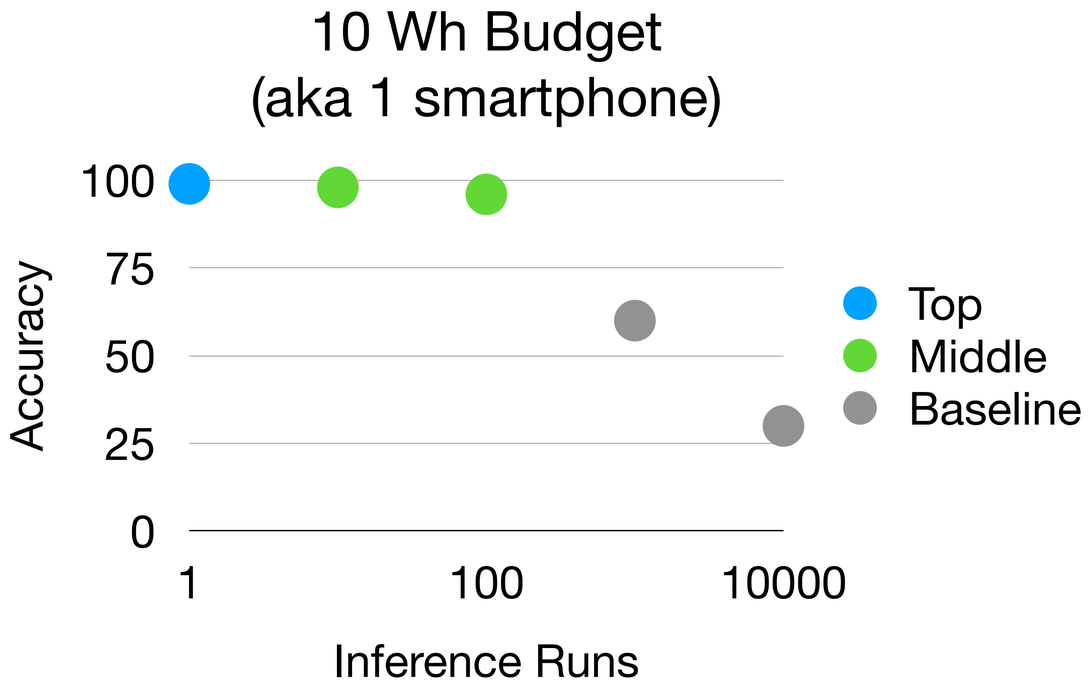
Main plot in final report

How many times can I run inference on a phone battery?

- Produce a performance-under-budget curve:
  Model accuracy vs *Number of times you can run inference*.



- We have provided all teams with multimeters to measure device power draw.
- Teams with server-level projects can use nvidia-smi, Intel RAPL interface, etc.
- Not all data points need to be “same” model
  1. Different detectors
  2. Different feature extractors
  3. Different size vocabularies
  4. Different image resolutions

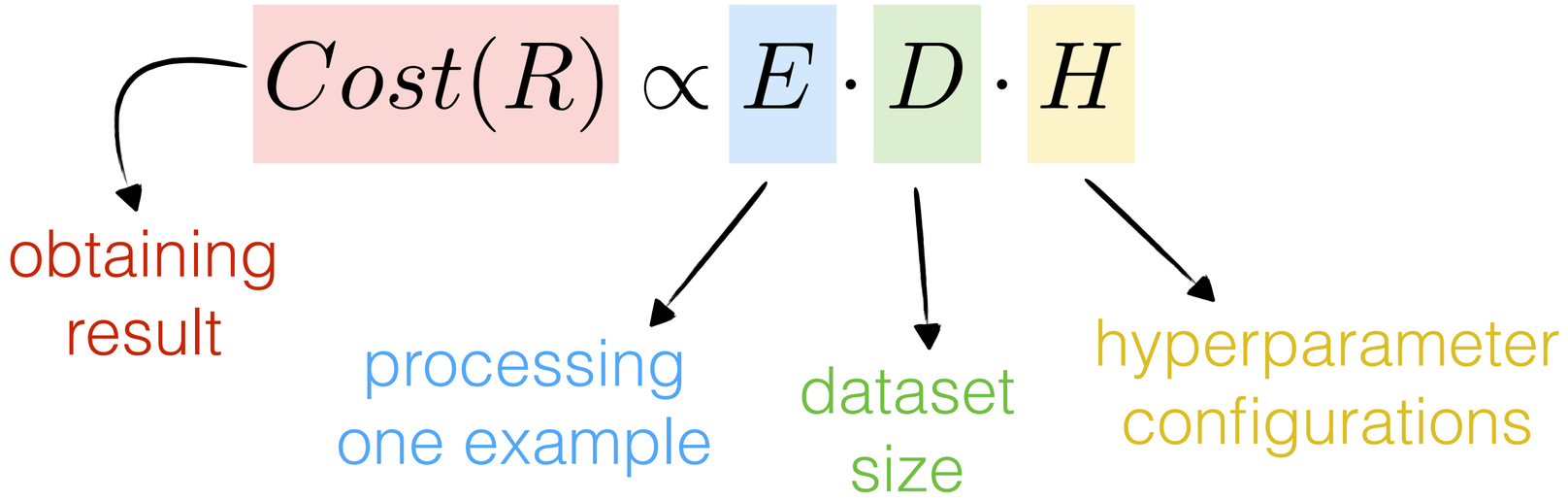
Carnegie Mellon University
Language Technologies Institute

