Model Compression

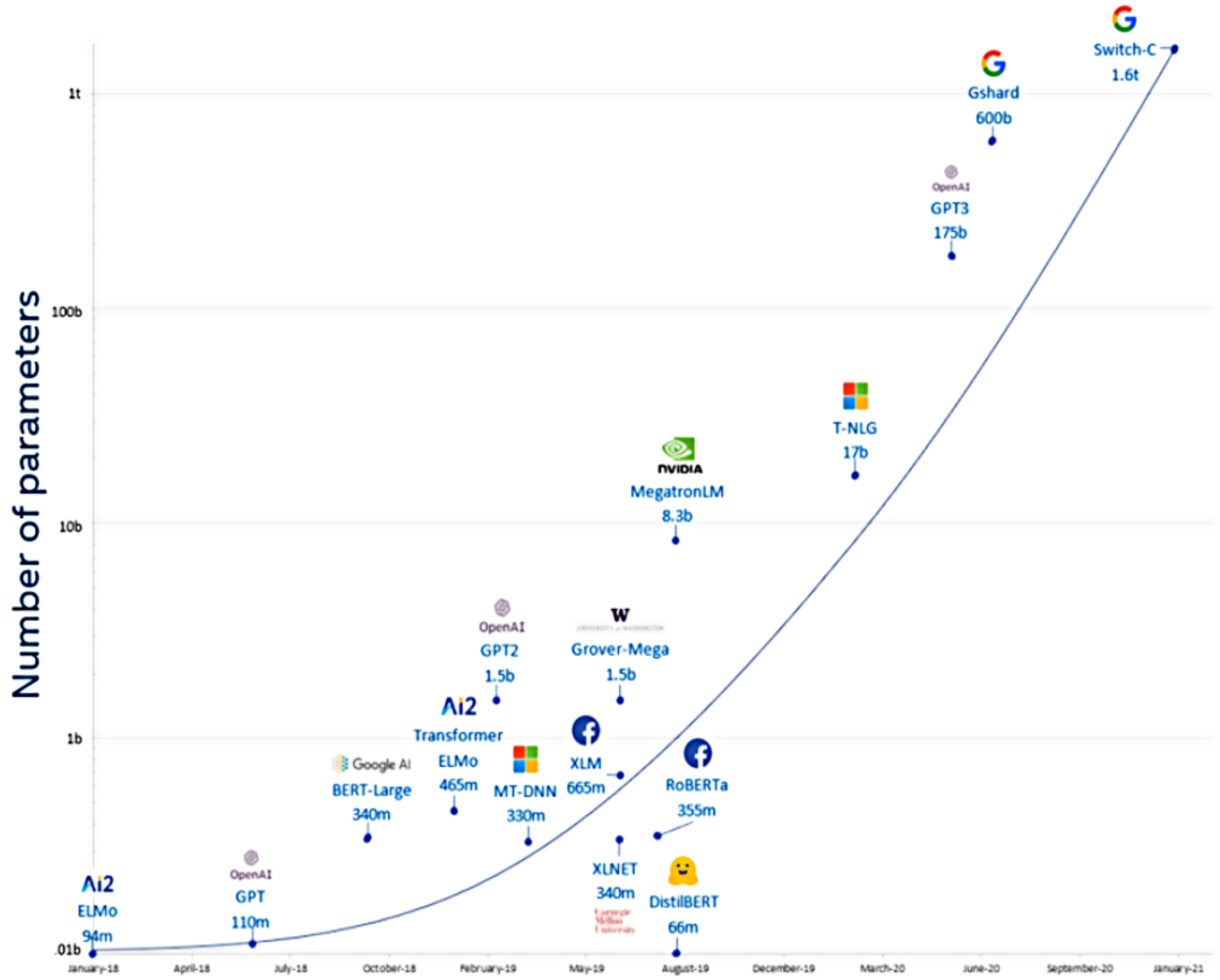
Antoine Bosselut

Outline

- Motivation
- Compression methods
  - Pruning
  - Quantization
  - Knowledge distillation
- Speculative Decoding

Growth of model parameters

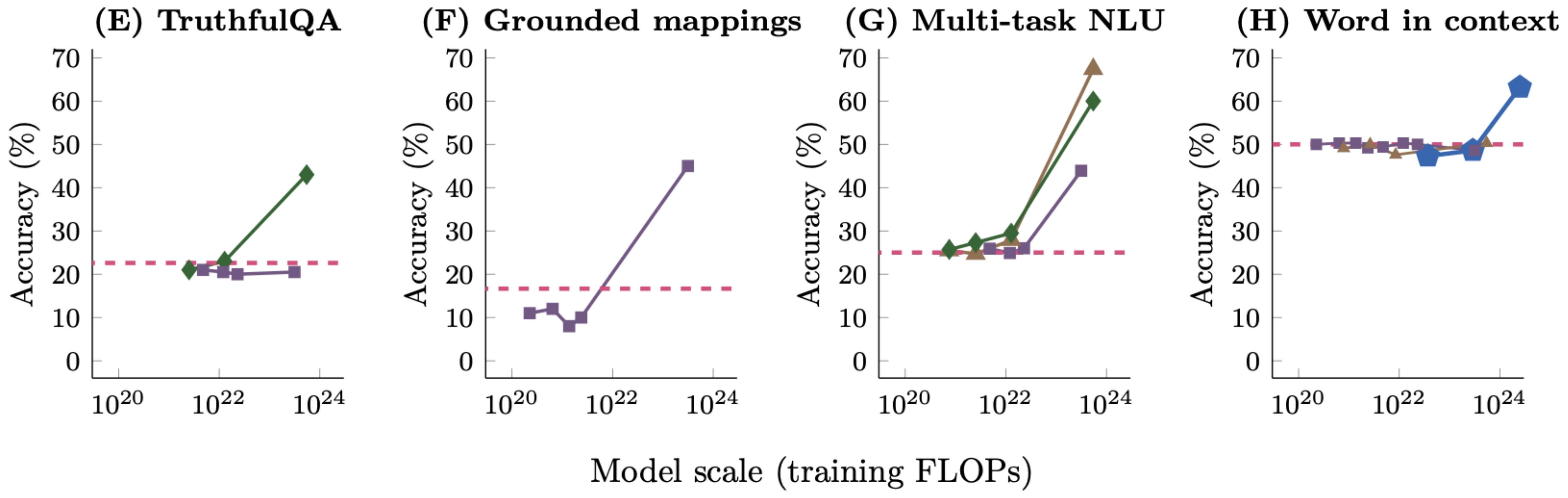
- Exponential growth in model parameters

