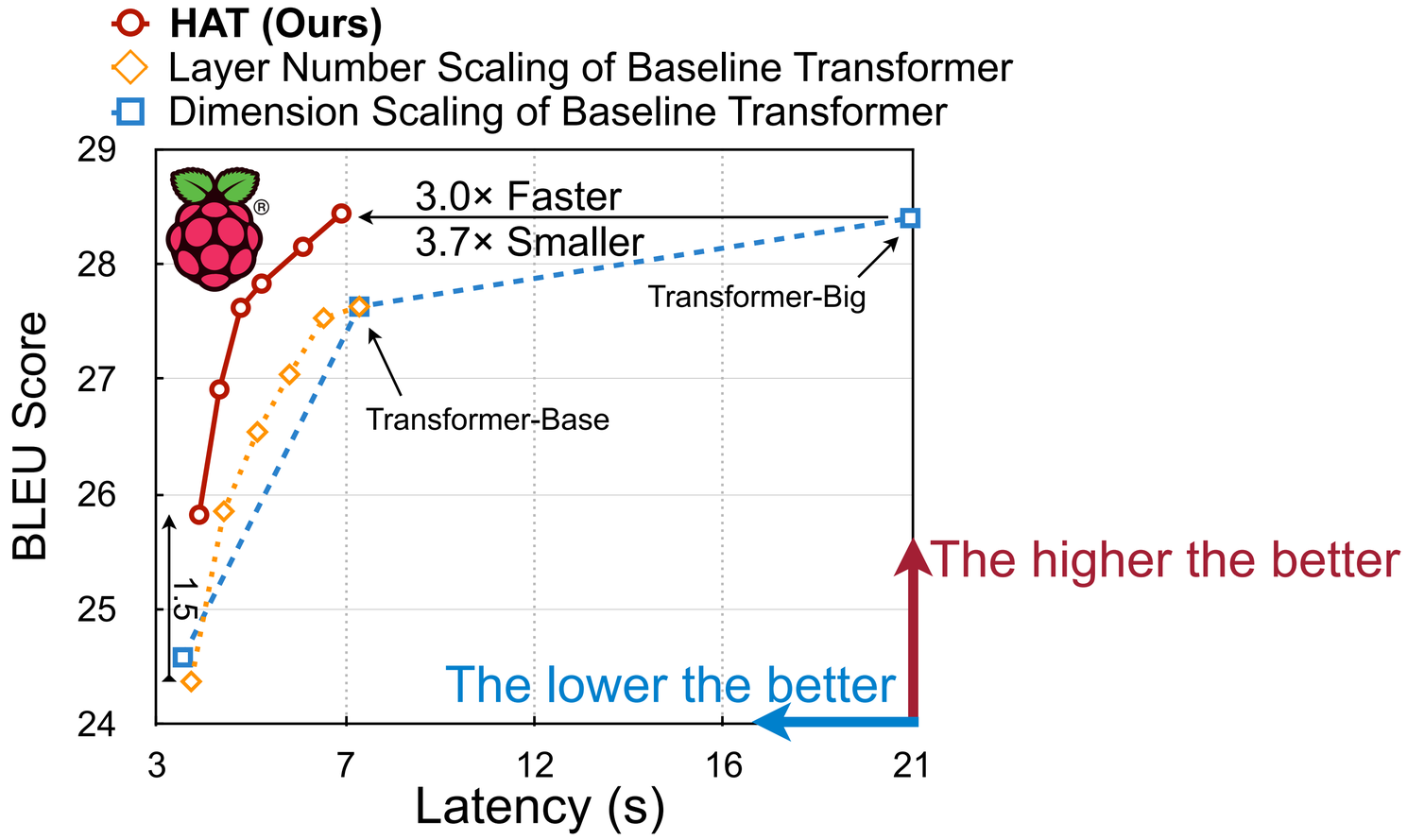


Three Steps in HAT

- Train a SuperTransformer
- Evolutionary search with a hardware latency constraint to find a SubTransformer
- Finally, train the searched SubTransformer from scratch