Accelerating Training

Off- and on-device

Emma Strubell & Yonatan Bisk

Training efficiency



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

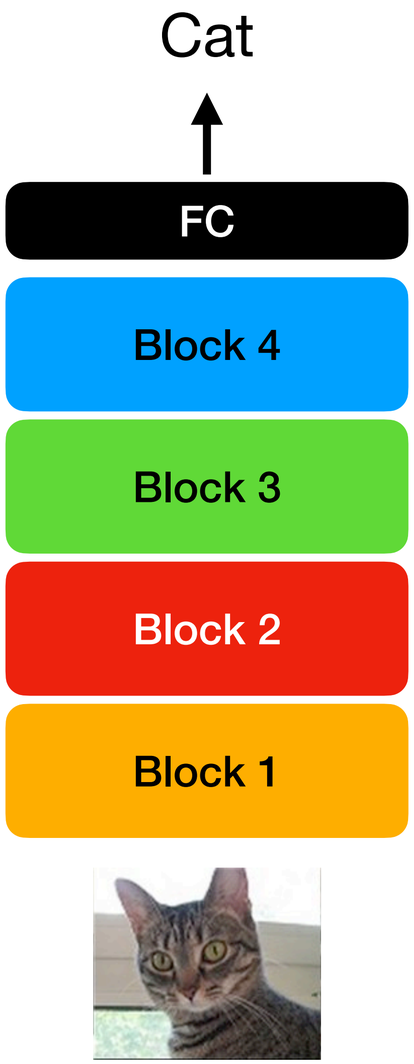
Carnegie Mellon University
Language Technologies Institute

Avoiding ~~Accelerating~~ Training

Off- and on-device

Emma Strubell & Yonatan Bisk

When and What to train?



Which layer should change?

New class:

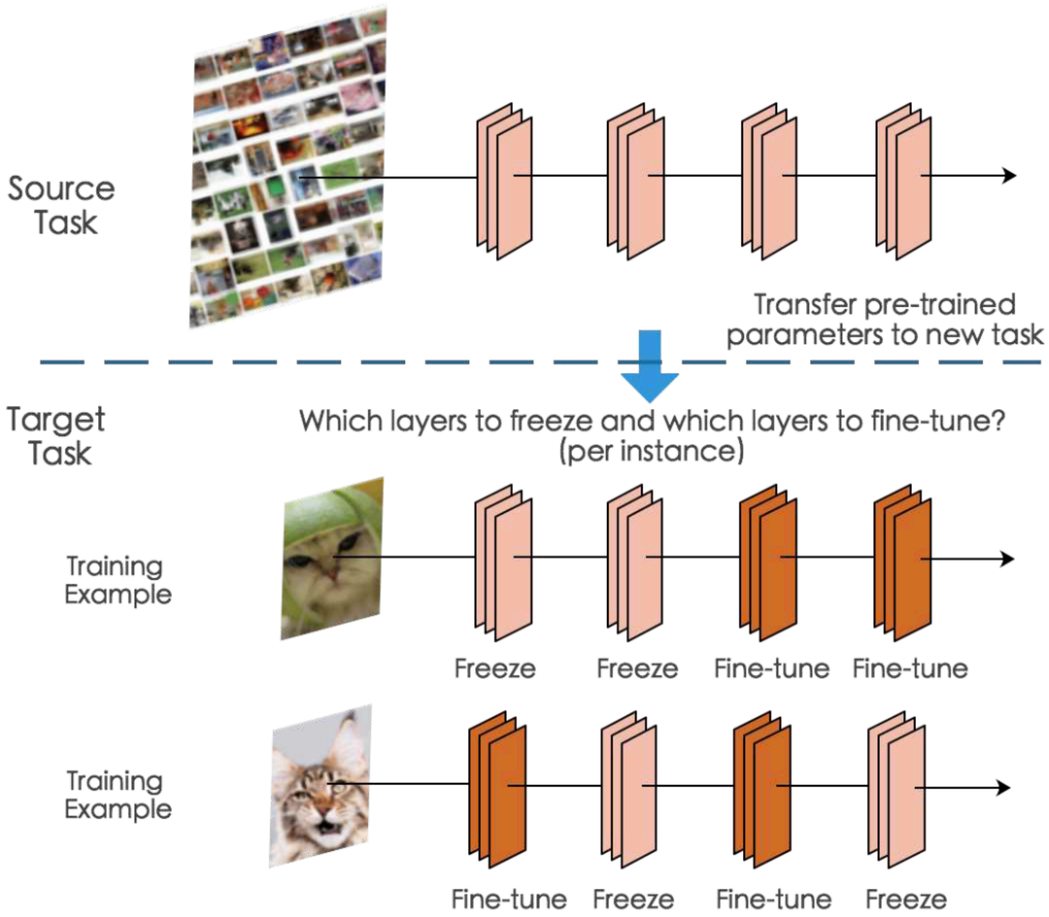
Felis catus

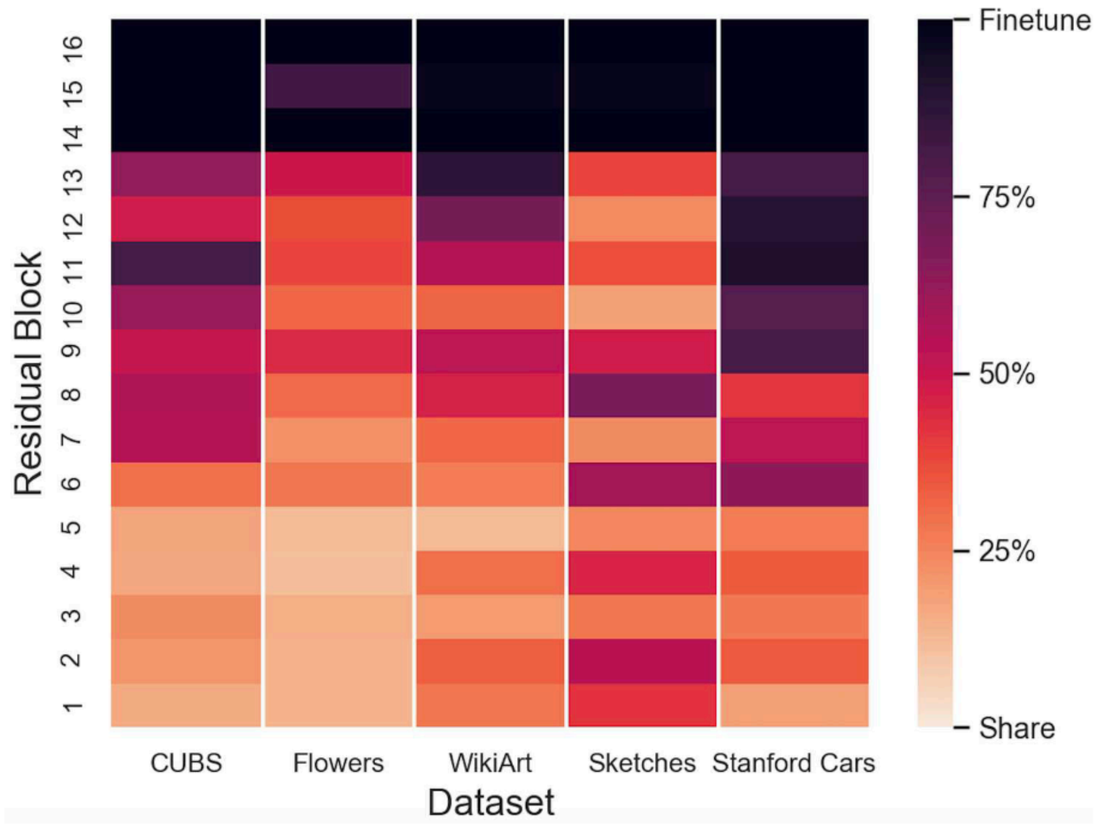
Dog

Yellow

Tumor

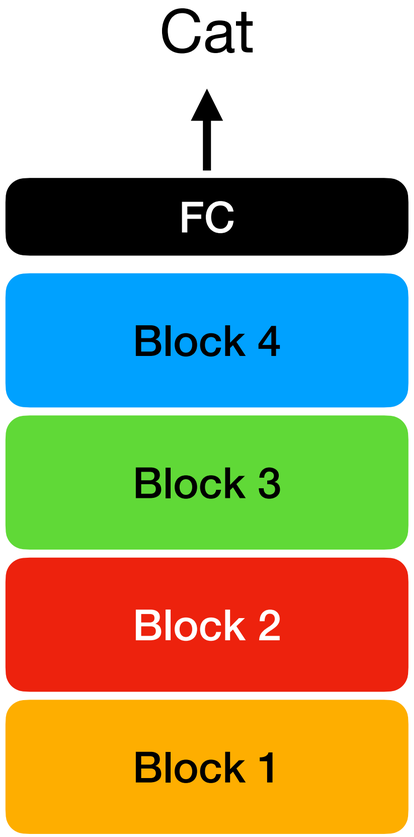
Learn where to adapt





SpotTune: Transfer Learning through Adaptive Fine-tuning
CVPR 2019

When and What to train?



Which should change?

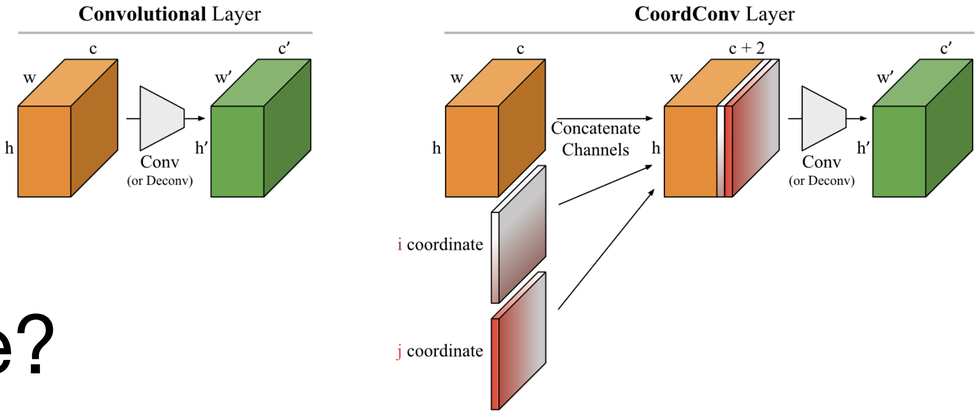
New task:

Coordinate of cat's nose

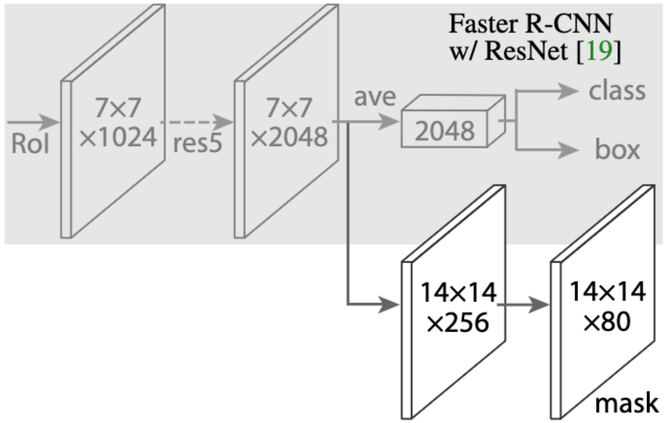
Training CLIP

Input to captioning model

Mask prediction



https://arxiv.org/abs/1807.03247v2



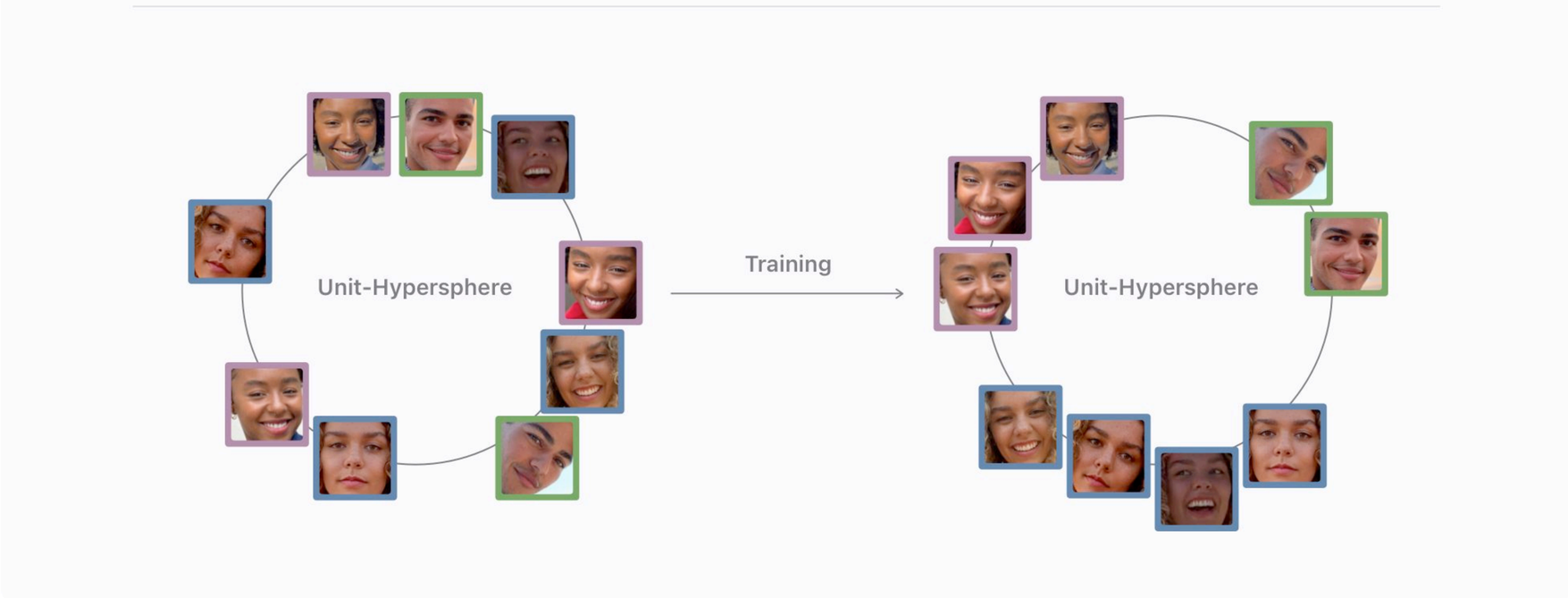
Domain and temporal shift

When do models have “zero-shot” performance?

When do models have “few-shot” performance?

Remember week 1?

- Learned exemplars
- Maximize margins

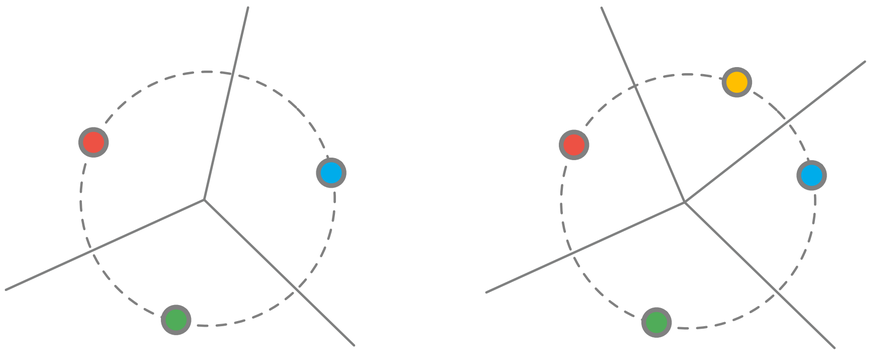


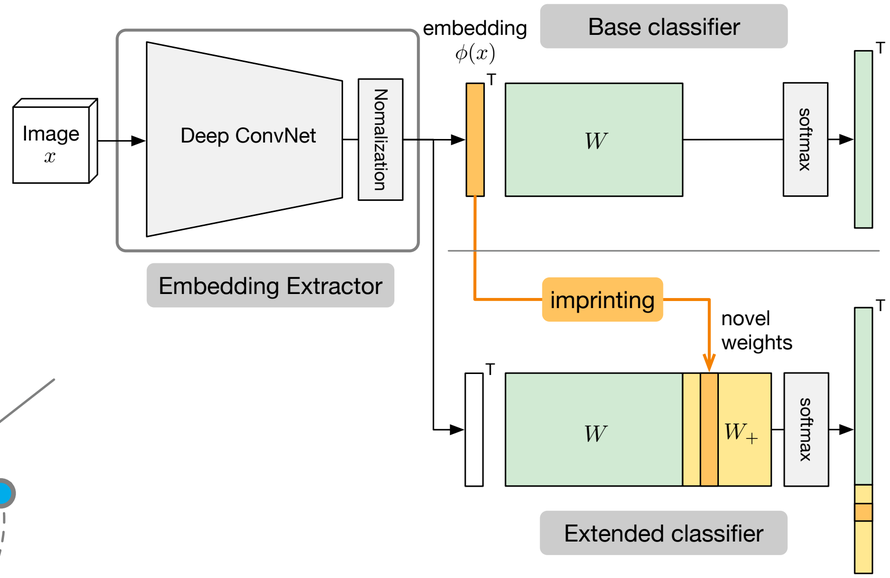
https://arxiv.org/abs/1712.07136

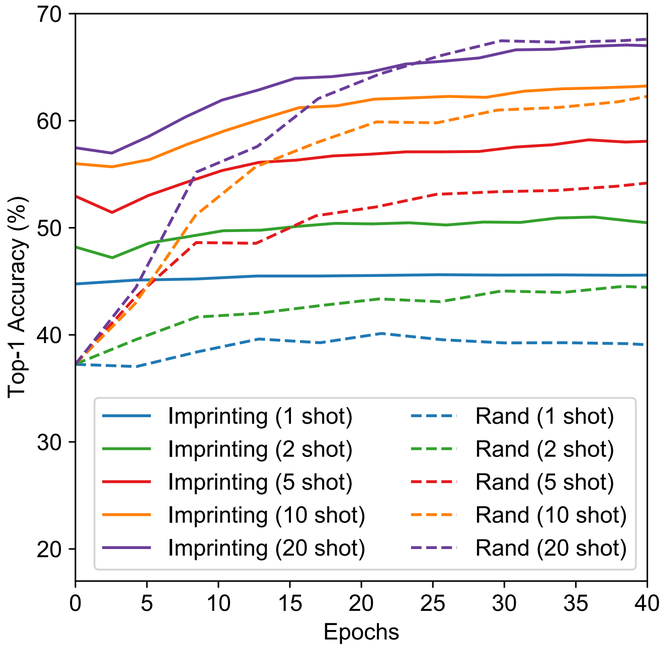
Weight imprinting

What is “learning”

If representations are sufficiently:
1.Consistent
2.Disjoint
We can add classes via exemplars