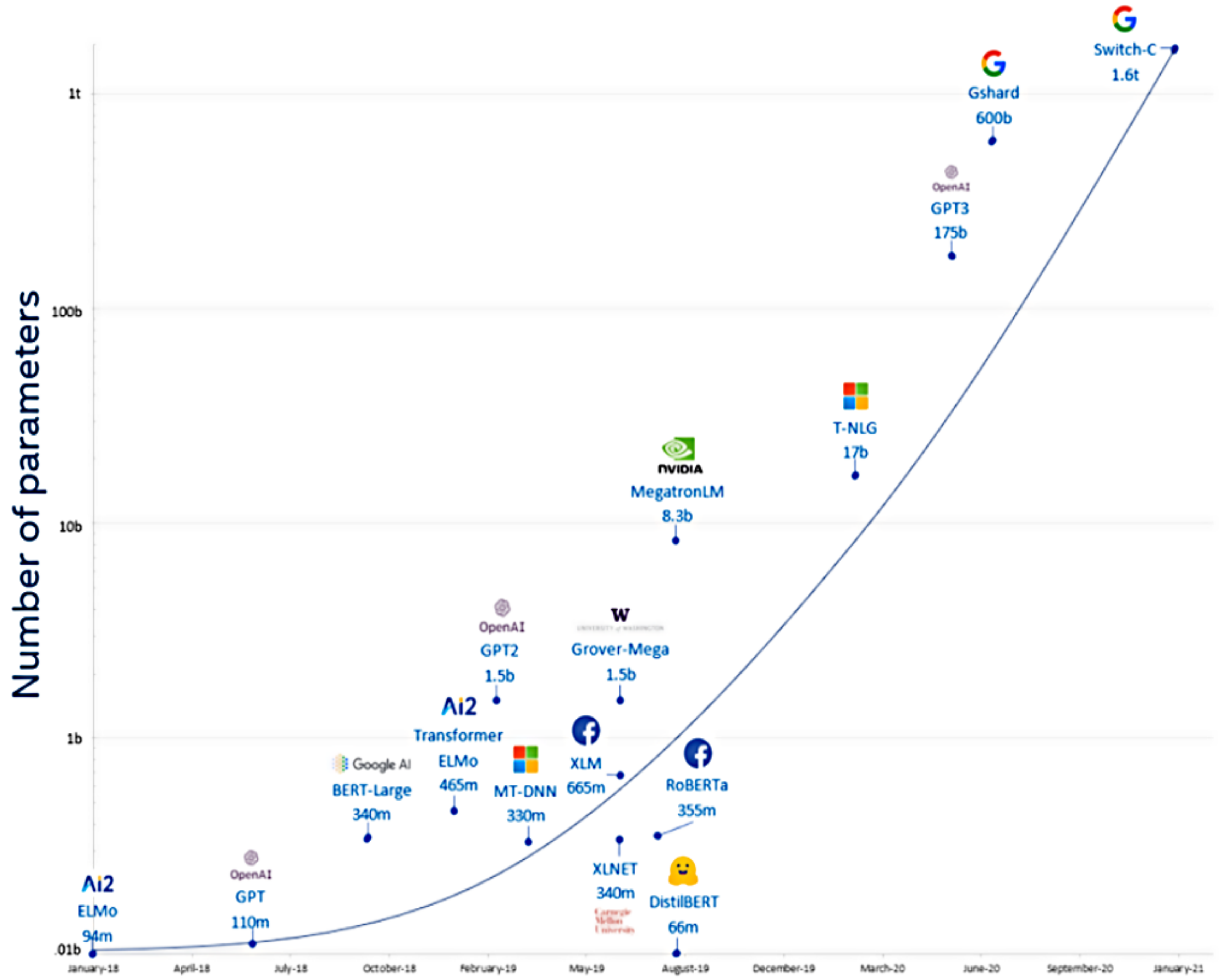


https://towardsdatascience.com/the-rise-of-cognitive-ai-a29d2b724ccc

Growth of model parameters

- Exponential growth in model parameters
  - Scaling laws
  - Emergent abilities of LLMs

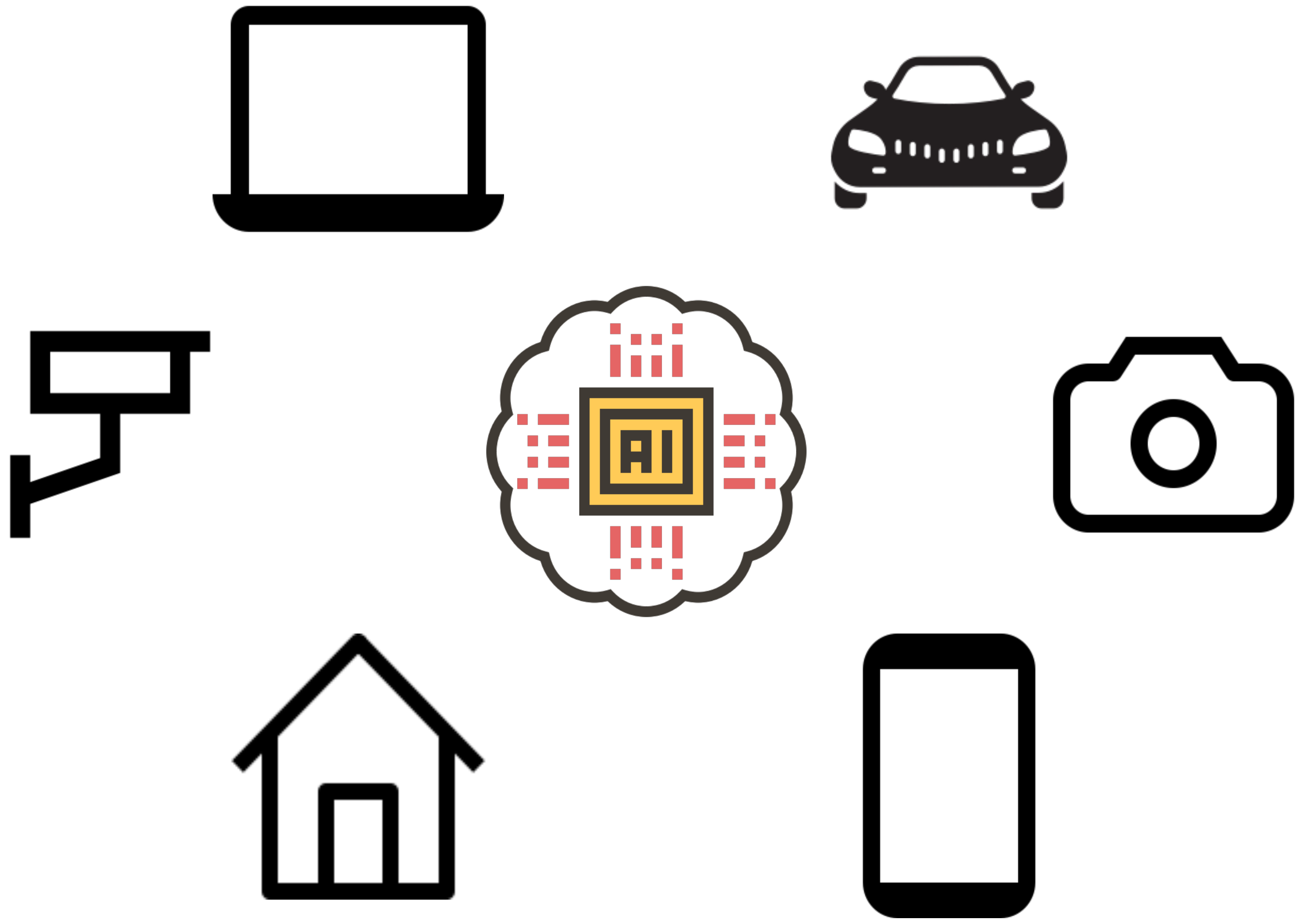




Kaplan, Jared, et al. "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361* (2020)
Wei, Jason, et al. "Emergent abilities of large language models." *arXiv preprint arXiv:2206.07682* (2022)