Comparison with Baseline Transformer

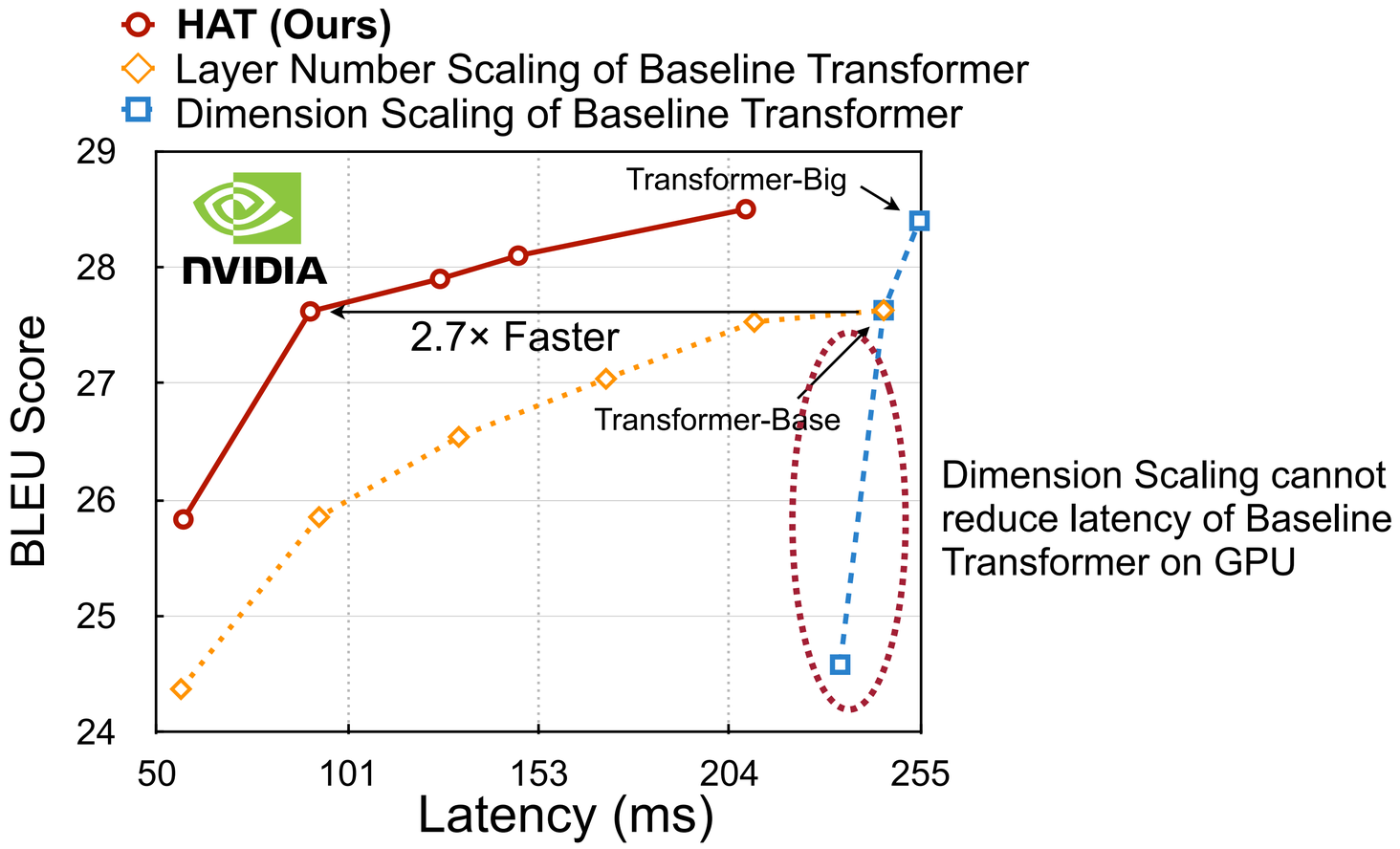
WMT'14 En-De Running on a Raspberry Pi



- HAT achieves 3x faster and 3.7x smaller size over baseline Transformer

Comparison with Baseline Transformer

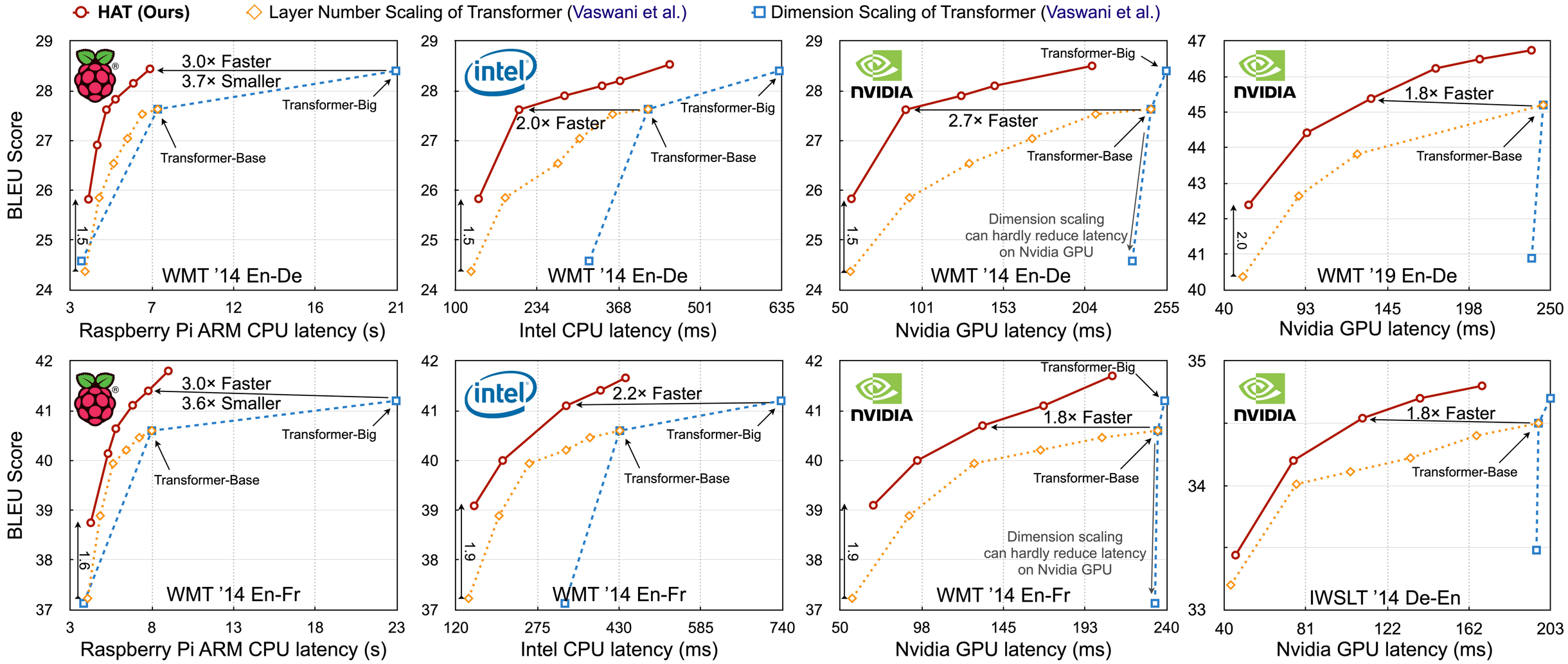
WMT'14 En-De Running on a Nvidia TITAN Xp GPU