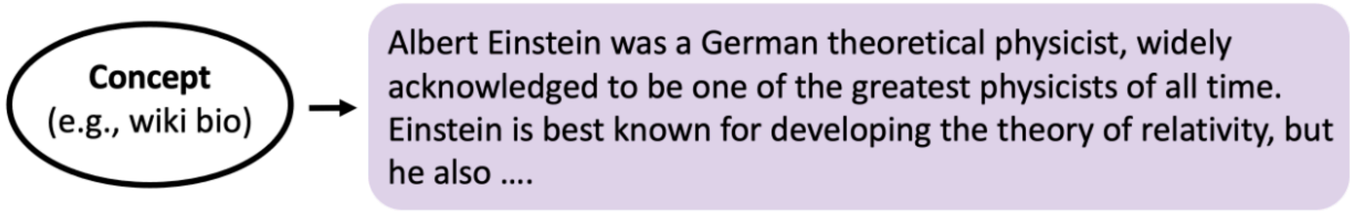


Shares conceptually with Nearest Neighbors

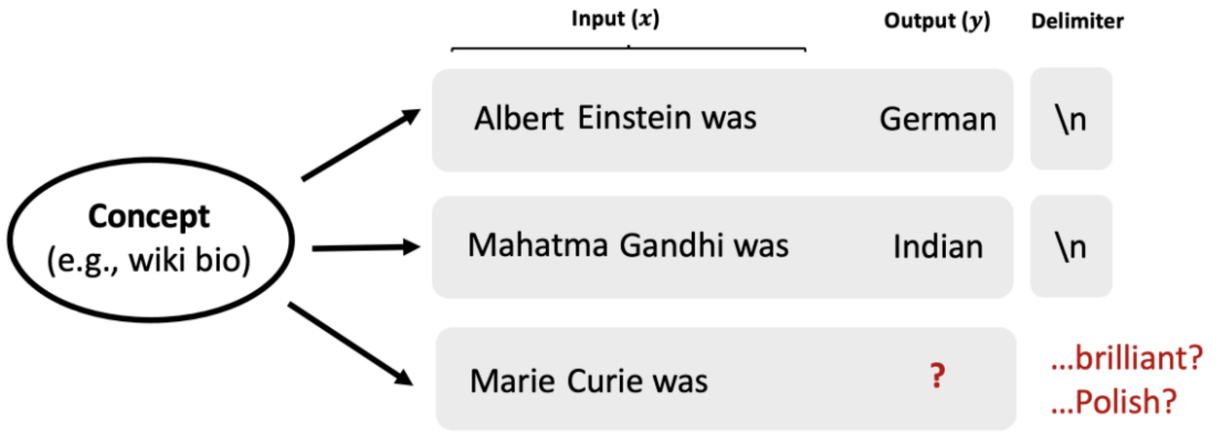
In-Context Learning

p("Polish" | "Albert Einstein was German \n Mahatma Gandhi was Indian \n Marie Curie was")

1. Pretraining documents are conditioned on a **latent concept** (e.g., biographical text)



2. Create **independent examples** from a **shared concept.** If we focus on full names, wiki bios tend to relate them to nationalities.



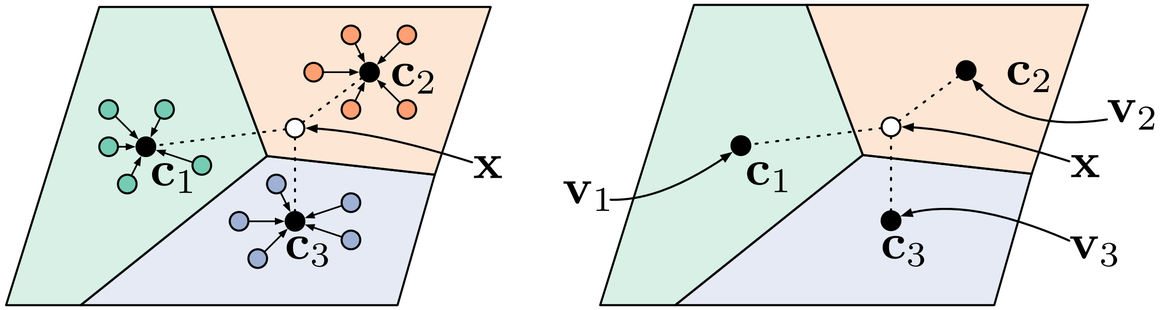
3. Concatenate examples into a prompt and predict next word(s). **Language model (LM) implicitly infers the shared concept** across examples despite the unnatural concatenation

https://arxiv.org/abs/2111.02080

Training on device

Can we get away with collecting and training on just a handful of examples?

- Few-shot and zero-shot learning:
  - 5, 10, 100 examples per label?
  - 0 examples?
- Metalearning: “learning to learn.”
  - Can we train the model to to be able to learn quickly later?
  - Compute/estimate gradients of gradients, e.g. MAML (Finn et al. 2017), Reptile (Nichol et al. 2018).



(a) Few-shot (b) Zero-shot

Figure: Snell et al. 2017

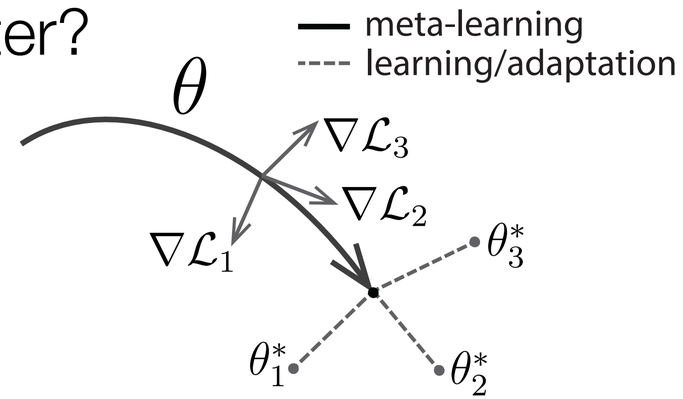


Figure: Finn et al. 2017.