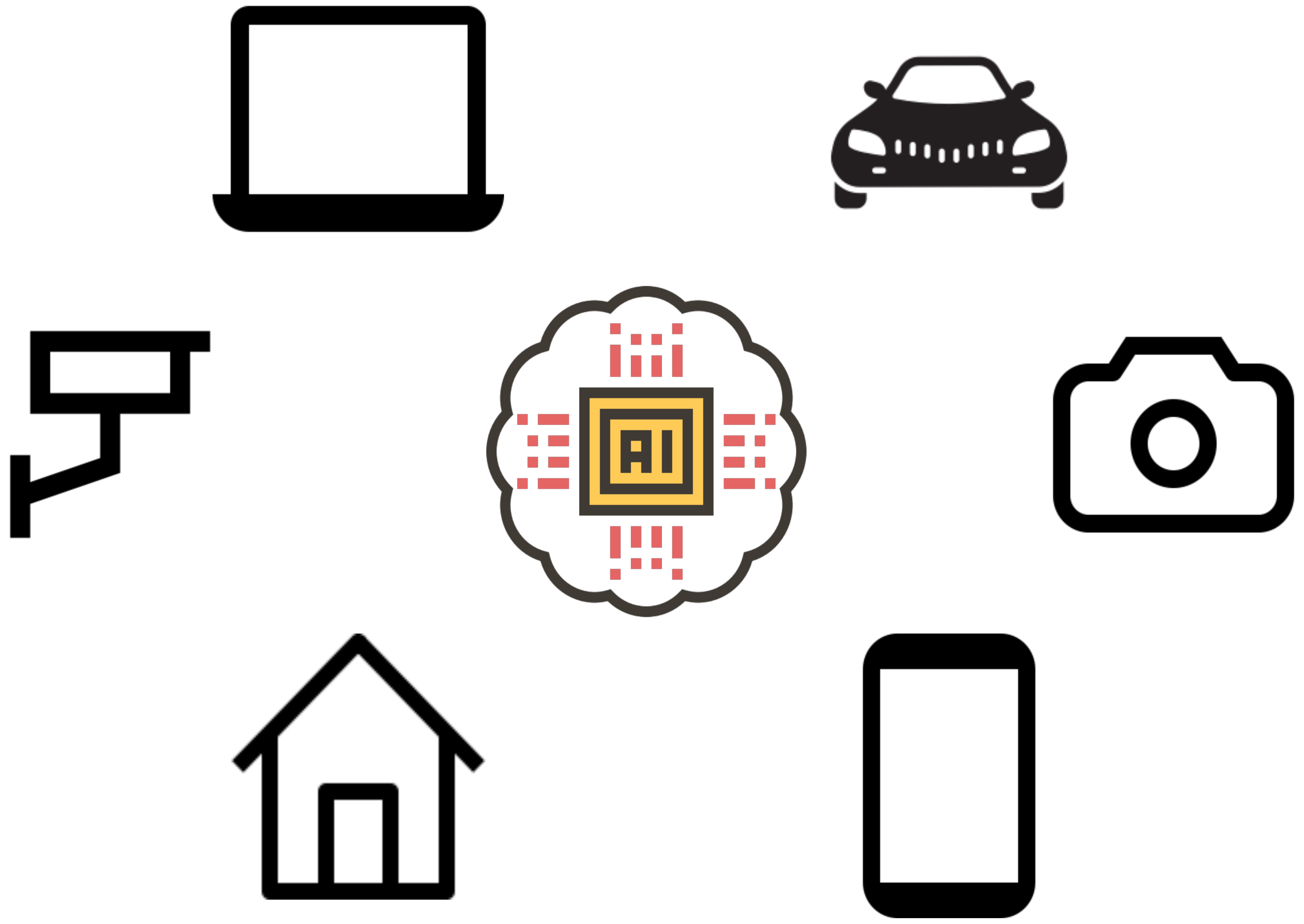
LLM Deployment in Production

- Cloud processing not always possible
  - Latency issue
  - Data privacy



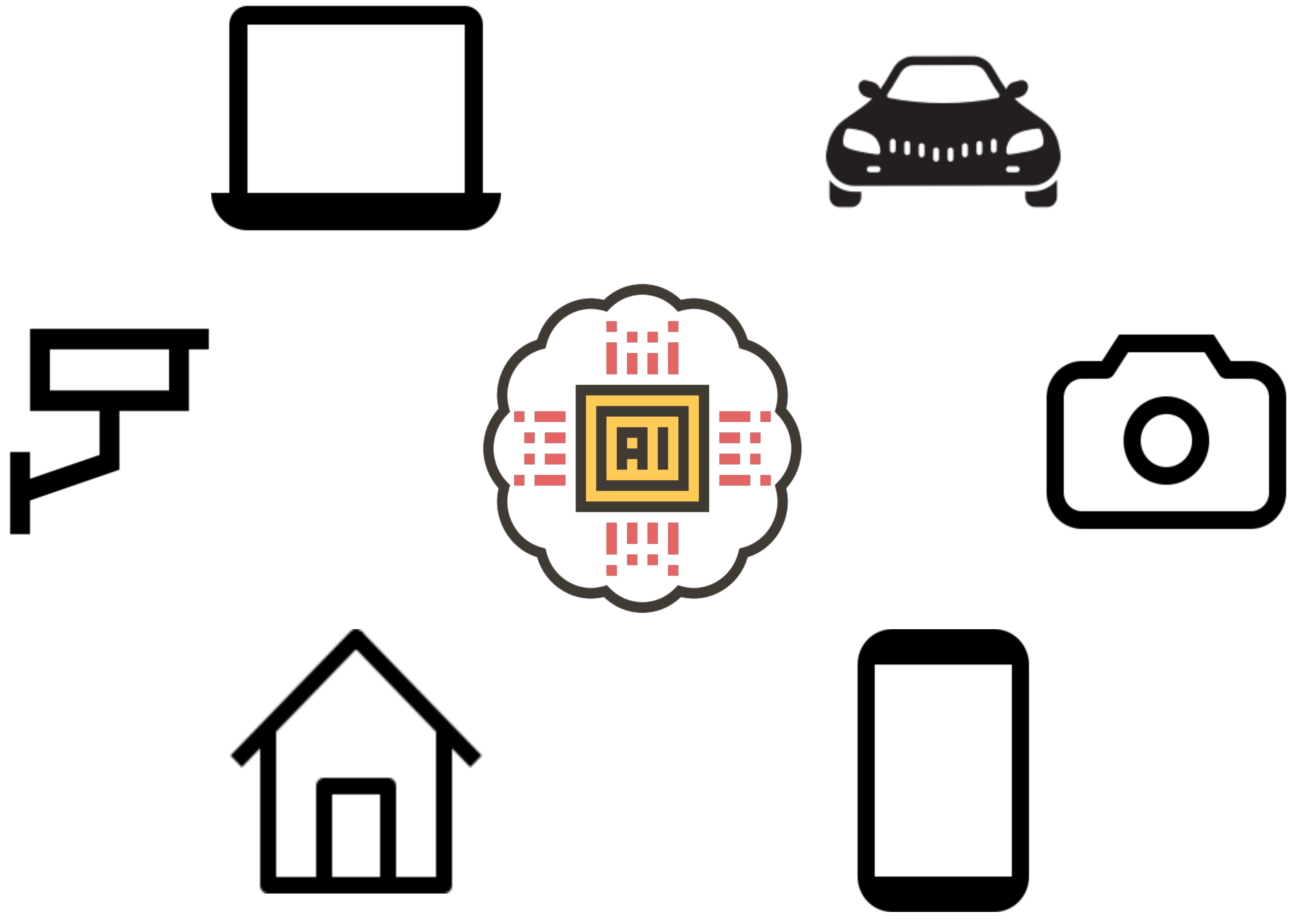
LLM Deployment in Production

- Cloud processing not always possible
  - Latency issue
  - Data privacy
- Inference time for edge devices