Dimension Scaling cannot reduce latency of Baseline Transformer on GPU

- HAT achieves 2.7x speedup

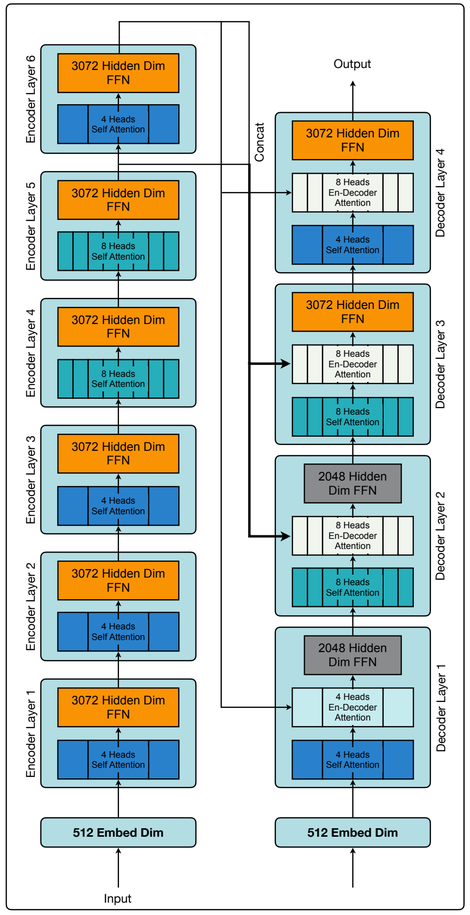
Comparison with Baseline Transformer



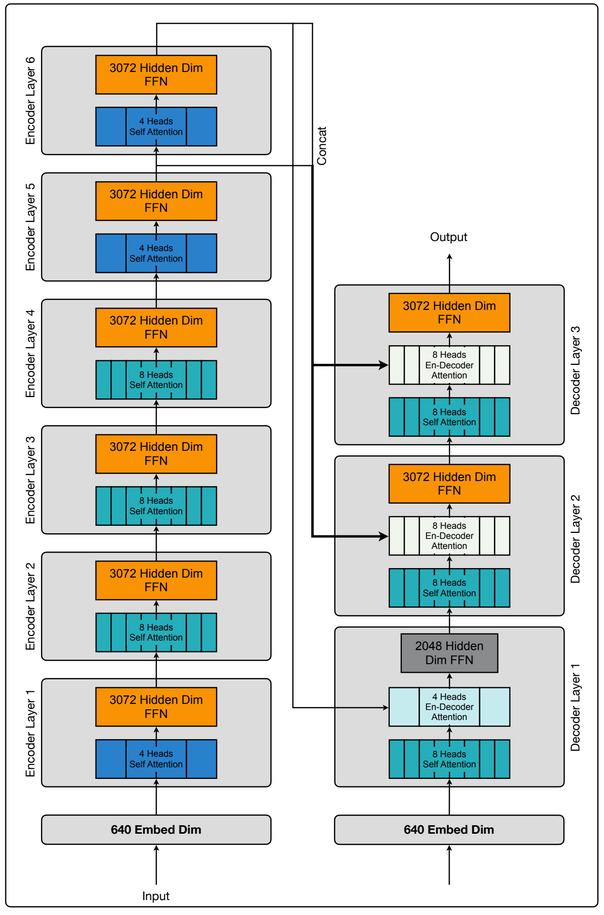
- HAT outperforms baseline Transformer on diverse hardware platforms