Carnegie Mellon University
Language Technologies Institute

Accelerating ~~Training~~ Finetuning

Off- and on-device

Emma Strubell & Yonatan Bisk

Parameter-efficient fine-tuning

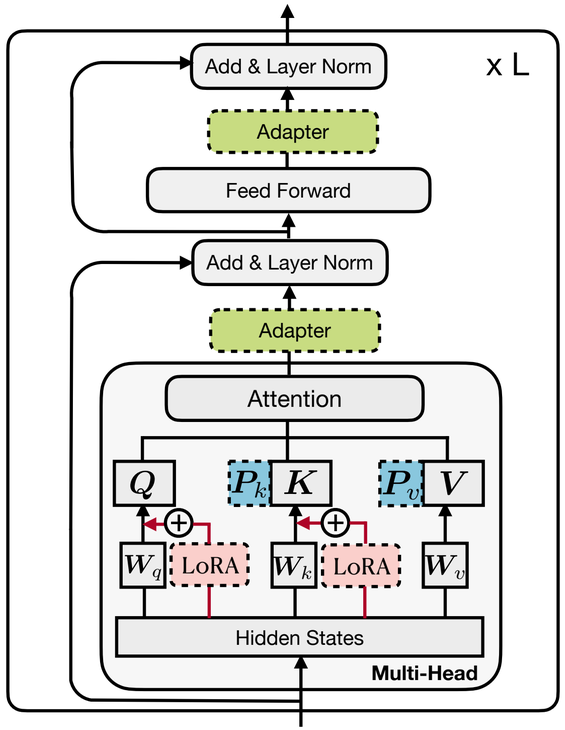
Motivation: BERT as fixed feature extractor is sub-optimal (Devlin et al. 2019).

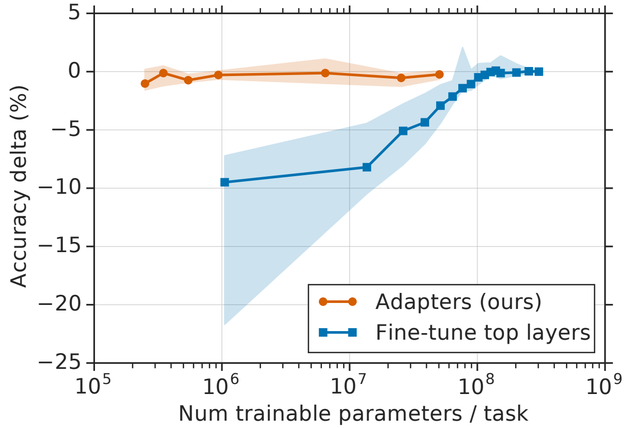
System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Full fine-tuning (updating all weights) works better than freezing weights and learning small transformation.

Parameter-efficient fine-tuning

Can we get away with updating a few additional parameters?





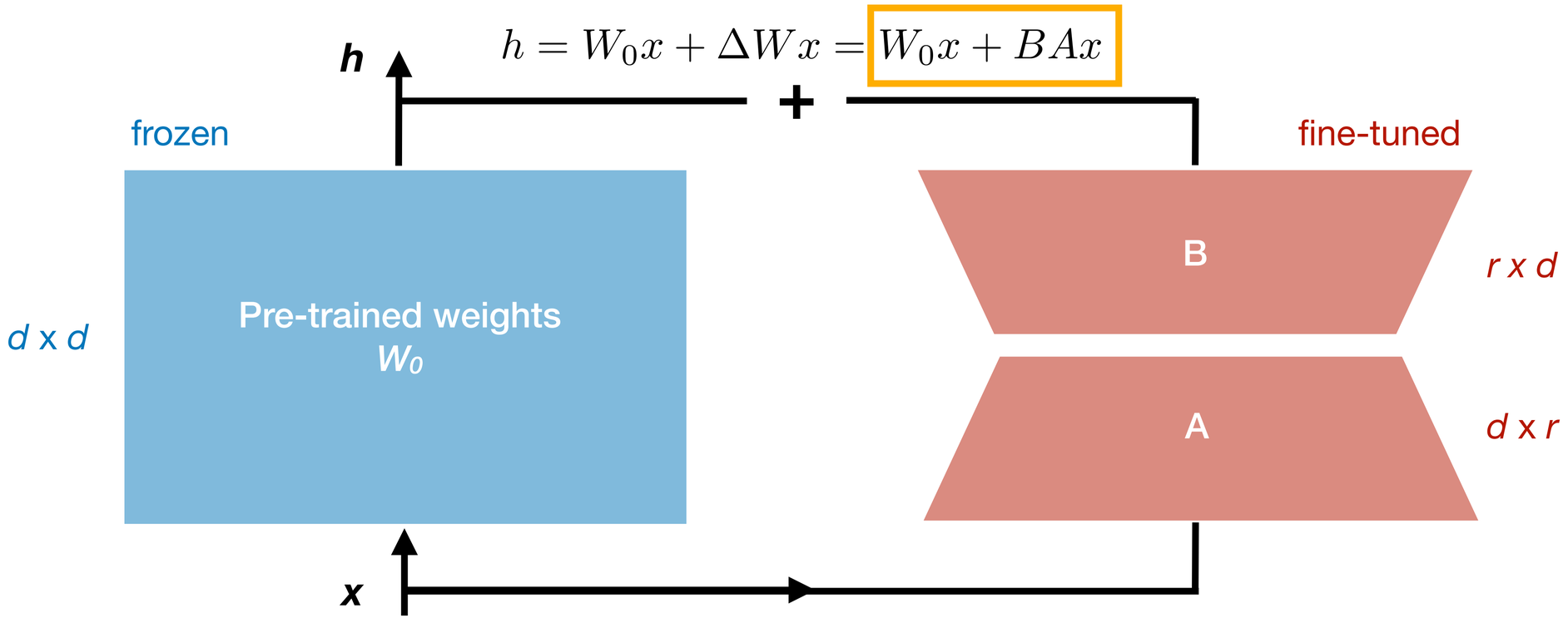
- **Adapters** (Houlsby et al. 2019): insert small feed-forwards between layers & update only those.
- **Prefix/prompt tuning** (Li & Liang, 2021, Lester et al. 2021): prepend learnable "tokens" at input or hidden layers.
- **LoRA** (Hu et al. 2021): learn low-rank approximation for parameter updates.

LoRA: Low-Rank Adaptation of LLMs

- **Idea:** Full fine-tuning requires too much memory.

  Adapters add inference latency, prefix tuning is hard to optimize.

- Instead of adding more model parameters, learn low-rank updates to weights.