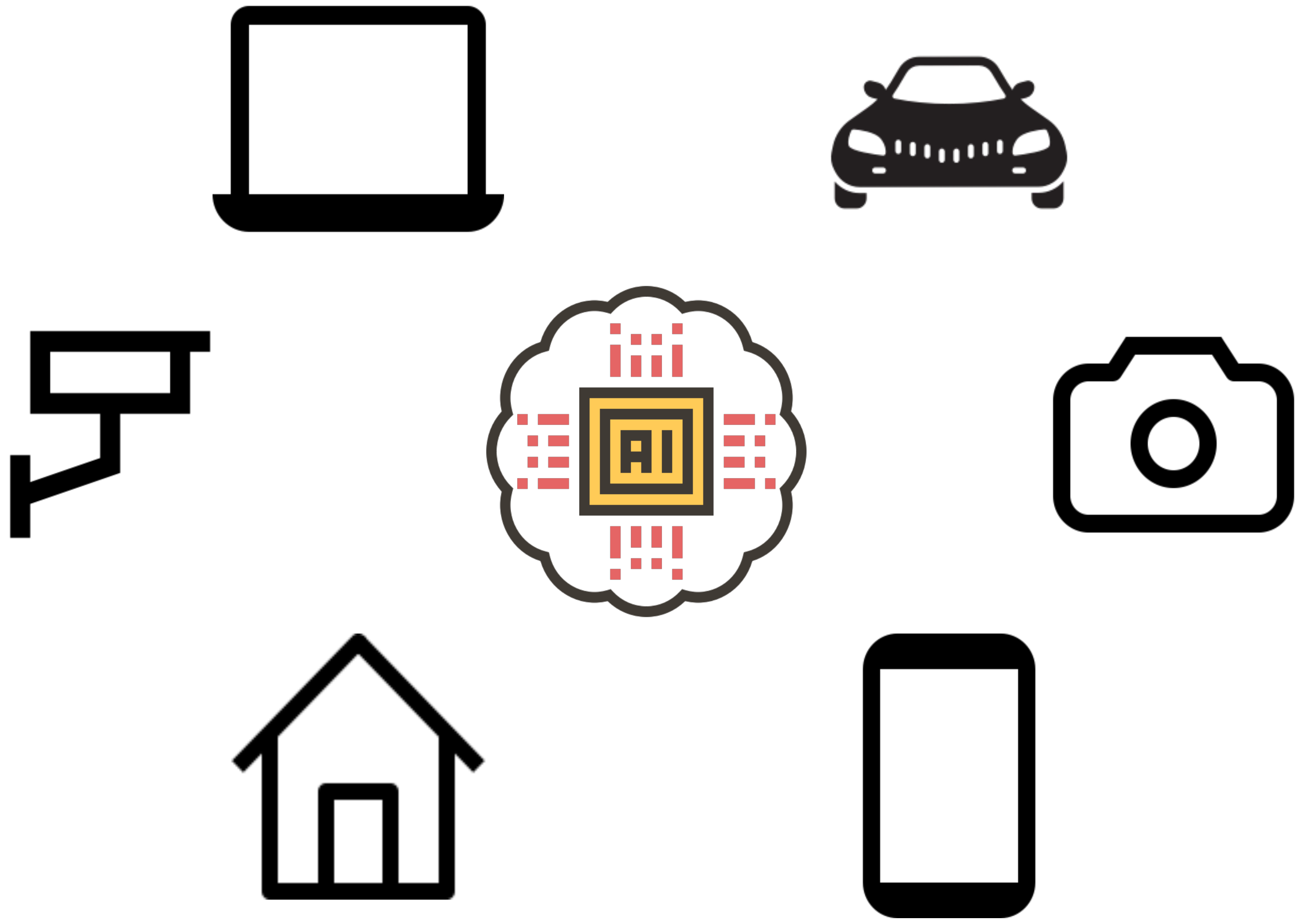
LLM Deployment in Production

- Cloud processing not always possible
  - Latency issue
  - Data privacy
- Inference time for edge devices
- Memory issue
  - ~350 GB just for storing LLM weights!



LLM Deployment in Production

- Cloud processing not always possible
  - Latency issue
  - Data privacy
- Inference time for edge devices
- Memory issue
  - ~350 GB just for storing LLM weights!
- Finetuning LLMs
  - Time-consuming
  - Expensive



What can we do instead?

Train smaller models!

Compression can reduce inference cost of deploying models!

Train Large, then Compress!

- Large models are more robust to compression techniques than small models

Li, Zhuohan et al. “Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers.” *ArXiv* abs/2002.11794 (2020)

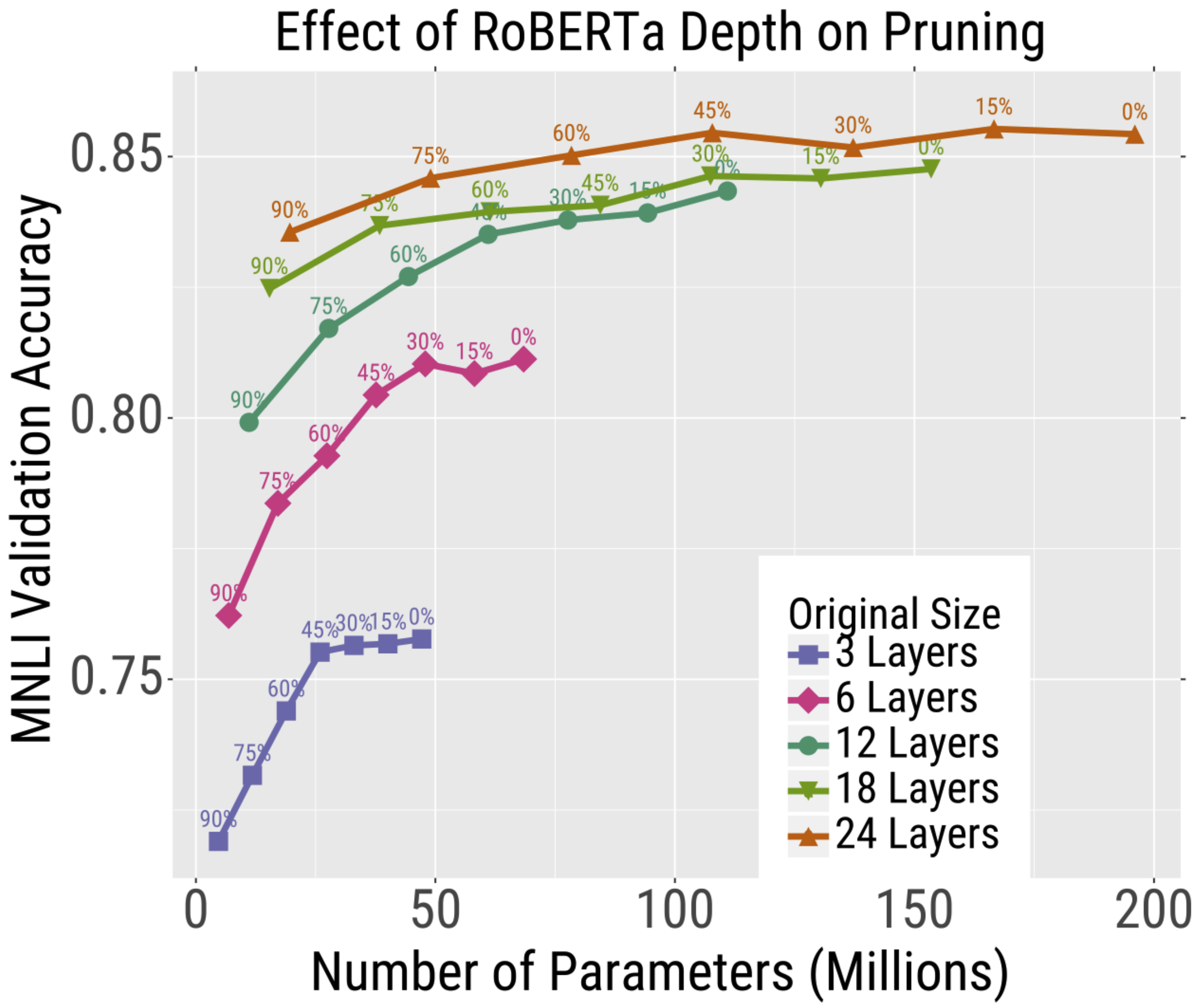
Train Large, then Compress!

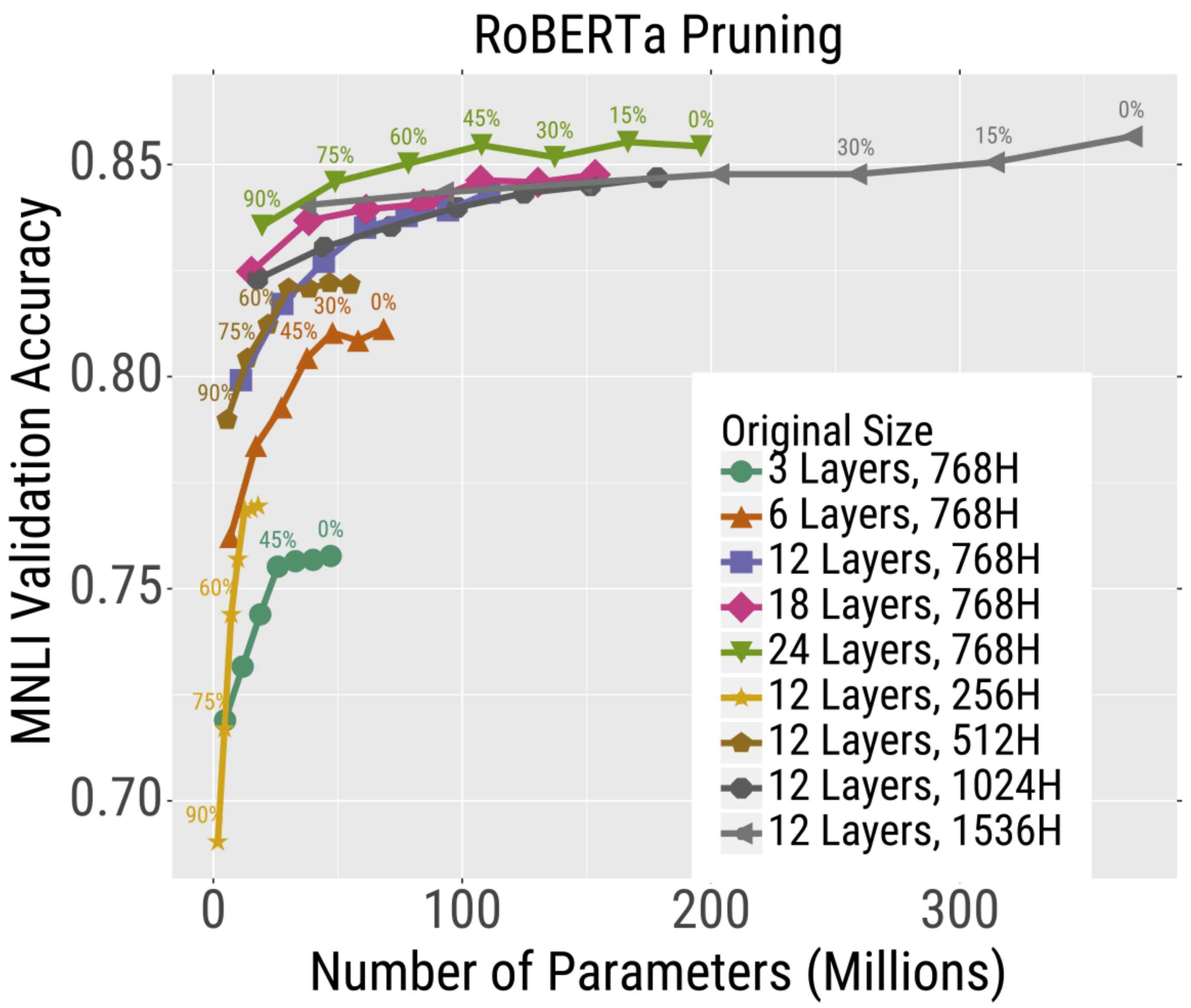
- Large models are more robust to compression techniques than small models
- For given test-time constraints (e.g., inference time, #parameter)
  - heavily compressed, large models > small models

