Carnegie Mellon University
Language Technologies Institute

Lecture 5: Benchmarking

What should we measure? What can we measure?

Emma Strubell & Yonatan Bisk

Lecture Highlight!

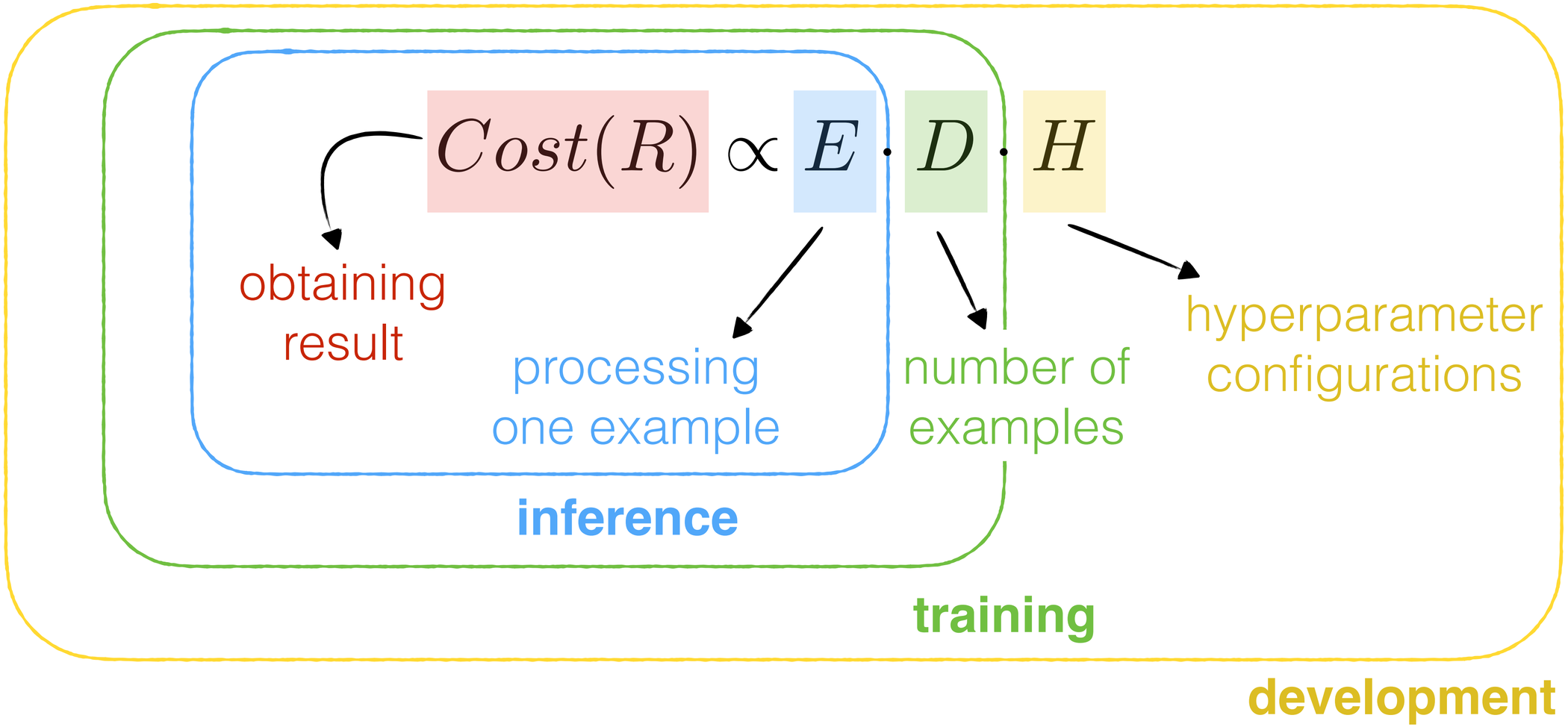
¿que es esto?

Lightweight. Each point below should only be a sentence or two. We are looking to see that you actually thought a bit and engaged with the content; try to avoid perfunctory answers (e.g. if you could give the same answer for “why is this important” for any lecture in this class, it’s probably not a good answer.)

- Why is this an important area of study? [0.5 points]
- Describe two different techniques/approaches discussed [0.5 points]
- Discuss relative strengths [0.5 points] and weaknesses [0.5 points] of the two techniques described above. [1 point total]

Q: What should we benchmark?

A: It depends. What is your target use case?



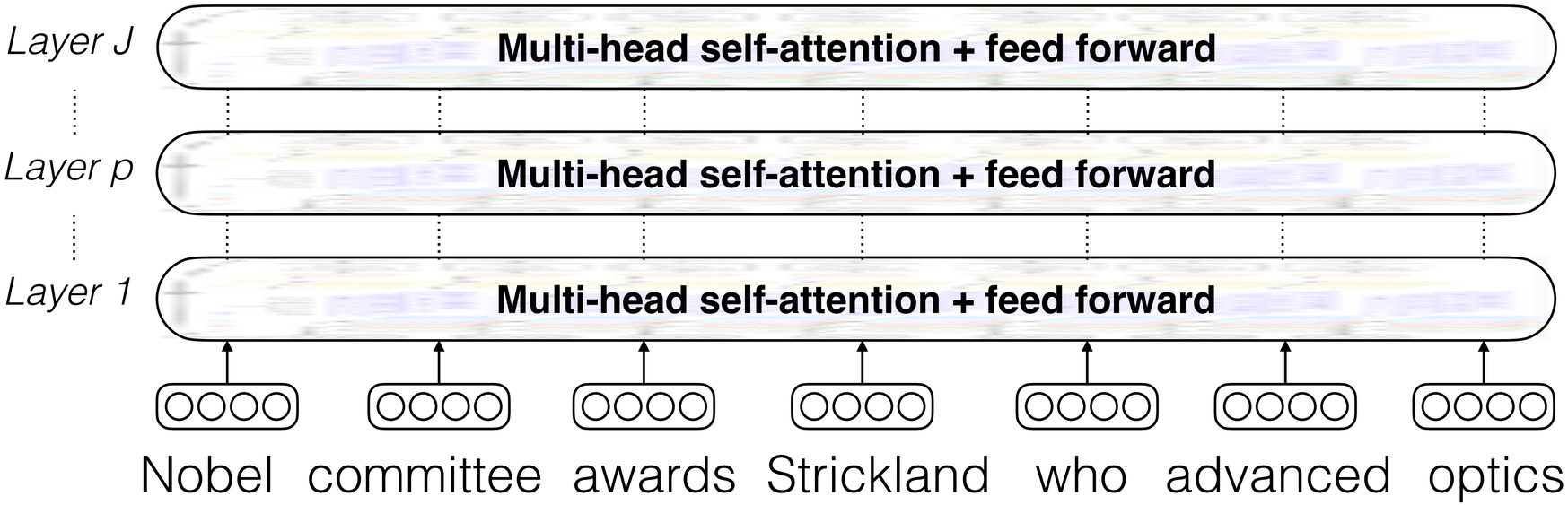
Equation: Schwartz, Dodge, Smith, Etzioni. Green AI. Communications of the ACM, December 2020.

What should we benchmark?