QLoRA: Fine-tuning Quantized LLMs

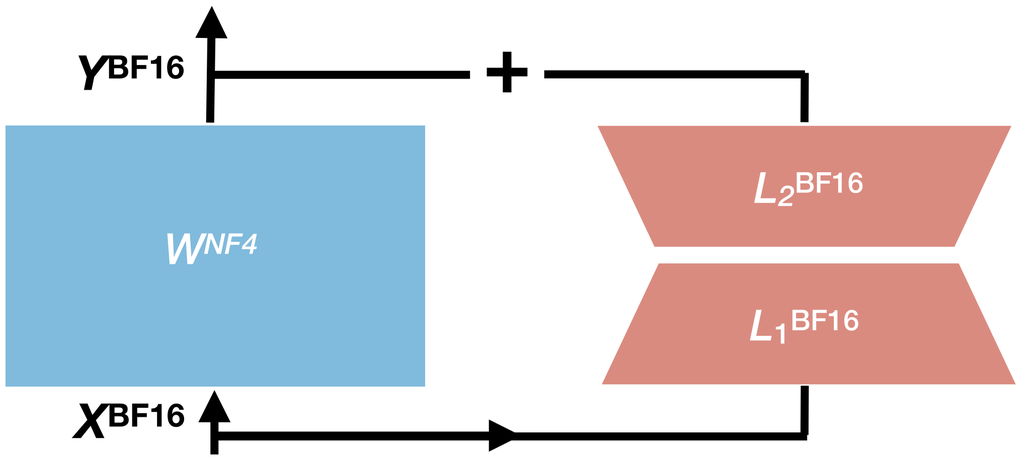
- **Idea:** LLMs *still* too big to fine-tune on a single GPU. Fine-tune quantized model!

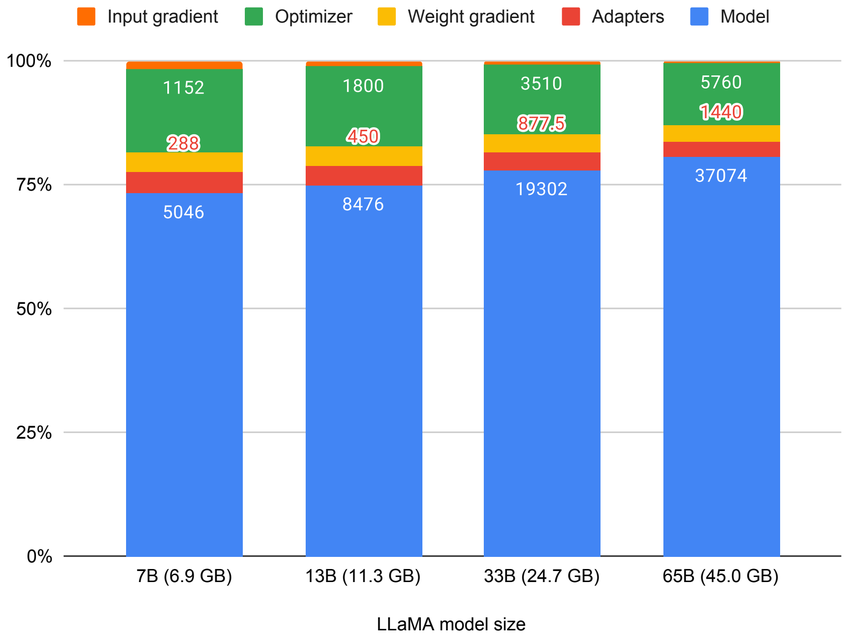
1. Quantize model to 4 bits in *NF4* using *blockwise* k-bit *double* quantization.

$$Q(r) = round(\frac{r}{S} + Z) \qquad \mathbf{Y}^{\text{BF16}} = \mathbf{X}^{\text{BF16}}\text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) + \mathbf{X}^{\text{BF16}}\mathbf{L}_1^{\text{BF16}}\mathbf{L}_2^{\text{BF16}},$$

2. Fine-tune quantized model using LoRA. Add LoRA to *all* linear layers.

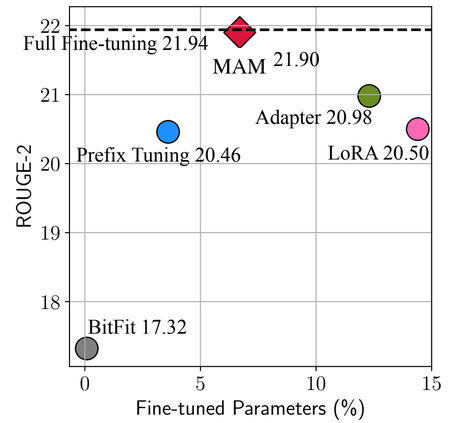
3. Paged optimizer states to handle long sequences.