Li, Zhuohan et al. "Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers." *ArXiv* abs/2002.11794 (2020)

Train Large, then Compress!

- Large models are more robust to compression techniques than small models
- For given test-time constraints (e.g., inference time, #parameter)
  - heavily compressed, large models > small models
- Comparing downstream task performance for discussed scenarios



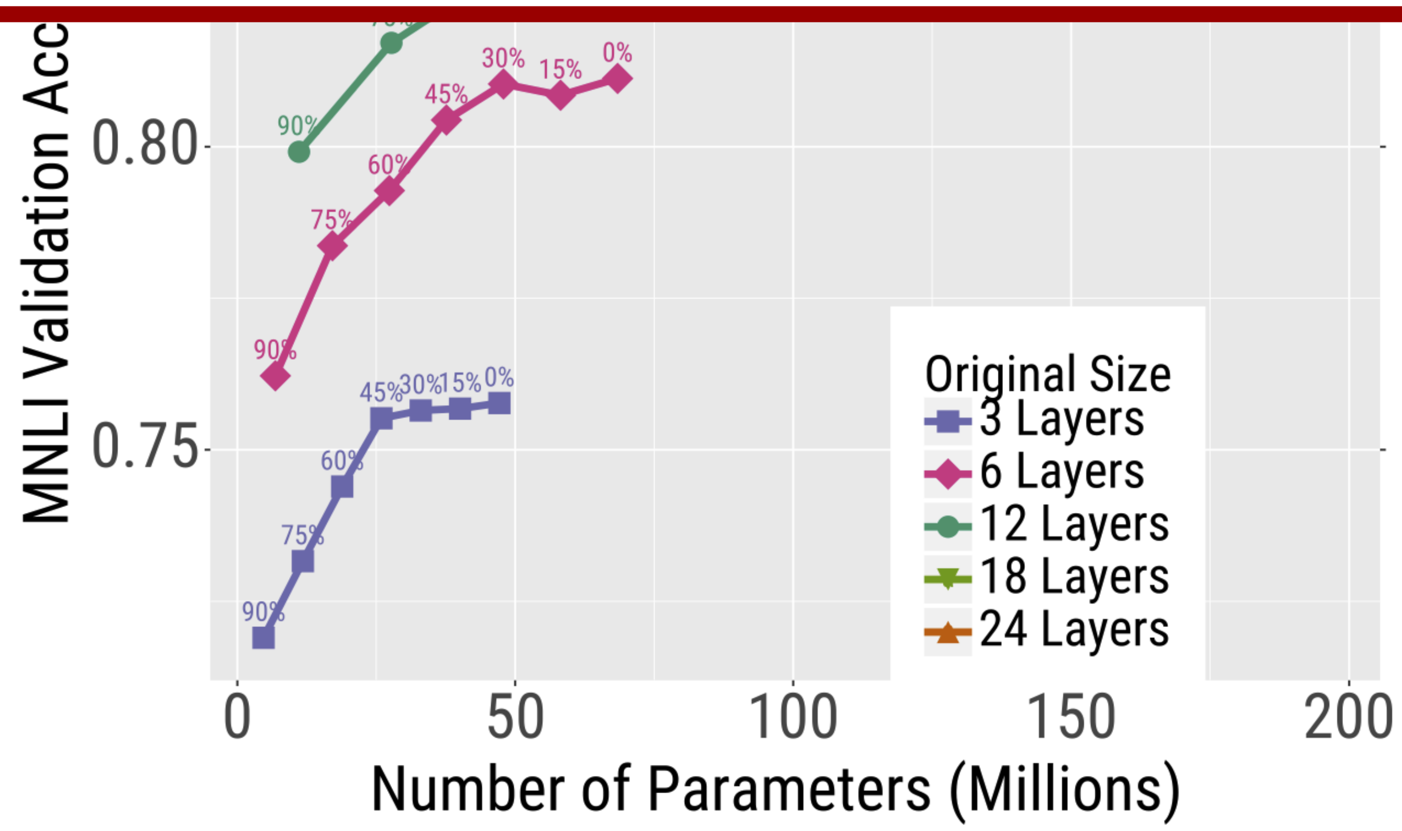


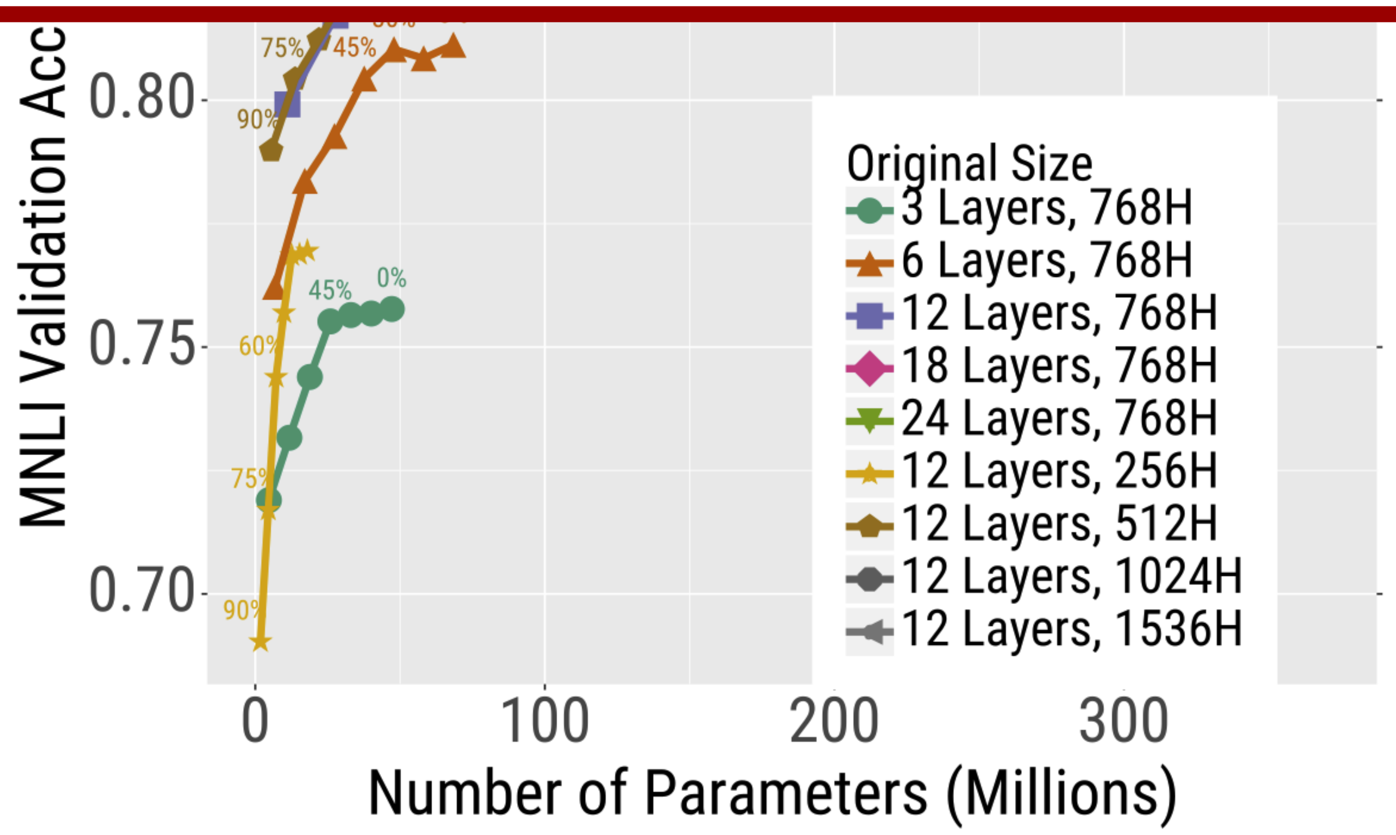
Li, Zhuohan et al. "Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers." *ArXiv* abs/2002.11794 (2020)

Train Large, then Compress!