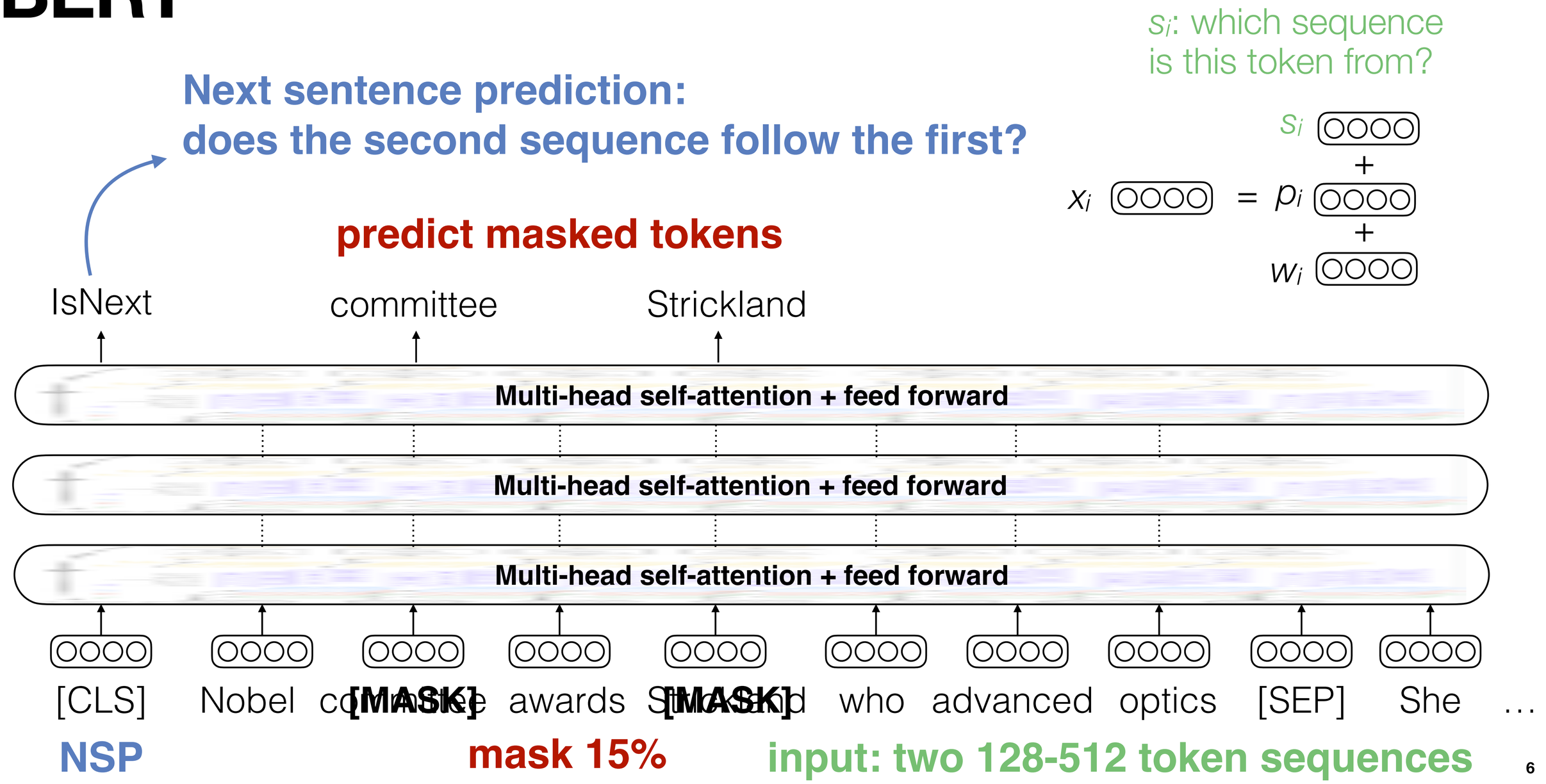
- Does *inference* efficiency come at the cost of *training* efficiency? *Tuning* efficiency?
  - What additional parameters does this approach add, and how fickle are they?
- Does *training* efficiency come at the cost of *fine-tuning* efficiency?

BERT

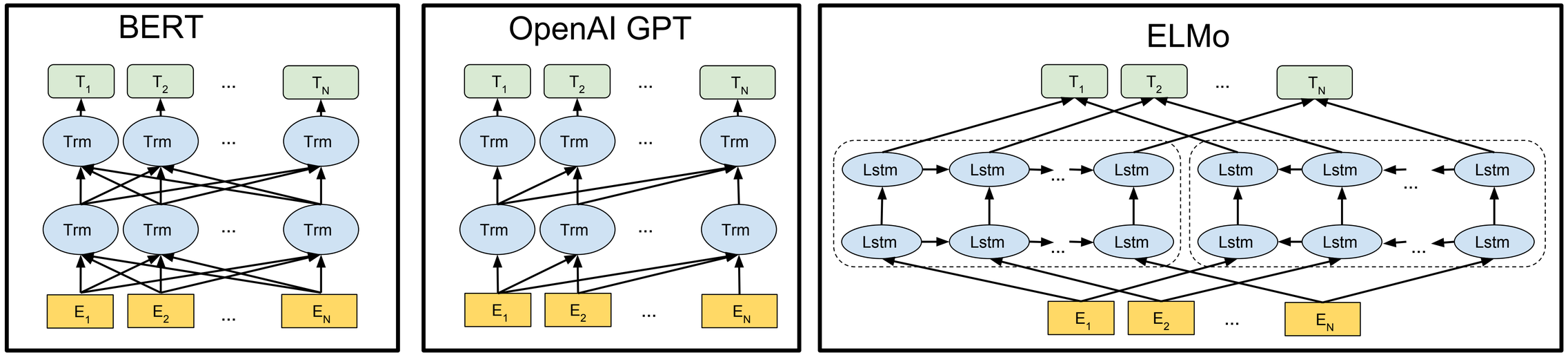
- Transformer language model trained with **non-autoregressive** masked language modeling (MLM) objective.



BERT


s_i: which sequence is this token from?
s_i
+
x_i = p_i
+
w_i
Next sentence prediction:
does the second sequence follow the first?
predict masked tokens
IsNext
committee
Strickland
Multi-head self-attention + feed forward
Multi-head self-attention + feed forward
Multi-head self-attention + feed forward
[CLS]
Nobel
[MASK]
awards
[MASK]
who
advanced
optics
[SEP]
She
…
NSP
mask 15%
input: two 128-512 token sequences

Other popular Transformer LMs



- GPT/XLNet
- RoBERTa
- ELECTRA

ELECTRA [Clark et al. 2020]

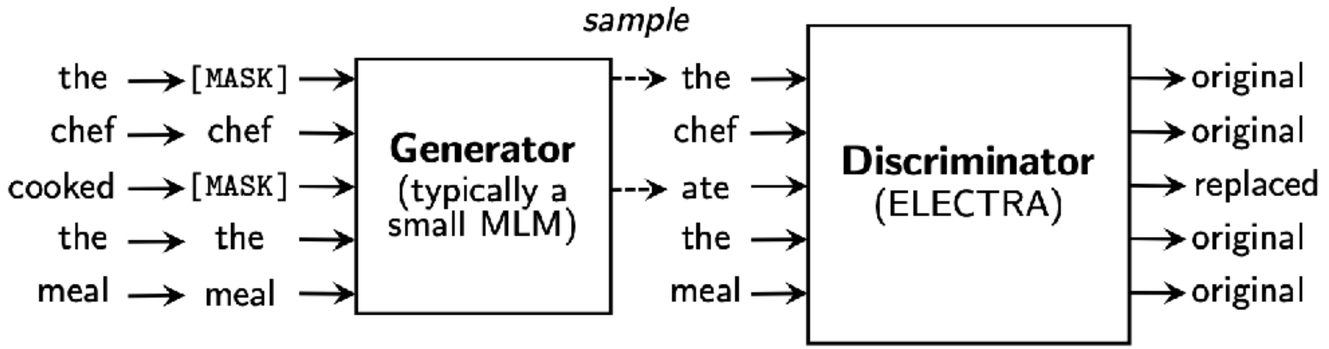
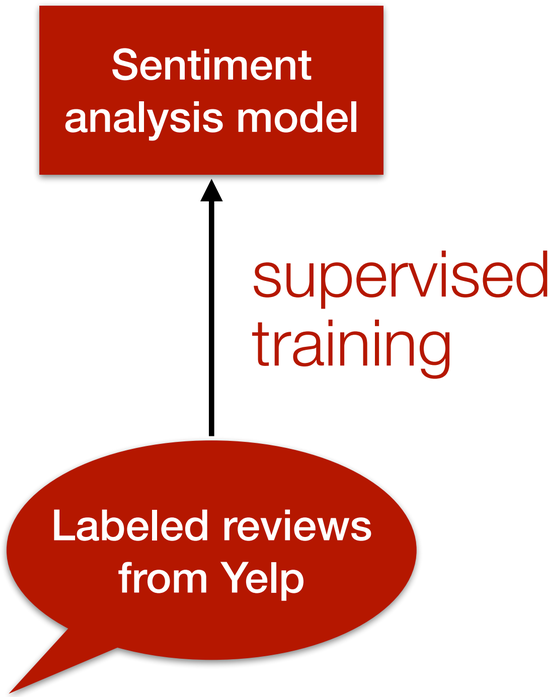


Figure from: Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ACL 2019.