HAT Searches Specialized Models

On WMT'14 En-De Task



SubTransformer for ARM CPU



SubTransformer for Nvidia GPU

- Searched SubTransformers for ARM CPU and Nvidia GPU, both have 28.1 BLEU

HAT Searches Specialized Models

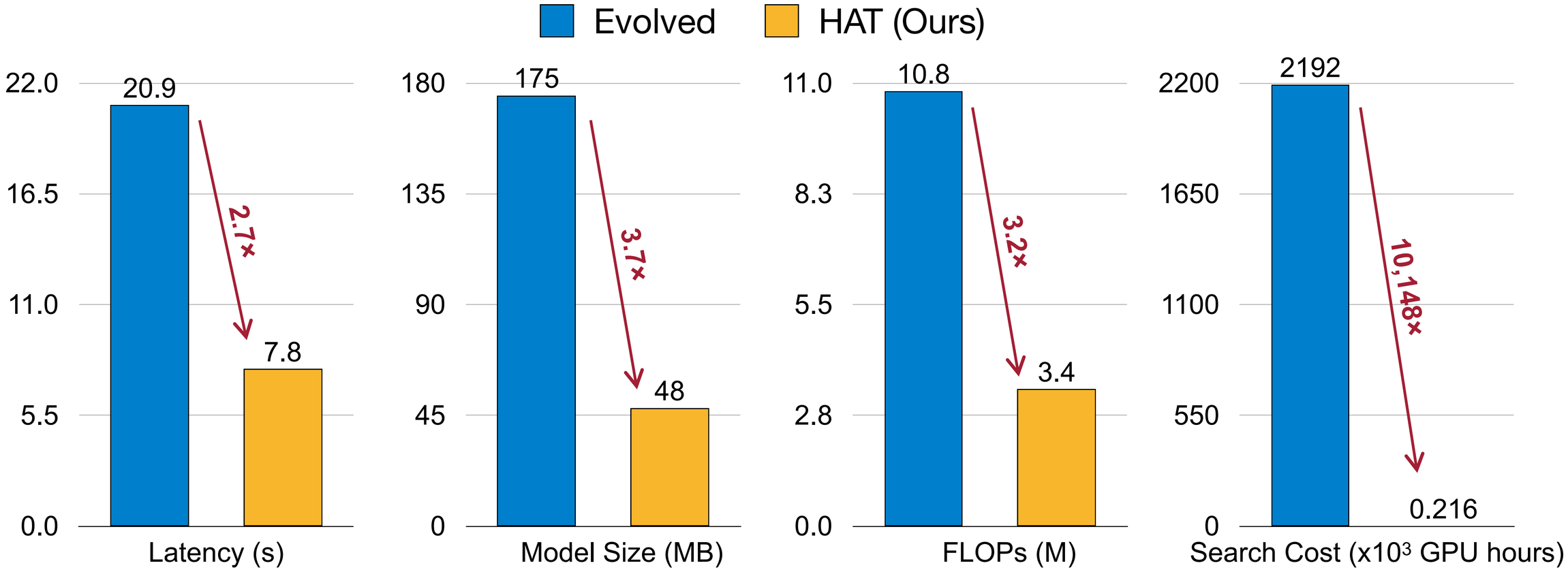
On WMT'14 En-De Task

	BLEU	GPU Latency	ARM CPU Latency
GPU-Efficient Model	28.1	**147ms**	6491ms
ARM-Efficient Model	28.1	184ms	**6042ms**

- The efficient model for GPU is not efficient for ARM CPU and vice versa
- Specialized model is necessary