- Large models are more robust to compression techniques than small models
- For given test-time constraints (e.g., inference time, #parameter)
  - heavily compressed, large models > small models
- Com

Compression improves the model's performance given a test-time budget!





Li, Zhuohan et al. "Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers." *ArXiv* abs/2002.11794 (2020)

How is compression done?

Compression Methods

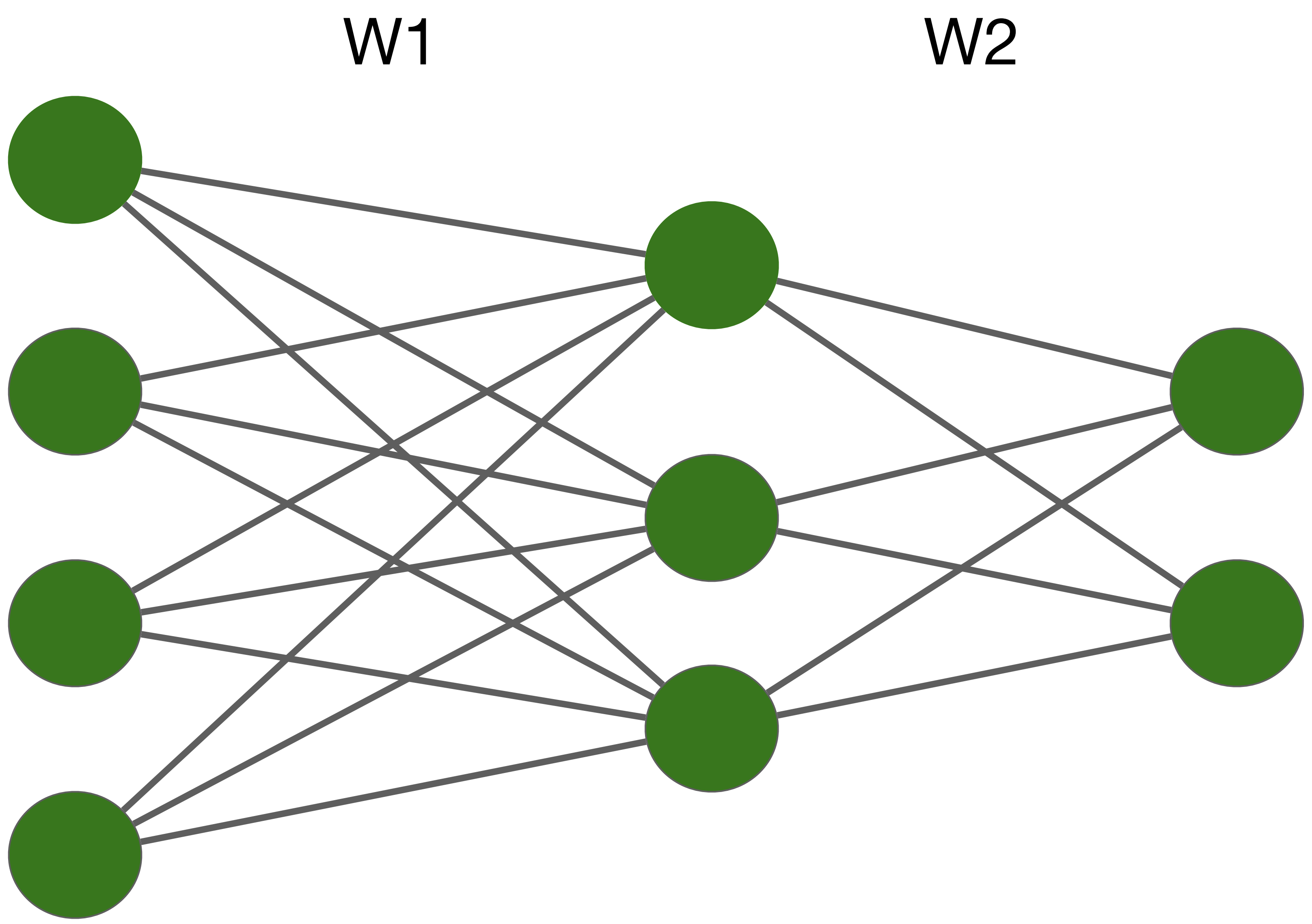
- Pruning
- Quantization
- Knowledge Distillation
- Speculative Decoding*

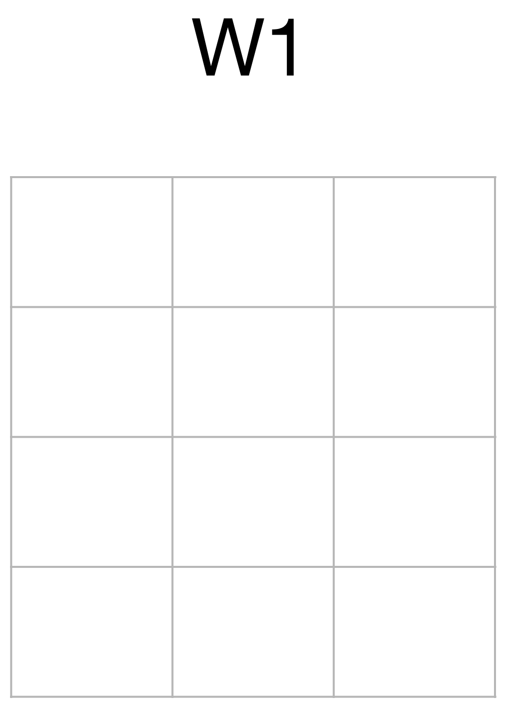
Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning		
Quantization		
Knowledge distillation		
Speculative Decoding		

Pruning

- Sparse connectivity inspired by biological neural networks
- Unstructured pruning Vs. structured pruning