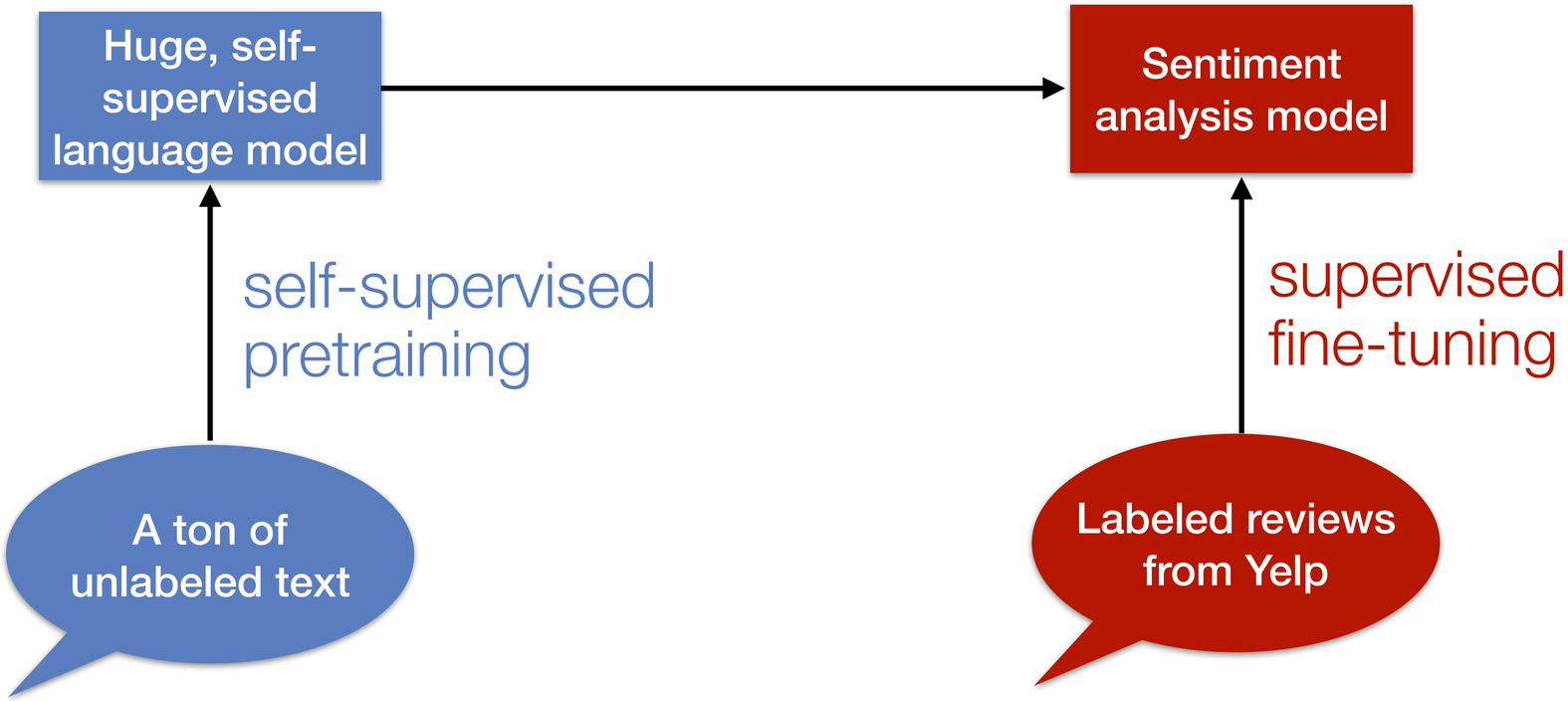
Transfer learning from language models

- Let's say I want to train a model for sentiment analysis.
- The old way: train the model using supervised learning on a corpus of labeled examples, e.g. Yelp reviews, IMDB movie reviews



Transfer learning from language models

- Let's say I want to train a model for sentiment analysis.
- The new way: **transfer learning** from a large language model trained using a self-supervised objective on a massive amount of unlabled text.



Case study: Evolved Transformer & Meena

- NAS to find Evolved Transformer at Google used 7.5 MWh energy.
- Evolved Transformer used as base architecture for Meena, 2.6B parameter chatbot.
- Meena energy savings = 15x NAS
- 2021: LaMDA
  2022: LaMDA 2
  2023: PaLM/Bard

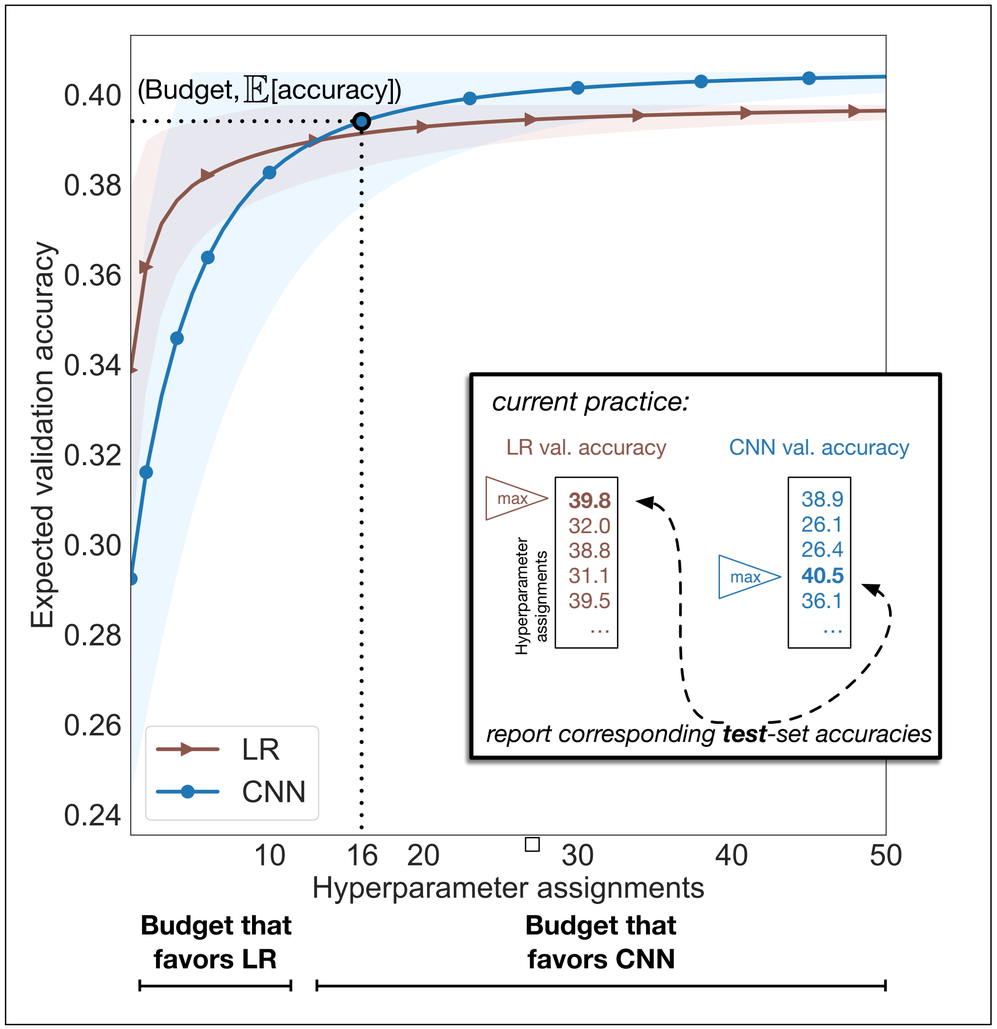
Jevons paradox: *technology increases the efficiency with which a resource is used, but the falling cost of use increases its demand—increasing, rather than reducing, resource use.*

https://blog.research.google/2020/01/towards-conversational-agent-that-can.html

D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. IEEE Computer, 2021.

What should we benchmark?