Comparison with the Evolved Transformer

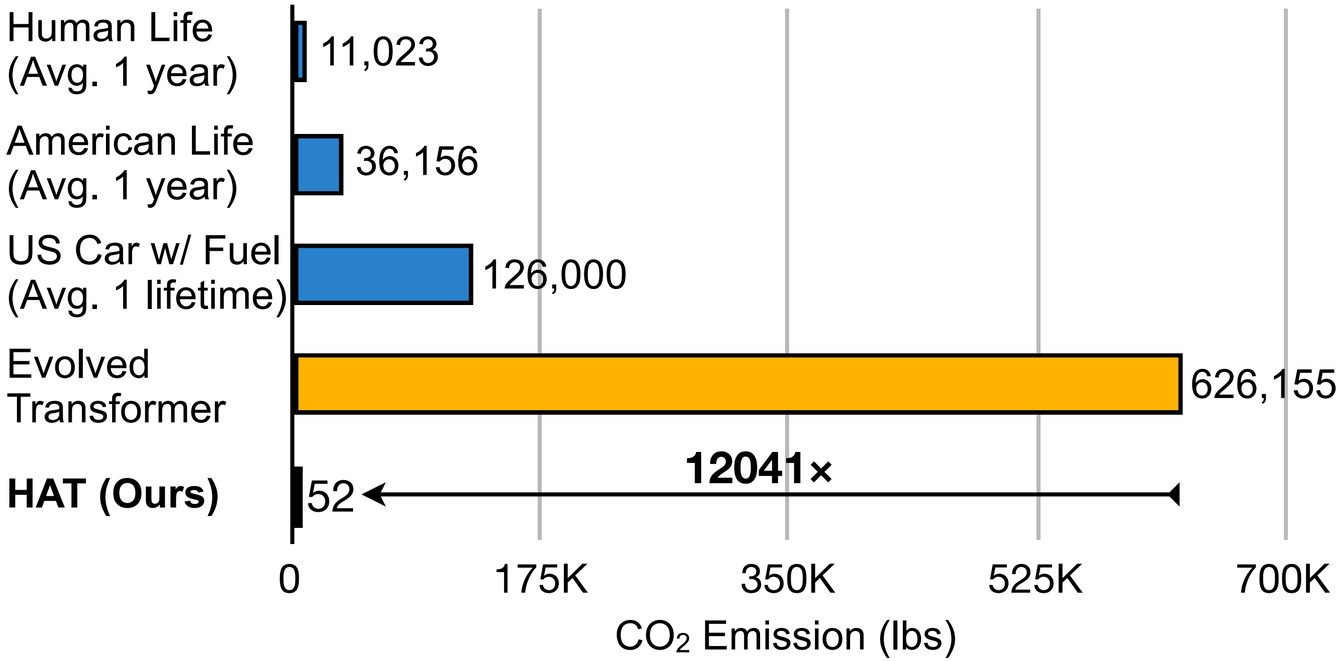
WMT'14 En-Fr running on a Raspberry Pi



- For WMT'14 En-Fr running on Raspberry Pi, HAT achieves 0.1 higher BLEU, 2.7x faster, 3.7x smaller model size, 3.2x fewer FLOPs, and 10,148x less search cost