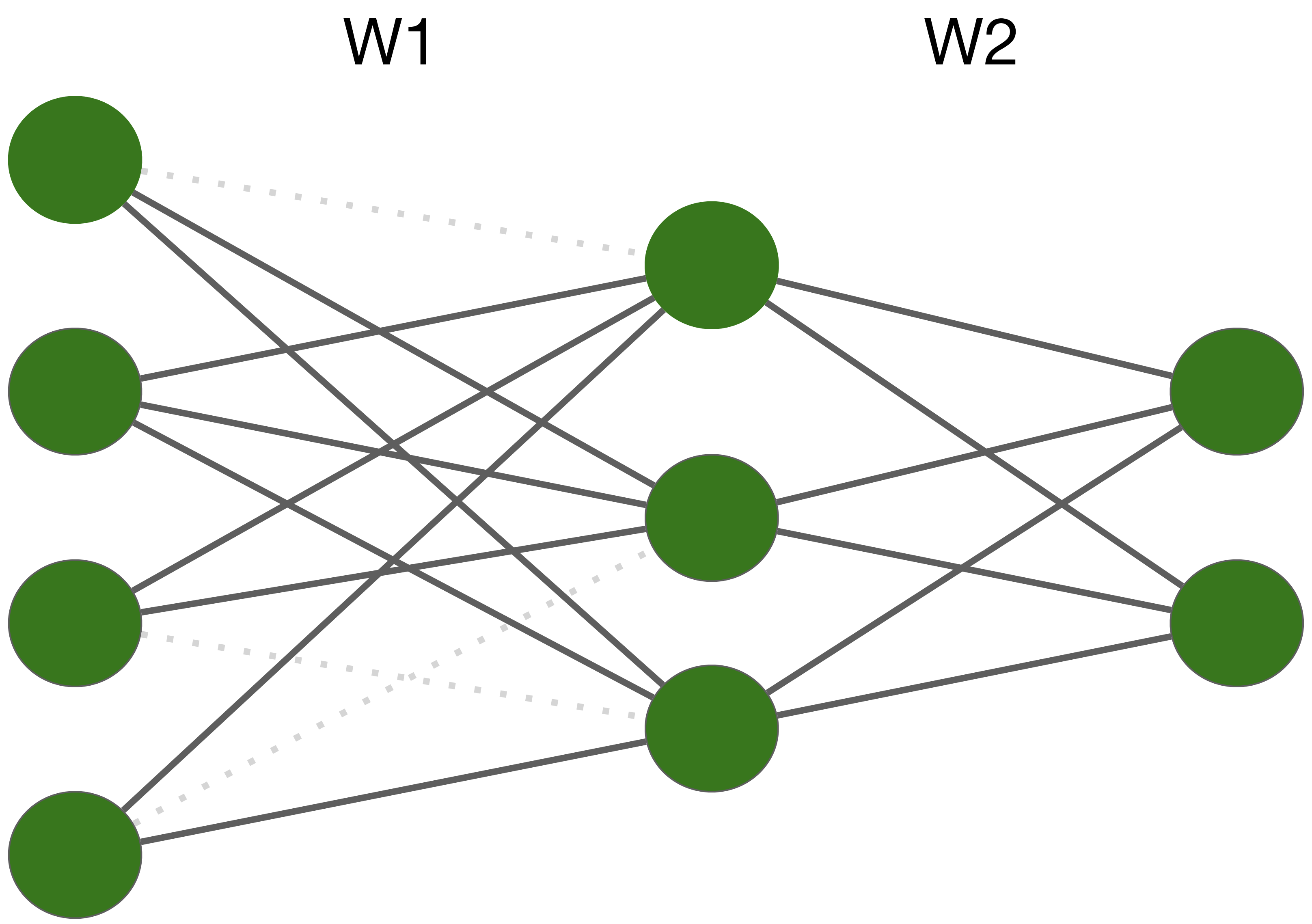


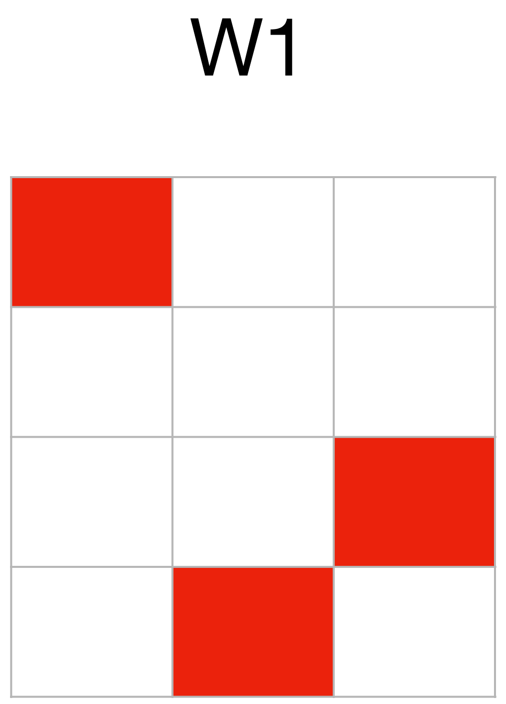


(No pruning)

Pruning

- Sparse connectivity inspired by biological neural networks
- Unstructured pruning (weight-level) Vs. structured pruning

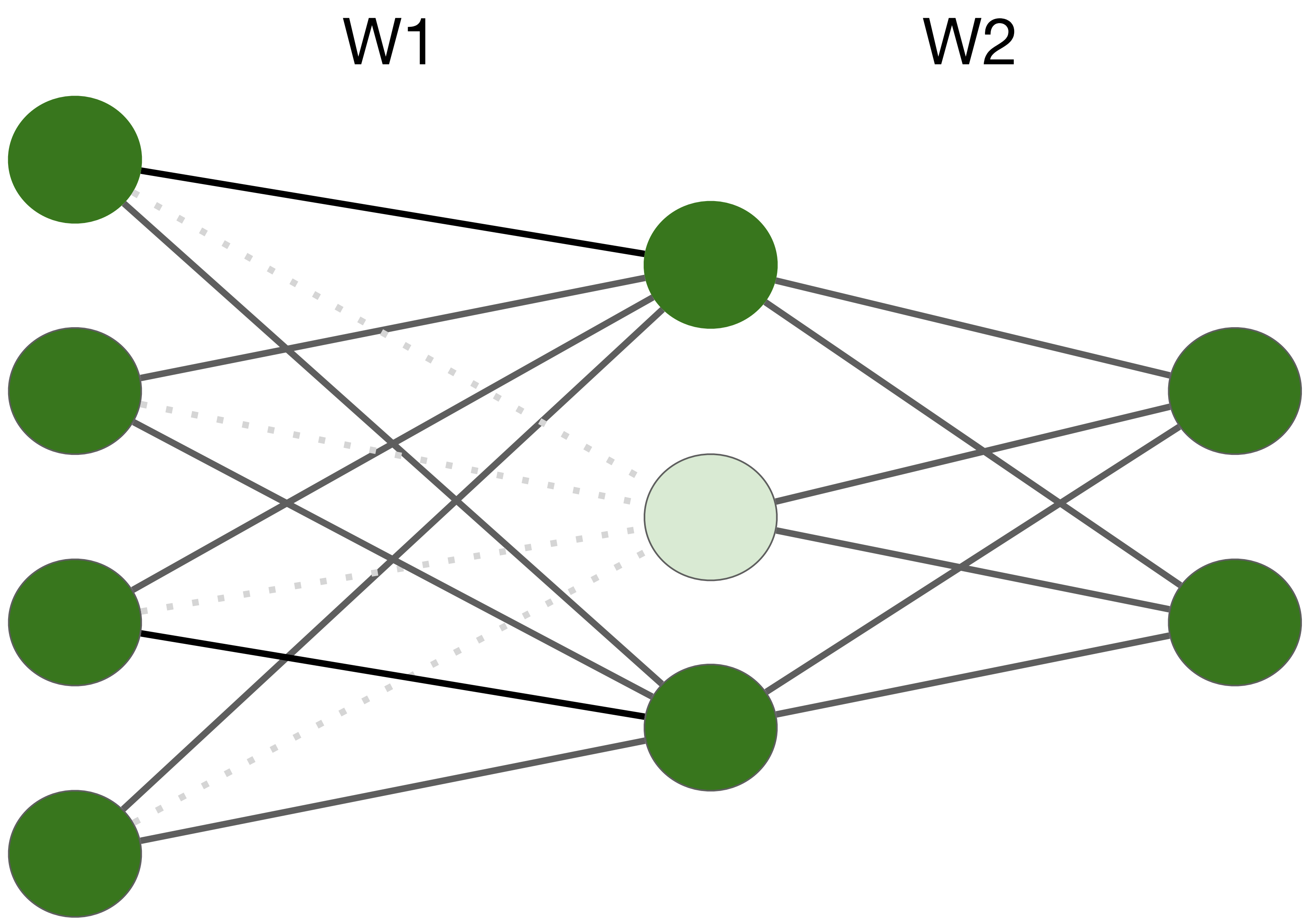


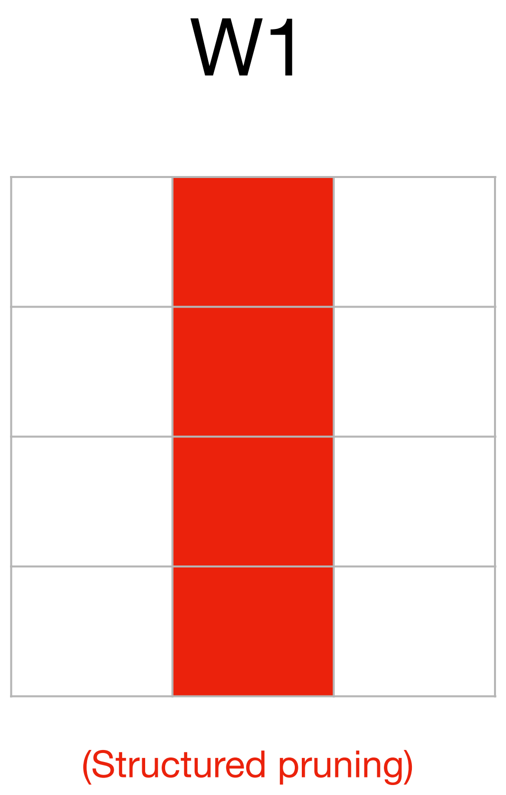


(Unstructured pruning)