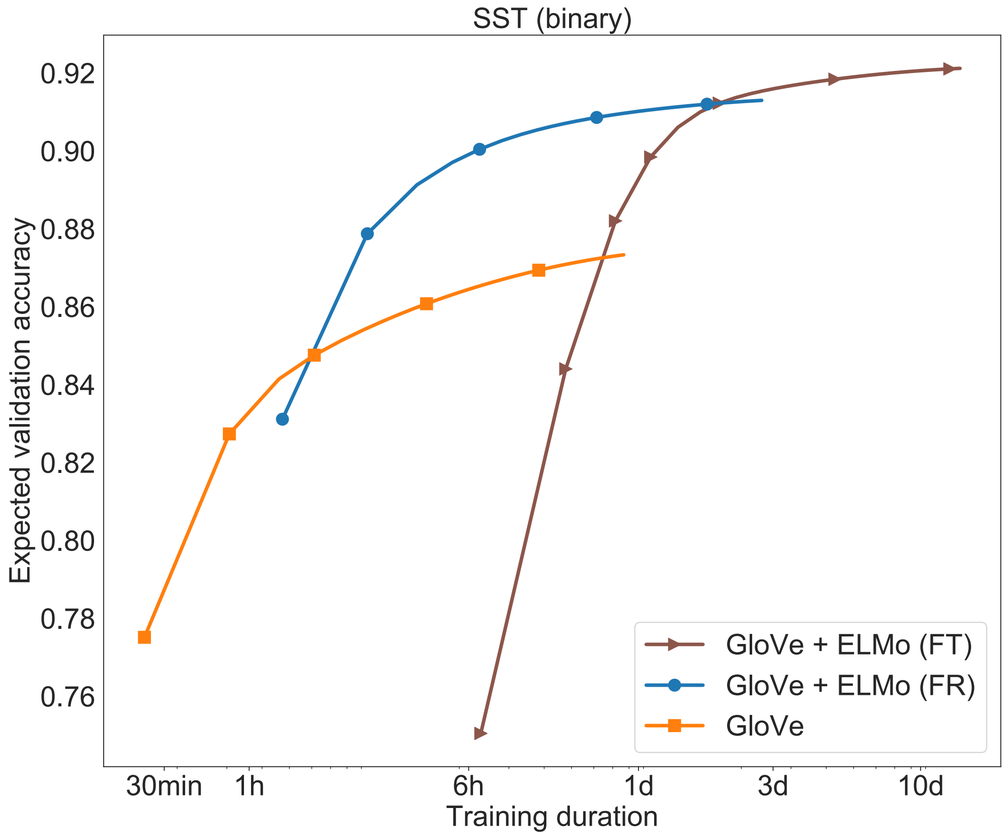
- Does *inference* efficiency come at the cost of *training* efficiency? *Tuning* efficiency?
- What additional parameters does this approach add, and how fickle are they?
- **Proposal:** Report *expected validation performance* (EVP) as a function of computation budget.
  - Budget (*B*): training time, # hyperparam trials
  - EVP: Estimate what the maximum validation performance would have been for $1 \leq n < B$ trials, in expectation.

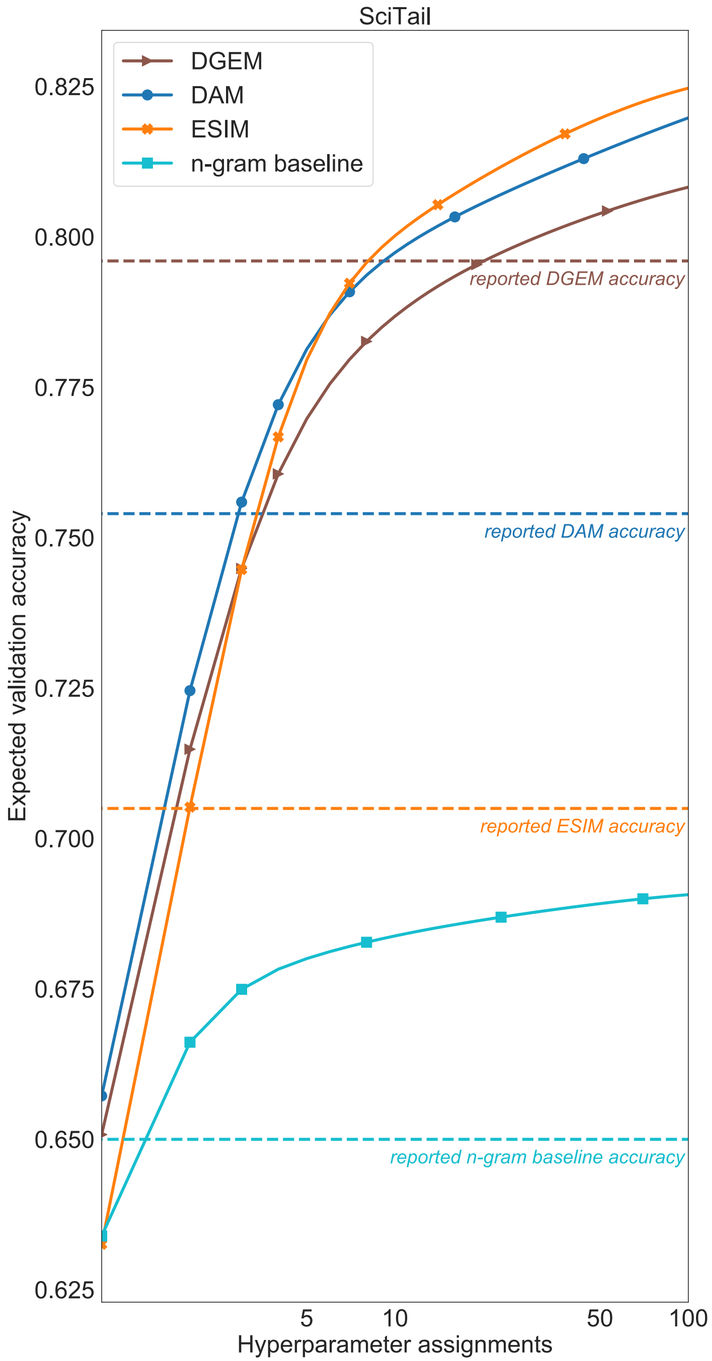


Dodge, Gururangan, Card, Schwartz, Smith. Show YourWork: Improved Reporting of Experimental Results. EMNLP 2019.

What should we benchmark?

- Report *expected validation performance* as a function of computation budget.





Dodge, Gururangan, Card, Schwartz, Smith. Show YourWork: Improved Reporting of Experimental Results. EMNLP 2019.

What should we benchmark?

- Does *inference* efficiency come at the cost of *training* efficiency? *Tuning* efficiency?
  - What additional parameters does this approach add, and how fickle are they?
- **Does training efficiency come at the cost of fine-tuning efficiency?**

HULK Efficiency Benchmark

Pre-training

Model	Hardware	Time	Cost	Params
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018)	4 TPU Pods	4 days	$1,728	108M
$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2018)	16 TPU Pods	4 days	$6,912	334M
$\text{XLNet}_{\text{BASE}}$ (Yang et al., 2019)	–	–	–	117M
$\text{XLNet}_{\text{LARGE}}$ (Yang et al., 2019)	512 TPU v3	2.5 days	$61,440	361M
$\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019)	1024 V100 GPUs	1 day	$75,203	125M
$\text{RoBERTa}_{\text{LARGE}}$ (Liu et al., 2019)	1024 V100 GPUs	1 day	$75,203	356M
$\text{ALBERT}_{\text{BASE}}$ (Lan et al., 2019)	64 TPU v3	–	–	12M
$\text{ALBERT}_{\text{LARGE}}$ (Lan et al., 2019)	–	–	–	18M
$\text{ALBERT}_{\text{XLARGE}}$ (Lan et al., 2019)	–	–	–	59M
$\text{ALBERT}_{\text{XXLARGE}}$ (Lan et al., 2019)	1024 TPU v3	32 hours	$65,536	223M
DistilBERT* (Sanh et al., 2019)	8×16G V100 GPU	90 hours	$2203.2	66M

Fine-tuning

Datasets	CoNLL 2003		SST-2		MNLI		
Model	Time	Score	Time	Score	Time	Score	Overall Score
$\text{BERT}_{\text{BASE}}$	43.43	2.08	207.15	0.45	N/A	0.00	2.53
$\text{BERT}_{\text{LARGE}}$	90.26	1.00	92.45	1.00	9,106.72	1.00	3.00
$\text{XLNet}_{\text{BASE}}$	67.14	1.34	102.45	0.90	7,704.71	1.18	3.42
$\text{XLNet}_{\text{LARGE}}$	243.00	0.37	367.11	0.25	939.62	9.69	10.31
$\text{RoBERTa}_{\text{BASE}}$	70.57	1.28	38.45	2.40	274.87	7.14	10.82
$\text{RoBERTa}_{\text{LARGE}}$	155.43	0.58	57.65	1.60	397.12	22.93	25.11
$\text{ALBERT}_{\text{BASE}}$	340.64	0.26	2,767.90	0.03	N/A	0.00	0.29
$\text{ALBERT}_{\text{LARGE}}$	844.85	0.11	3,708.49	0.02	N/A	0.00	0.13

Tables from: Zhou, et al. HULK: An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing. EACL 2021.

What should we benchmark?