Comparison with the Evolved Transformer

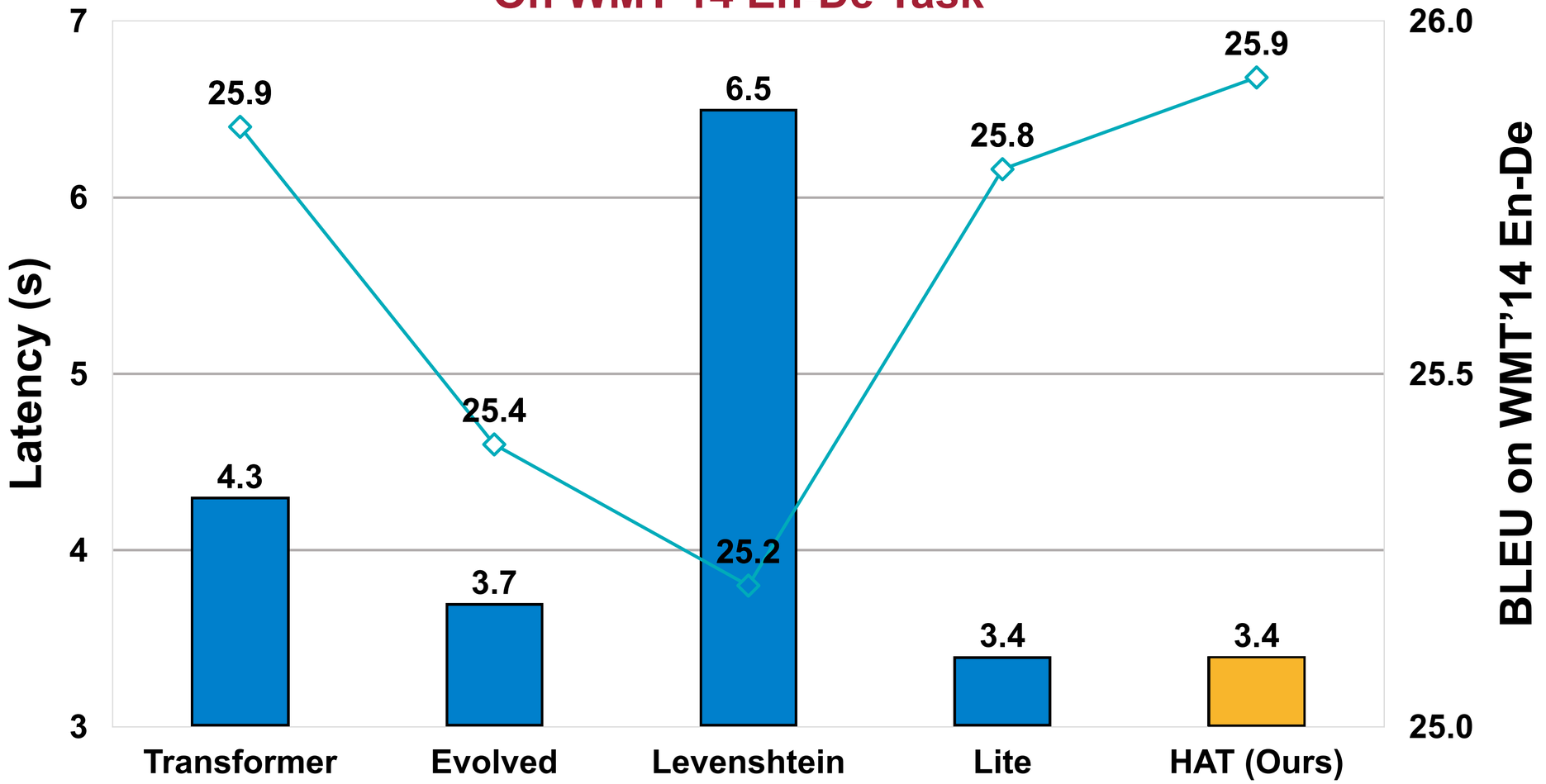
On WMT'14 En-De Task



- For WMT'14 En-De, the search cost of HAT is 12,000x less than the Evolved Transformer
- HAT is cost-efficient because in evolutionary search, it leverages the performance proxy instead of performance trained to the end

Comparison with Other Models

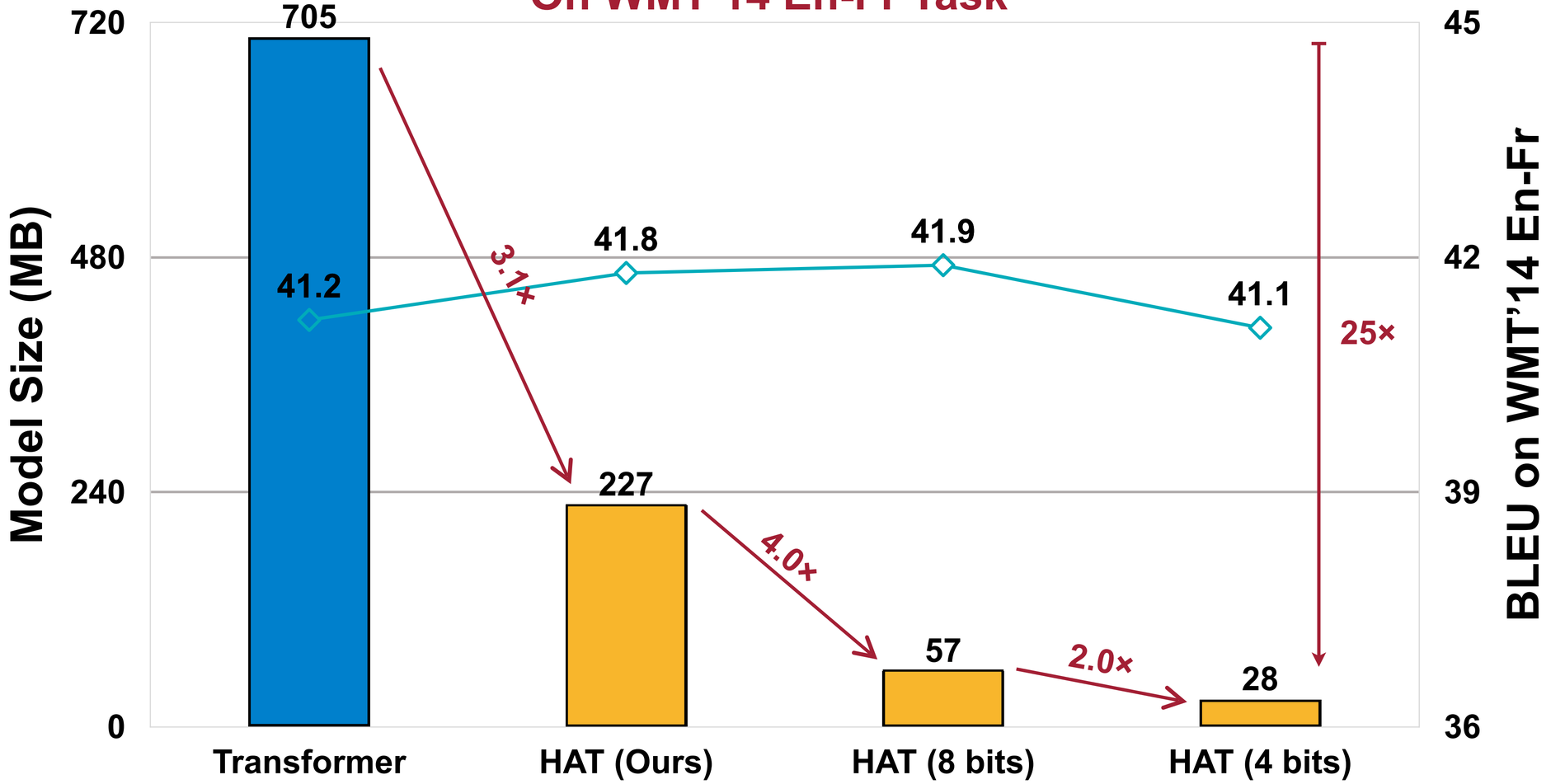
On WMT'14 En-De Task



- HAT has lowest latency and highest BLEU
- HAT is orthogonal to new operations

Further Compress HAT by 25x

On WMT'14 En-Fr Task



- HAT is orthogonal to general model compression techniques

Open-Source

- Code and 50 pre-trained models are released
- Latency, BLEU, model size, FLOPs are provided
- Push-the-button to run models