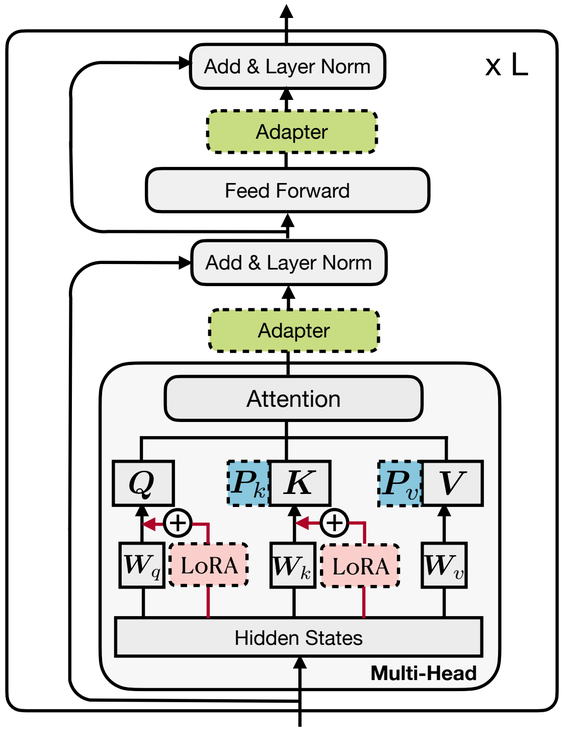


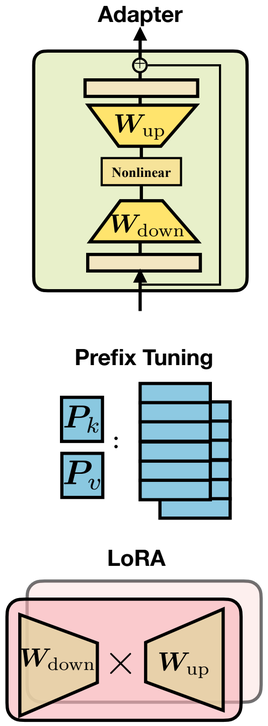


Parameter-efficient fine-tuning

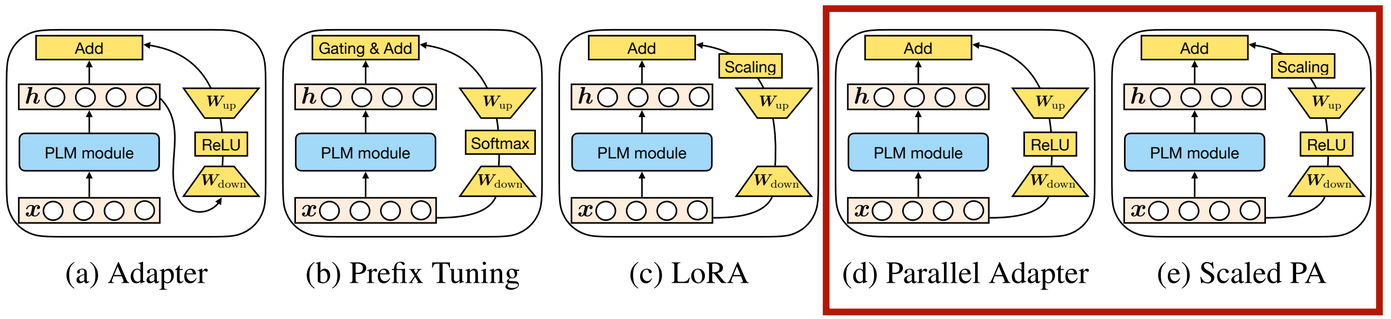
Can we get away with updating a few additional parameters?







- **Adapters** (Houlsby et al. 2019): insert small feed-forwards between layers & update only those.
- **Prefix/prompt tuning** (Li & Liang, 2021, Lester et al. 2021): prepend learnable "tokens" at input or hidden layers.
- **LoRA** (Hu et al. 2021): learn low-rank approximation for parameter updates.
- **MAM** (He et al. 2022): Mix-and-match all of the above.

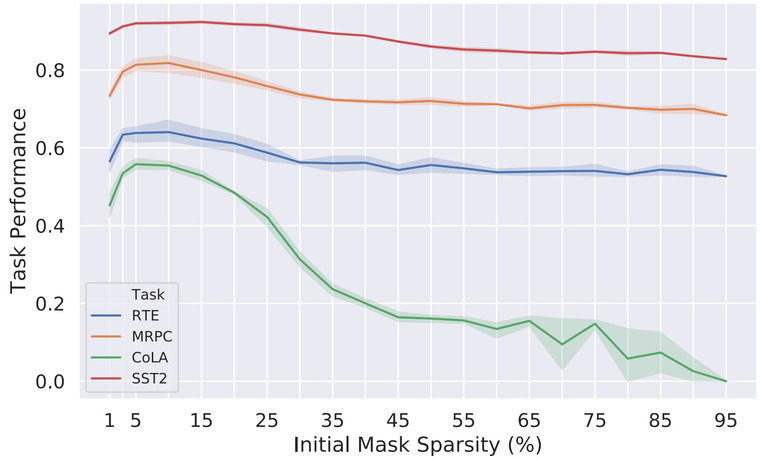


(a) Adapter (b) Prefix Tuning (c) LoRA (d) Parallel Adapter (e) Scaled PA

Parameter-efficient fine-tuning

Can we get away with updating just a subset of parameters, or none at all?

- **Masking** (Zhao et al. 2020): Simply learn a binary mask over all parameters!
- **Diff Pruning** (Guo et al. 2021): Learn a sparse additive mask over all parameters.
- **BitFit** (Zaken et al. 2022): Fine-tune just the bias vectors (or a subset of them).



	Train size	%Param	QNLI 105k	SST-2 67k	MNLI_m 393k	MNLI_mm 393k	CoLA 8.5k	MRPC 3.7k	STS-B 7k	RTE 2.5k	QQP 364k	Avg.
(V)	Full-FT†	100%	**93.5**	**94.1**	**86.5**	**87.1**	**62.8**	**91.9**	89.8	71.8	**87.6**	**84.8**
(V)	Full-FT	100%	91.7±0.1	93.4±0.2	85.5±0.4	85.7±0.4	62.2±1.2	90.7±0.3	**90.0±0.4**	**71.9±1.3**	87.5±0.4	84.1
(V)	Diff-Prune†	0.5%	**93.4**	**94.2**	**86.4**	**86.9**	63.5	91.3	89.5	71.5	**86.6**	**84.6**
(V)	BitFit	0.08%	91.4±2.4	93.2±0.4	84.4±0.2	84.8±0.1	**63.6±0.7**	**91.7±0.5**	**90.3±0.1**	**73.2±3.7**	85.4±0.1	84.2
(T)	Full-FT‡	100%	91.1	**94.9**	86.7	85.9	**60.5**	**89.3**	**87.6**	70.1	**72.1**	**81.8**
(T)	Full-FT†	100%	**93.4**	94.1	86.7	**86.0**	59.6	88.9	86.6	**71.2**	71.7	81.5
(T)	Adapters‡	3.6%	90.7	94.0	84.9	85.1	59.5	89.5	**86.9**	71.5	**71.8**	81.1
(T)	Diff-Prune†	0.5%	**93.3**	94.1	**86.4**	**86.0**	**61.1**	**89.7**	86.0	70.6	71.1	**81.5**
(T)	BitFit	0.08%	92.0	**94.2**	84.5	84.8	59.7	88.9	85.5	**72.0**	70.5	80.9

BitFit: Comparable to adapters, diff pruning while using 45x, 6x fewer parameters.

What about in-context learning?

- Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning (Liu et al. 2022).