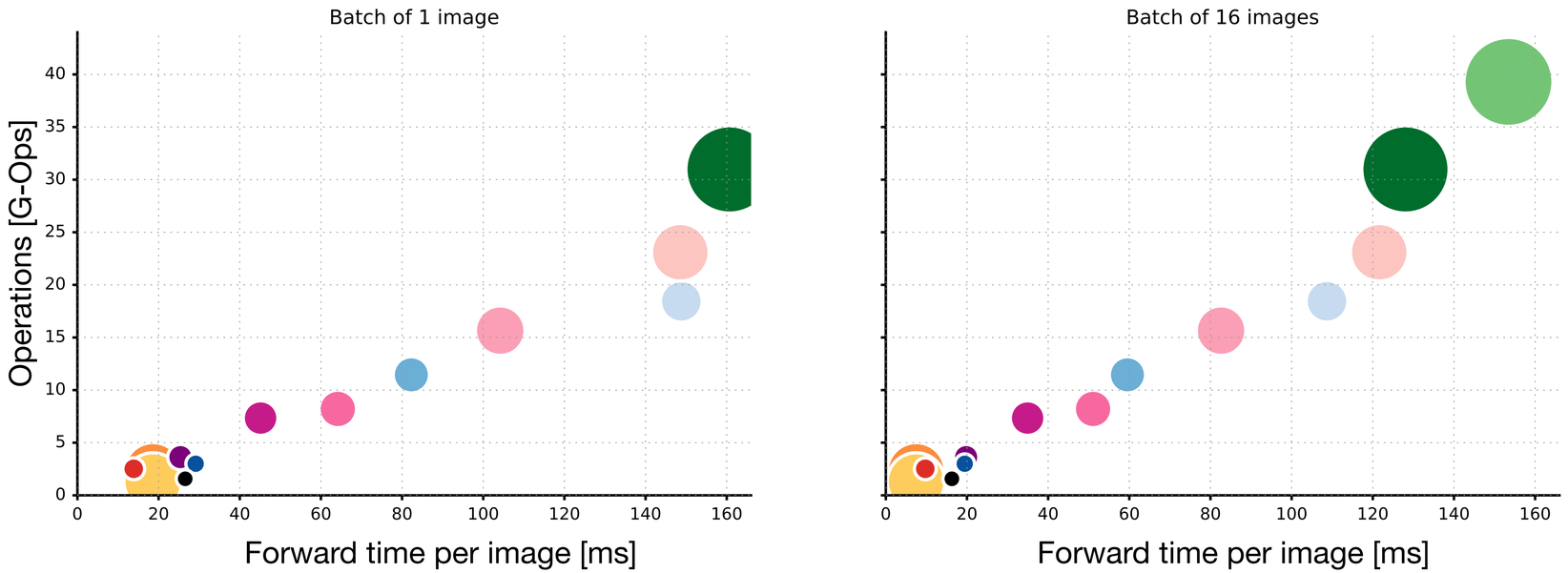


Figure 7: **Operations *vs.* inference time, size $\propto$ parameters.** Relationship between operations and inference

Units of measurement:

- If you care about memory footprint: # parameters, activations, gradients, bitwidth.
- If you care about speed: FLOPs, latency (wall-clock time).
- If you care about battery: Energy use (watt-hours).

Figure: Canziani, Paszke, Culurciello. An Analysis of Deep Neural Network Models for Practical Applications. arXiv preprint 1605.07678.

FLOPs

- As you may have noticed, this is kind of hard and implementation-dependent
- Rough estimate for amount of computation; big-O for matrix ops w/ potentially large constant factors.
- **Vector-vector operations:** Given vectors $a, b \in \mathbb{R}^n$
  - **Add/multiply** (a+b, a∘b) **=** n FLOPs; **Dot (inner) product** (a·b) **=** $2n$ FLOPs
- **Matrix-vector product:** Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$
  - **Dense:** Ab **=** $2mn$ FLOPs; ***A k*-banded:** $2nk$; ***A s*-sparse:** $2s$
- **Matrix-matrix product:** Given $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$
  - **Dense:** $AB = 2mnp$ FLOPs …

Further reading: https://www.stat.cmu.edu/~ryantibs/convexopt-F18/scribes/Lecture_19.pdf

What is “Energy”? “Power”?

Joule, Watt, BTU, calorie, foot-pound, foe, …

1 Watt = 1 Amp * 1 Volt

1 Amp = 1 Coulomb / s

1 Coulomb = 1/(1.602176634×10−19) e

…

1 Kilowatt = 1,000 Watts

1 Kilowatt hour = 1,000 Watts for an hour

Amps = How much stuff

Voltage = How much pressure

Watt = How much work

Batteries store stuff (mA)

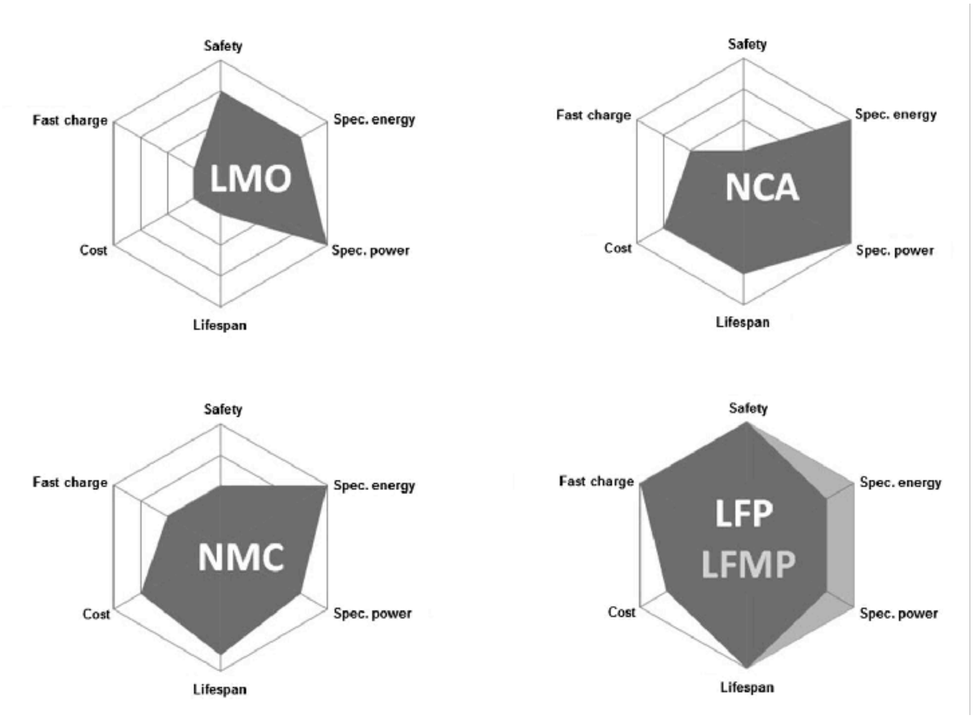
Batteries can provide pressures (3V/5V)

If Voltage is known, we can report Watts

Devices

ML is a “bonus”

Price? vs speed? vs lifespan? vs …



https://pushevs.com/2020/04/04/comparison-of-different-ev-batteries-in-2020/

A few example batteries

Tesla Model 3 — 76 kWh (168 Wh/kg)