https://github.com/mit-han-lab/hardware-aware-transformers

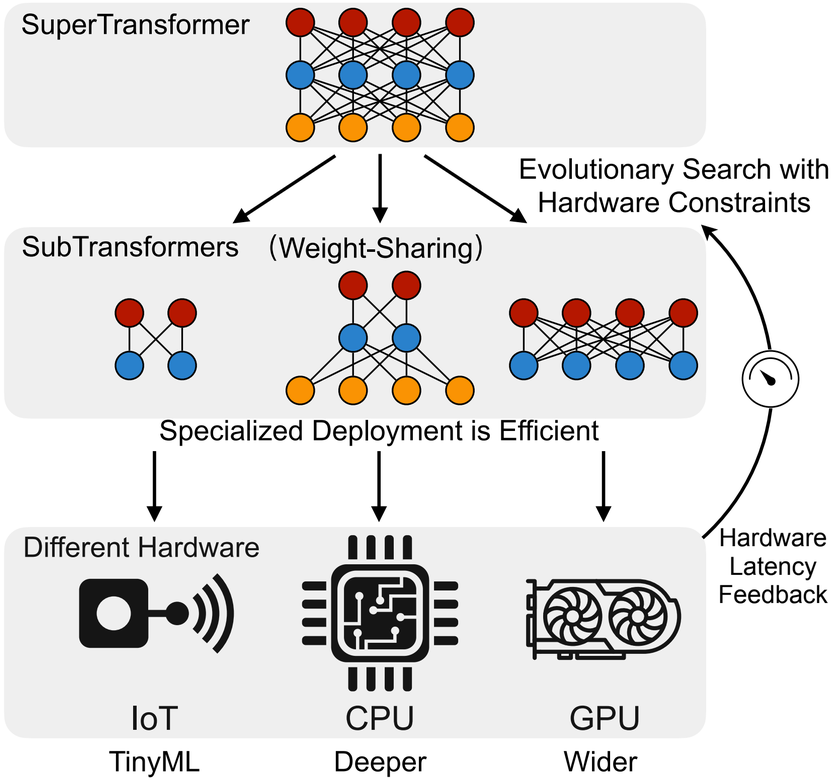
Task	Hardware	Latency	#Params (M)	FLOPs (G)	BLEU	Sacre BLEU	model_name	Link
WMT'14 En-De	Raspberry Pi ARM Cortex-A72 CPU	3.5s	25.22	1.53	25.8	25.6	HAT_wmt14ende_raspberrypi@3.5s_bleu@25.8	link
		4.0s	29.42	1.78	26.9	26.6	HAT_wmt14ende_raspberrypi@4.0s_bleu@26.9	link
		4.5s	35.72	2.19	27.6	27.1	HAT_wmt14ende_raspberrypi@4.5s_bleu@27.6	link
		5.0s	36.77	2.26	27.8	27.2	HAT_wmt14ende_raspberrypi@5.0s_bleu@27.8	link
		6.0s	44.13	2.70	28.2	27.6	HAT_wmt14ende_raspberrypi@6.0s_bleu@28.2	link
		6.9s	48.33	3.02	28.4	27.8	HAT_wmt14ende_raspberrypi@6.9s_bleu@28.4	link
WMT'14 En-De	Intel Xeon E5-2640 CPU	137.9ms	30.47	1.87	25.8	25.6	HAT_wmt14ende_xeon@137.9ms_bleu@25.8	link
		204.2ms	35.72	2.19	27.6	27.1	HAT_wmt14ende_xeon@204.2ms_bleu@27.6	link
		278.7ms	40.97	2.54	27.9	27.3	HAT_wmt14ende_xeon@278.7ms_bleu@27.9	link
		340.2ms	46.23	2.86	28.1	27.5	HAT_wmt14ende_xeon@340.2ms_bleu@28.1	link
		369.6ms	51.48	3.21	28.2	27.6	HAT_wmt14ende_xeon@369.6ms_bleu@28.2	link
		450.9ms	56.73	3.53	28.5	27.9	HAT_wmt14ende_xeon@450.9ms_bleu@28.5	link
WMT'14 En-De	Nvidia TITAN Xp GPU	57.1ms	30.47	1.87	25.8	25.6	HAT_wmt14ende_titanxp@57.1ms_bleu@25.8	link
		91.2ms	35.72	2.19	27.6	27.1	HAT_wmt14ende_titanxp@91.2ms_bleu@27.6	link
		126.0ms	40.97	2.54	27.9	27.3	HAT_wmt14ende_titanxp@126.0ms_bleu@27.9	link
		146.7ms	51.20	3.17	28.1	27.5	HAT_wmt14ende_titanxp@146.7ms_bleu@28.1	link
		208.1ms	49.38	3.09	28.5	27.8	HAT_wmt14ende_titanxp@208.1ms_bleu@28.5	link
WMT'14 En-Fr	Raspberry Pi ARM Cortex-A72 CPU	4.3s	25.22	1.53	38.8	36.0	HAT_wmt14enfr_raspberrypi@4.3s_bleu@38.8	link
		5.3s	35.72	2.23	40.1	37.3	HAT_wmt14enfr_raspberrypi@5.3s_bleu@40.1	link
		5.8s	36.77	2.26	40.6	37.8	HAT_wmt14enfr_raspberrypi@5.8s_bleu@40.6	link
		6.9s	44.13	2.70	41.1	38.3	HAT_wmt14enfr_raspberrypi@6.9s_bleu@41.1	link
		7.8s	49.38	3.09	41.4	38.5	HAT_wmt14enfr_raspberrypi@7.8s_bleu@41.4	link
		9.1s	56.73	3.57	41.8	38.9	HAT_wmt14enfr_raspberrypi@9.1s_bleu@41.8	link
WMT'14 En-Fr	Intel Xeon E5-2640 CPU	154.7ms	30.47	1.84	39.1	36.3	HAT_wmt14enfr_xeon@154.7ms_bleu@39.1	link