Macbook Pro — 58 Wh

Macbook Air — 49.9 Wh

iPhone — 9.57-16.75 Wh

Apple Watch — 1.17 Wh

AirPods — 93 mWh

Batching and parallelism

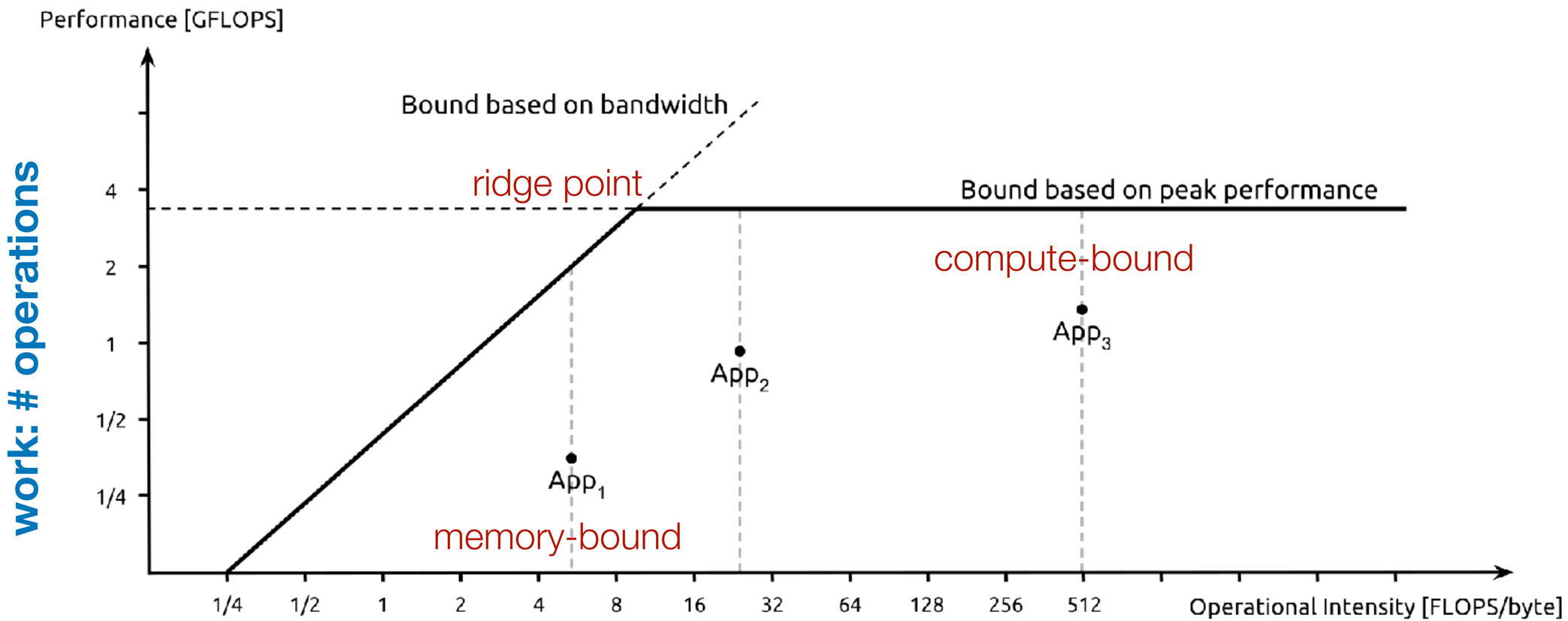
Batching makes neural accelerators fast by increasing throughput.

- Parallelism: more FLOPs, same latency.™ **Energy Δ?**
- How to measure parallel hardware efficiency theoretically: Roofline model.

- **Question:** To batch or not to batch when benchmarking?
- **Short Answer:** Both! Vary batch size and plot.
- **Long Answer:** In training, use most efficient batch size for hardware. In inference, ideally both, but **keep in mind your use case.**

Roofline

Roofline: upper bound on (memory, compute) performance

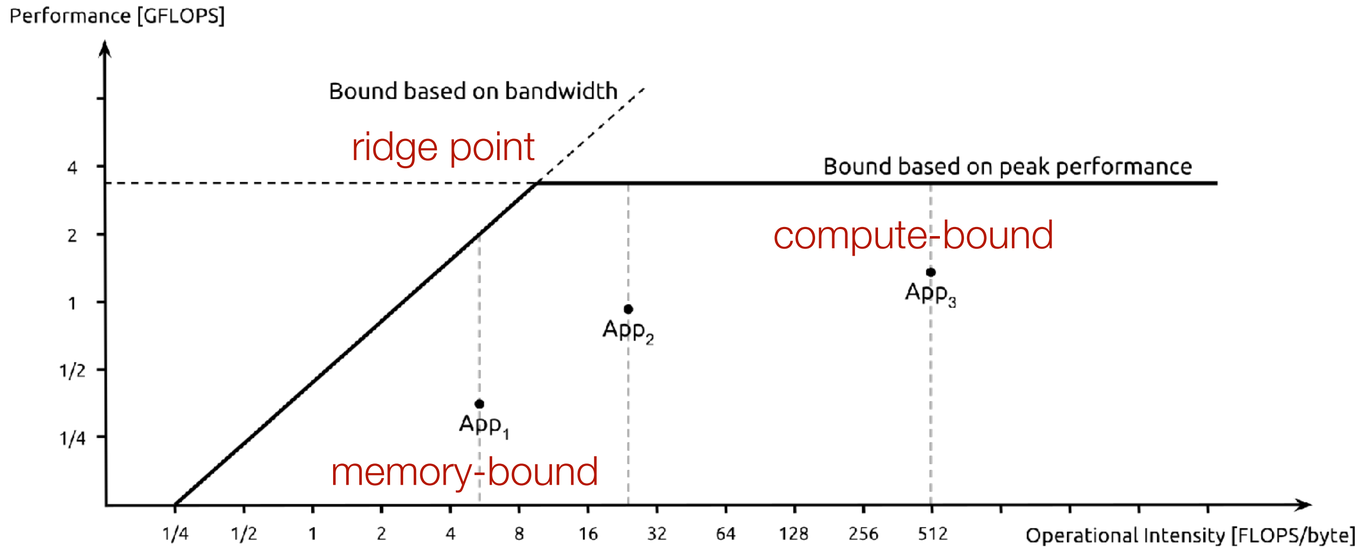


arithmetic intensity: ratio of #ops to data movement (DRAM traffic)

Figure: https://en.wikipedia.org/wiki/Roofline_model

S. Williams, A. Waterman, D. Patterson. Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. CACM, 2009.

Roofline

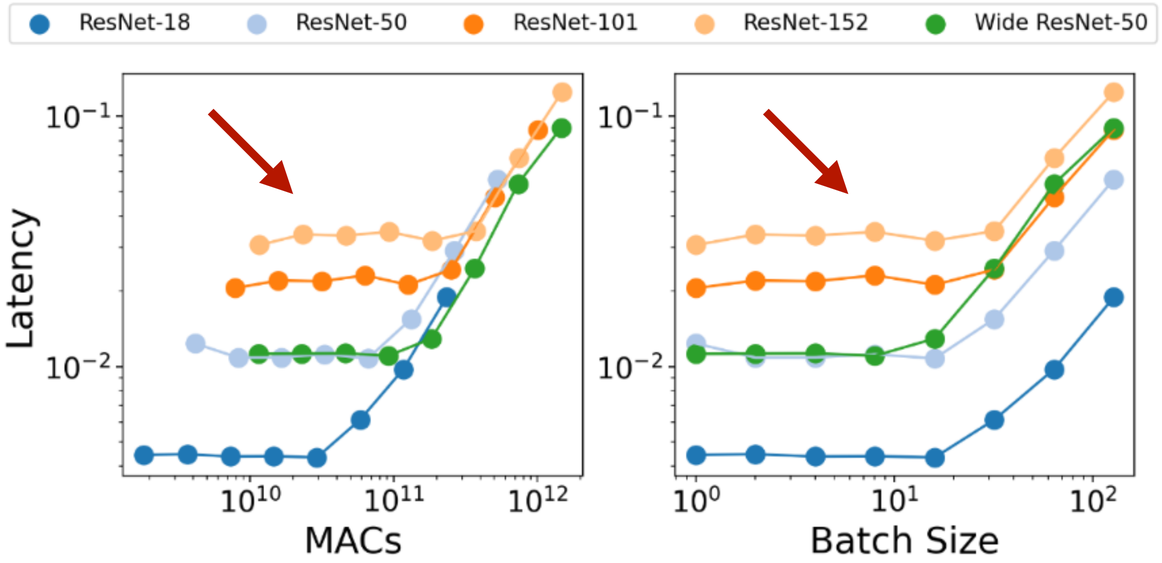


- **Compute-bound?**
  - Improve parallelism: maximize number of instructions per clock cycle
  - Balance adds and multiplies
- **Memory-bound?**
  - Leverage caching, software prefetching
  - Consider moving to a different framework (optimized for hardware prefetching?)

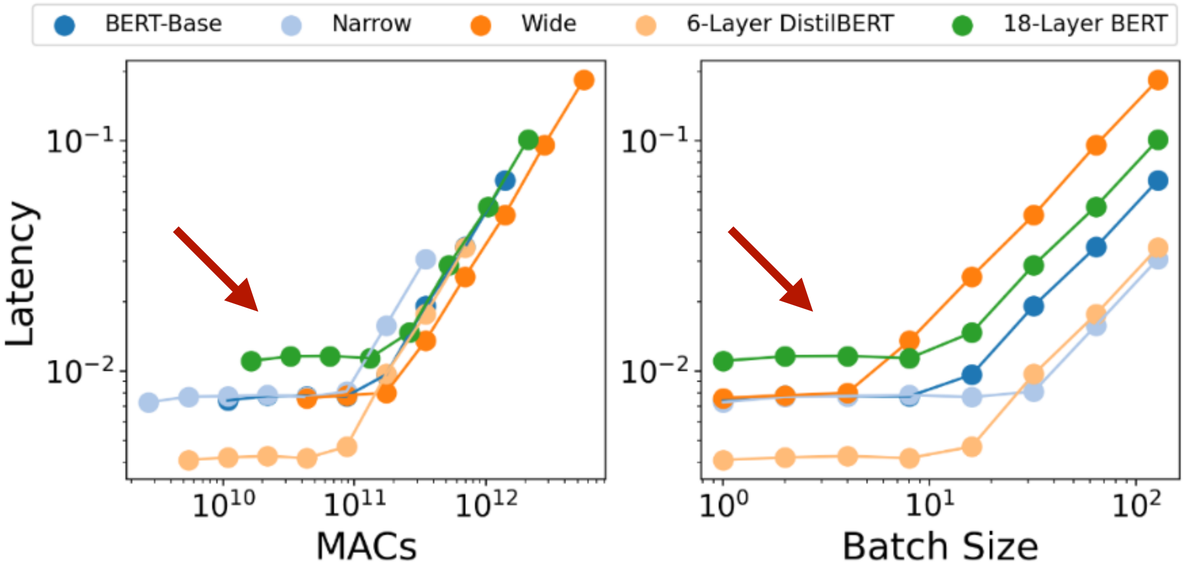
Figure: https://en.wikipedia.org/wiki/Roofline_model

S. Williams, A. Waterman, D. Patterson. Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. CACM, 2009.

Is your model saturating compute?



CV (ResNet)



NLP (BERT)

- Increases in model width lead to no increase in latency at low batch sizes.

J. Fernandez, J. Kahn, C. Na, Y. Bisk, E. Strubell. The Framework Tax: Disparities Between Inference Efficiency in Research and Deployment. 2023.

Same device — multiple benchmarks

Models don’t have a speed/time/latency/… **Systems do.**

Benchmark multiple settings:

1. Performance cores (CPU)
2. Efficiency cores (CPU)
3. Accelerators
4. Batching
5. Federated learning (WiFi/Cellular)

Frameworks and training

Same hardware — different software

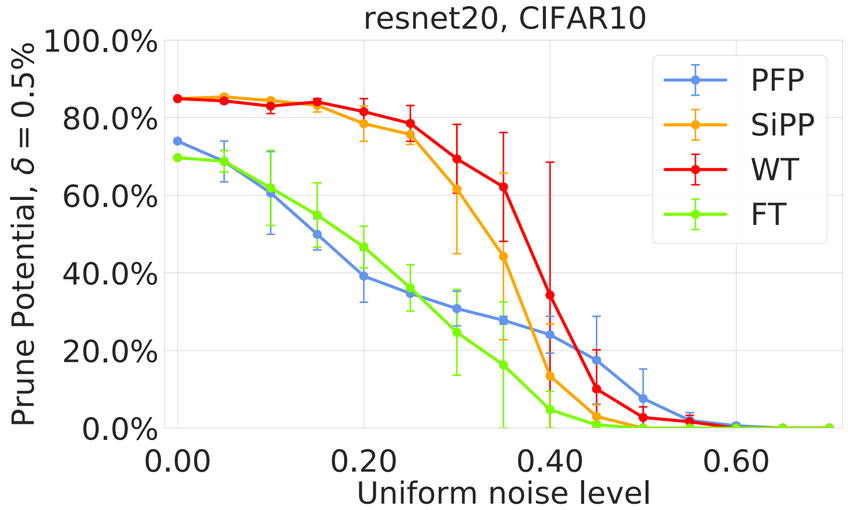
- no_grad()
- torchscript
- ONNX
- Huggingface Inifinity (?)

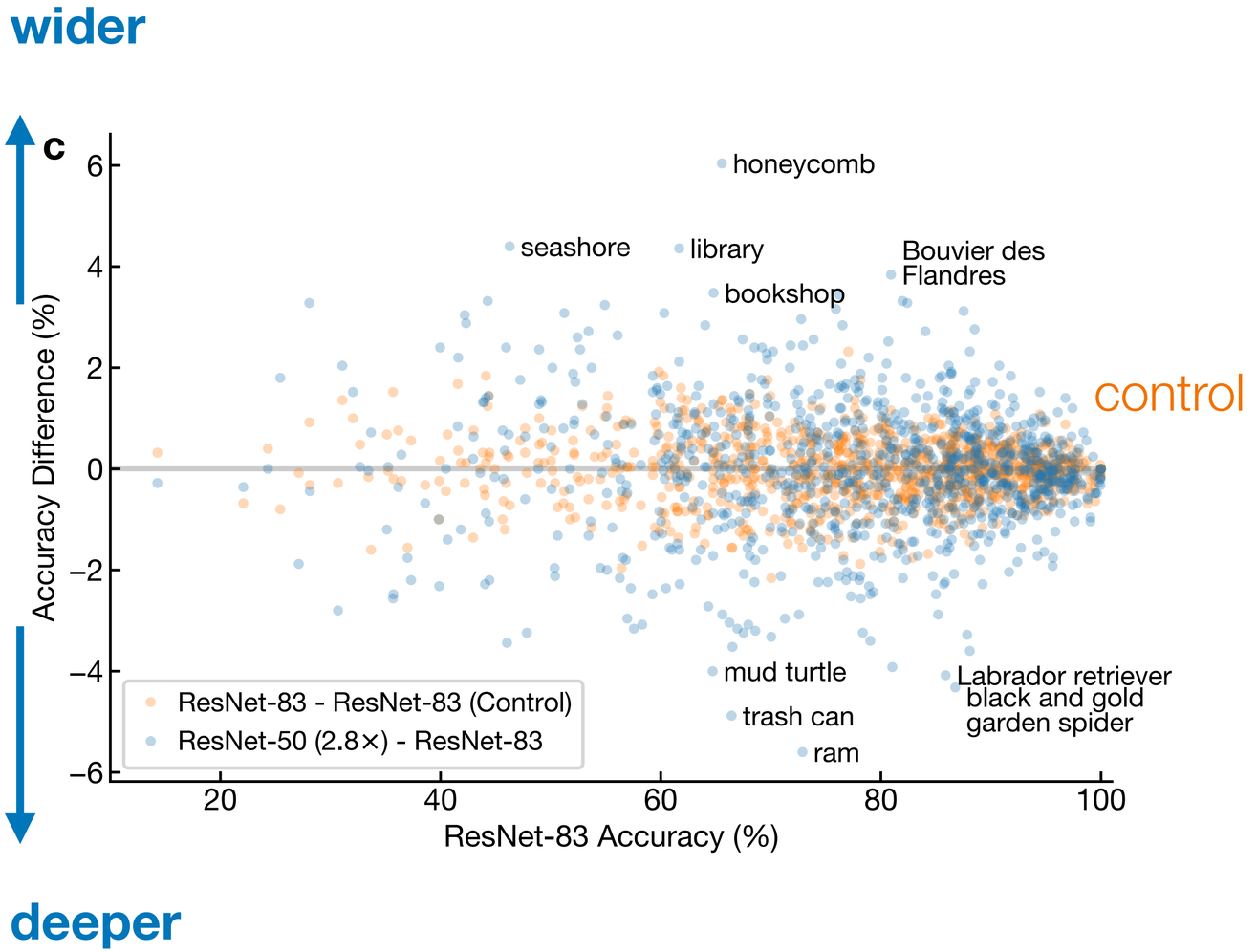
What if we want to train?

- Gradients for Hogwild
- Raw Data
- Features

Beyond preserved accuracy

- Top-1 accuracy is a bad benchmark.
- Wide & deep models w/ same average accuracy demonstrate statistically significant differences in example- and class-level predictions.
- Efficient models less robust to noise.