		208.8ms	35.72	2.23	40.0	37.2	HAT_wmt14enfr_xeon@208.8ms_bleu@40.0	link
		329.4ms	44.13	2.70	41.1	38.2	HAT_wmt14enfr_xeon@329.4ms_bleu@41.1	link
		394.5ms	51.48	3.28	41.4	38.5	HAT_wmt14enfr_xeon@394.5ms_bleu@41.4	link
		442.0ms	56.73	3.57	41.7	38.8	HAT_wmt14enfr_xeon@442.0ms_bleu@41.7	link
WMT'14 En-Fr	Nvidia TITAN Xp GPU	69.3ms	30.47	1.84	39.1	36.3	HAT_wmt14enfr_titanxp@69.3ms_bleu@39.1	link
		94.9ms	35.72	2.23	40.0	37.2	HAT_wmt14enfr_titanxp@94.9ms_bleu@40.0	link
		132.9ms	40.97	2.51	40.7	37.8	HAT_wmt14enfr_titanxp@132.9ms_bleu@40.7	link
		168.3ms	46.23	2.90	41.1	38.3	HAT_wmt14enfr_titanxp@168.3ms_bleu@41.1	link
		208.3ms	51.48	3.25	41.7	38.8	HAT_wmt14enfr_titanxp@208.3ms_bleu@41.7	link
WMT'19 En-De	Nvidia TITAN Xp GPU	55.7ms	36.89	2.27	42.4	41.9	HAT_wmt19ende_titanxp@55.7ms_bleu@42.4	link
		93.2ms	42.28	2.63	44.4	43.9	HAT_wmt19ende_titanxp@93.2ms_bleu@44.4	link
		134.5ms	40.97	2.54	45.4	44.7	HAT_wmt19ende_titanxp@134.5ms_bleu@45.4	link
		176.1ms	46.23	2.86	46.2	45.6	HAT_wmt19ende_titanxp@176.1ms_bleu@46.2	link
		204.5ms	51.48	3.18	46.5	45.7	HAT_wmt19ende_titanxp@204.5ms_bleu@46.5	link
		237.8ms	56.73	3.53	46.7	46.0	HAT_wmt19ende_titanxp@237.8ms_bleu@46.7	link
IWSLT'14 De-En	Nvidia TITAN Xp GPU	45.6ms	16.82	0.78	33.4	32.5	HAT_iwslt14deen_titanxp@45.6ms_bleu@33.4	link
		74.5ms	19.98	0.93	34.2	33.3	HAT_iwslt14deen_titanxp@74.5ms_bleu@34.2	link
		109.0ms	23.13	1.13	34.5	33.6	HAT_iwslt14deen_titanxp@109.0ms_bleu@34.5	link
		137.8ms	27.33	1.32	34.7	33.8	HAT_iwslt14deen_titanxp@137.8ms_bleu@34.7	link
		168.8ms	31.54	1.52	34.8	33.9	HAT_iwslt14deen_titanxp@168.8ms_bleu@34.8	link

HAT: Hardware-Aware Transformers

Pushing the frontier of Green AI and Tiny AI

https://hat.mit.edu

- A specialized transformer for each hardware
- Arbitrary encoder-decoder attention and heterogeneous layers improve performance
- 3x speedup, 3.7x smaller size, 12,000x less cost over baselines



Live Q&A:

Wed. July 8, 13:00 UTC+0 13B Machine Translation-15

Wed. July 8, 21:00 UTC+0 15B Machine Translation-18

Paper ID: 148