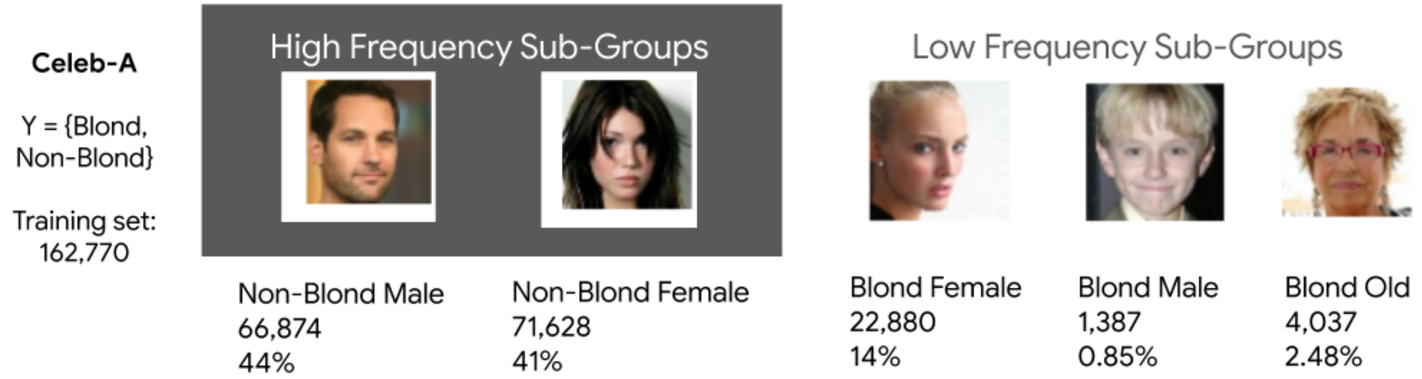
T. Nguyen, M. Raghu, S. Kornblith. Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth. ICLR 2021.

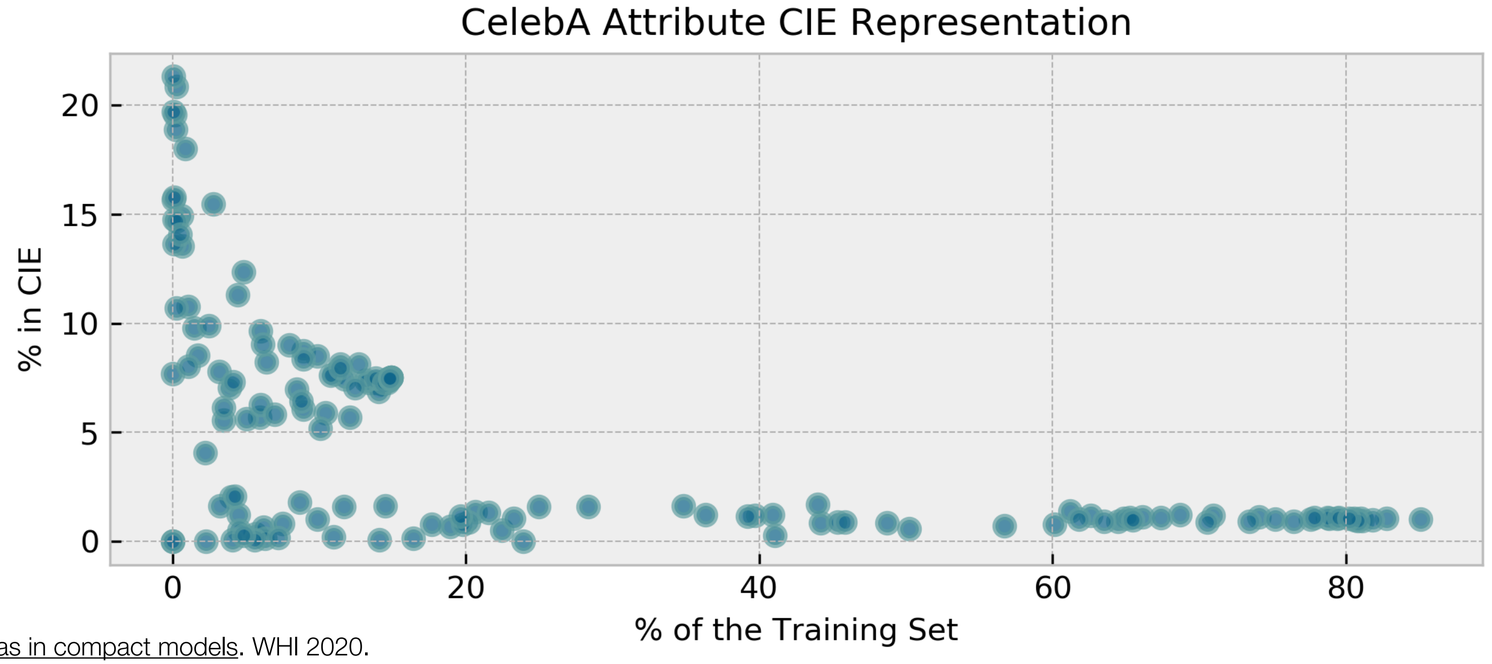
L. Liebenwein, C. Baykal, B.Carter, D. Gifford, D. Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. MLSys 2022.

Beyond preserved accuracy

- Efficient models sacrifice accuracy on long-tail phenomena in order to preserve accuracy on frequent examples.

- “Compression consistently **amplifies the disparate treatment of underrepresented protected subgroups** for all levels of compression”
(Hooker et al. 2020).





Hooker, S., Moorosi, N., Clark, G., Bengio, S., & Denton, E. Characterising and mitigating bias in compact models. WHI 2020.

Beyond preserved accuracy: Loyalty & Robustness

- **Label loyalty:** How similar are the predicted labels compared to the full model?

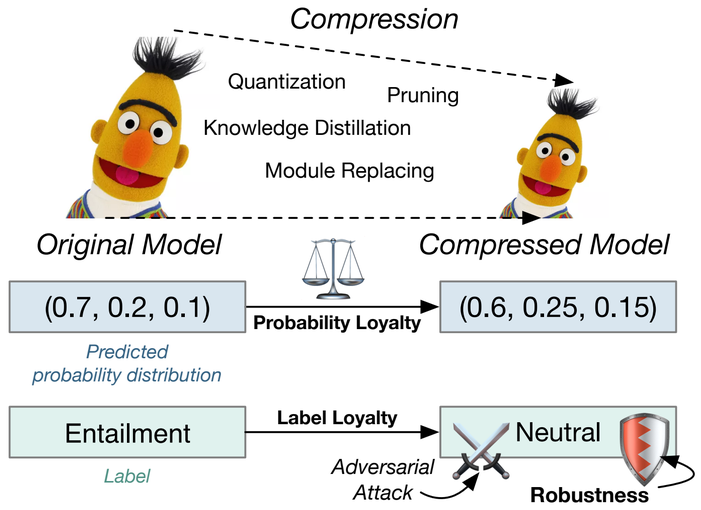
$$L_l = \text{Accuracy}(pred_t, pred_s)$$

- **Probability loyalty:** How similar are the compressed model's output probabilities compared to the full model?

$$D_{\text{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right) \qquad L_p(P\|Q) = 1 - \sqrt{D_{\text{JS}}(P\|Q)}$$

$$D_{\text{JS}}(P\|Q) = \frac{1}{2}D_{\text{KL}}(P\|M) + \frac{1}{2}D_{\text{KL}}(Q\|M)$$

- **Robustness:** How robust is the compressed model against an adversarial attack, compared to the full model?



Method	Speed	MNLI	L-L	P-L	AA	# Q
Teacher	1.0×	84.5 / 83.3	100	100	8.1	89.6
Head Prune	1.2×	80.9 / 80.6	87.8	85.5	9.1	90.5
+Finetune	1.2×	83.2 / 81.9	89.1	85.5	7.2	83.2
+KD	1.2×	**84.2 / 83.0**	**93.3**	**93.0**	8.3	90.5
+KD+PTQ	2.2×	80.8 / 80.4	89.6	86.3	**38.4**	**90.9**
Q8-QAT	1.8×	83.4 / 82.4	89.7	88.2	6.8	82.7
Q8-PTQ	1.8×	80.7 / 80.4	89.6	80.8	**40.2**	**91.6**
+Finetune	1.8×	82.9 / 81.9	89.7	84.8	7.1	84.5
+KD	1.8×	**84.1 / 83.5**	**94.0**	**93.9**	7.5	86.1
BERT-PKD	2.0×	81.3 / 81.1	88.9	89.0	6.4	81.9
Theseus	2.0×	81.8 / 80.7	88.1	82.5	8.3	89.7
+KD	2.0×	**82.6 / 81.7**	**91.2**	**91.4**	8.0	88.7
+KD+PTQ	3.6×	80.2 / 79.9	89.5	80.3	**36.5**	**91.3**

Xu, Zhou, Ge, Xu, McAuley, Wei. Beyond Preserved Accuracy: Evaluating Loyalty and Robustness of BERT Compression. EMNLP 2021.