Pruning

- Sparse connectivity inspired by biological neural networks
- Unstructured pruning Vs. structured pruning (module-level)

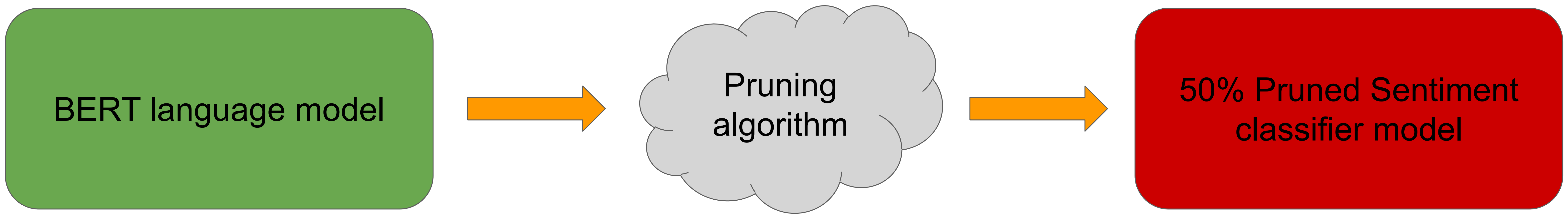




How to choose pruned weights?

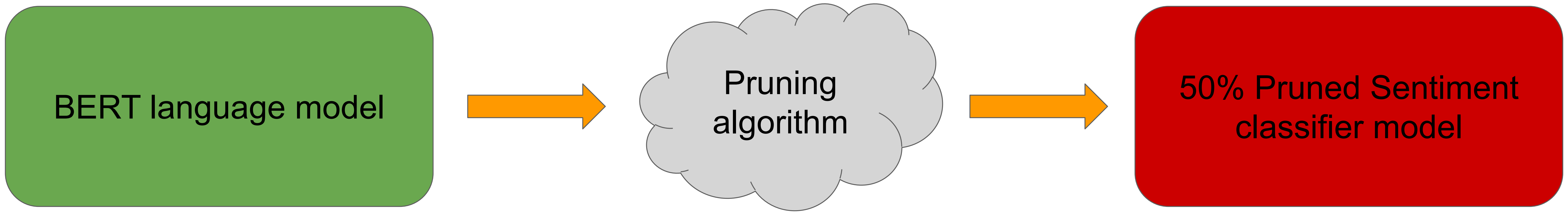
Pruning: case study

- Goal: a BERT-based sentiment classifier model
  - constraints: 50% of weights should be pruned



Pruning: case study

- Goal: a BERT-based sentiment classifier model
  - constraints: 50% of weights should be pruned

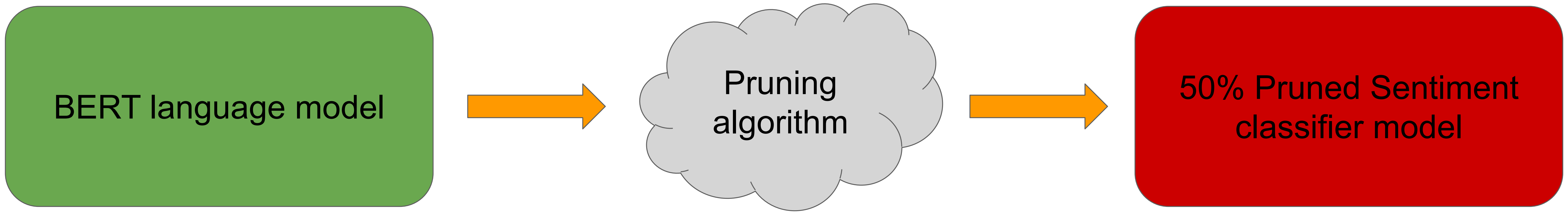


- which weights should be pruned?

$$\underbrace{\begin{bmatrix} 1 & 2 & 0.01 & 3 \\ -0.01 & -1 & -2 & -0.01 \\ -10 & -20 & 0.01 & -0.01 \end{bmatrix}}_{\text{Linear Layer}} \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \end{bmatrix}$$

Pruning: case study

- Goal: a BERT-based sentiment classifier model
  - constraints: 50% of weights should be pruned



- which weights should be pruned?

$$\underbrace{\begin{bmatrix} 1 & 2 & \color{red}{0.01} & 3 \\ \color{red}{-0.01} & -1 & -2 & \color{red}{-0.01} \\ -10 & -20 & \color{red}{0.01} & \color{red}{-0.01} \end{bmatrix}}_{\text{Linear Layer}} \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \end{bmatrix}$$

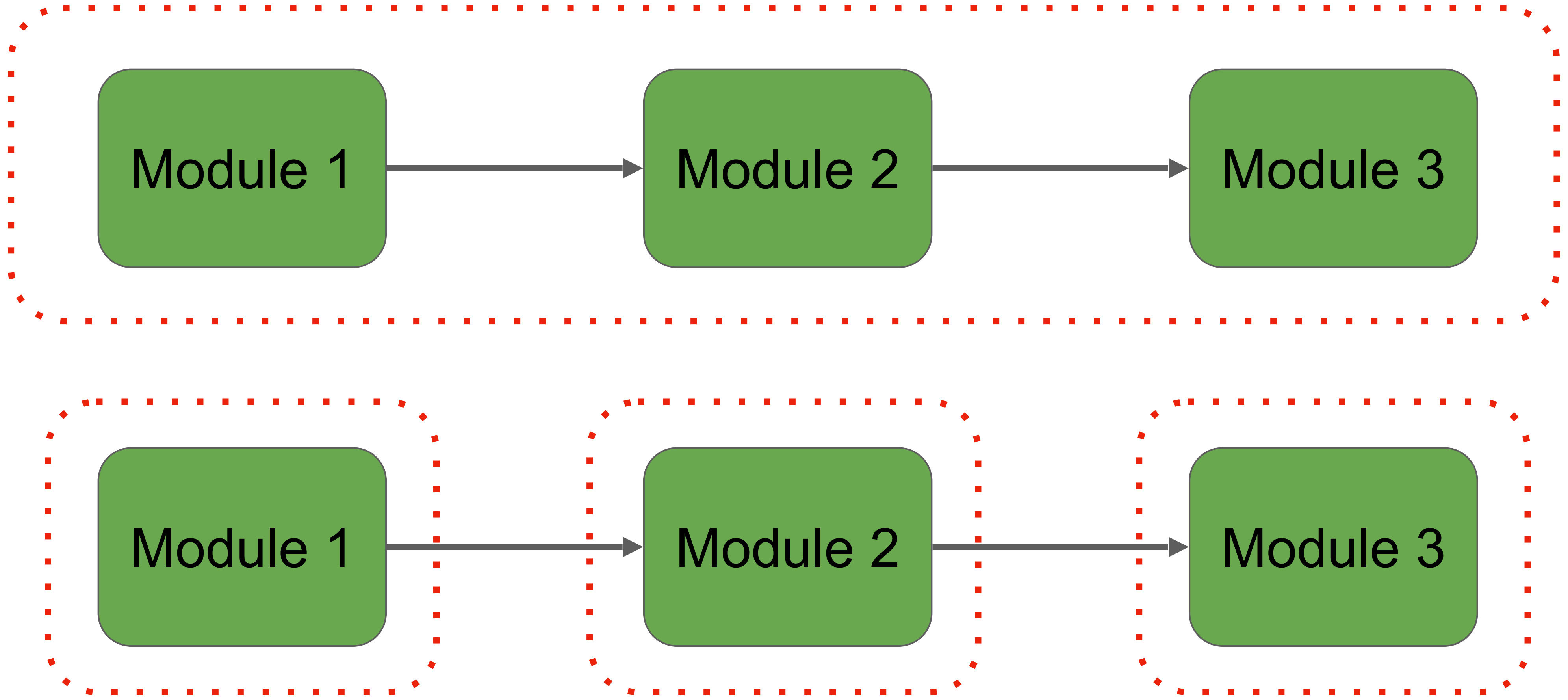
Weight Pruning Methods

- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level

Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).
Sanh, Victor, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning." *Advances in Neural Information Processing Systems* 33 (2020): 20378-20389.
Zhao, Mengjie, et al. "Masking as an efficient alternative to finetuning for pretrained language models." *arXiv preprint arXiv:2004.12406* (2020).

Weight Pruning Methods

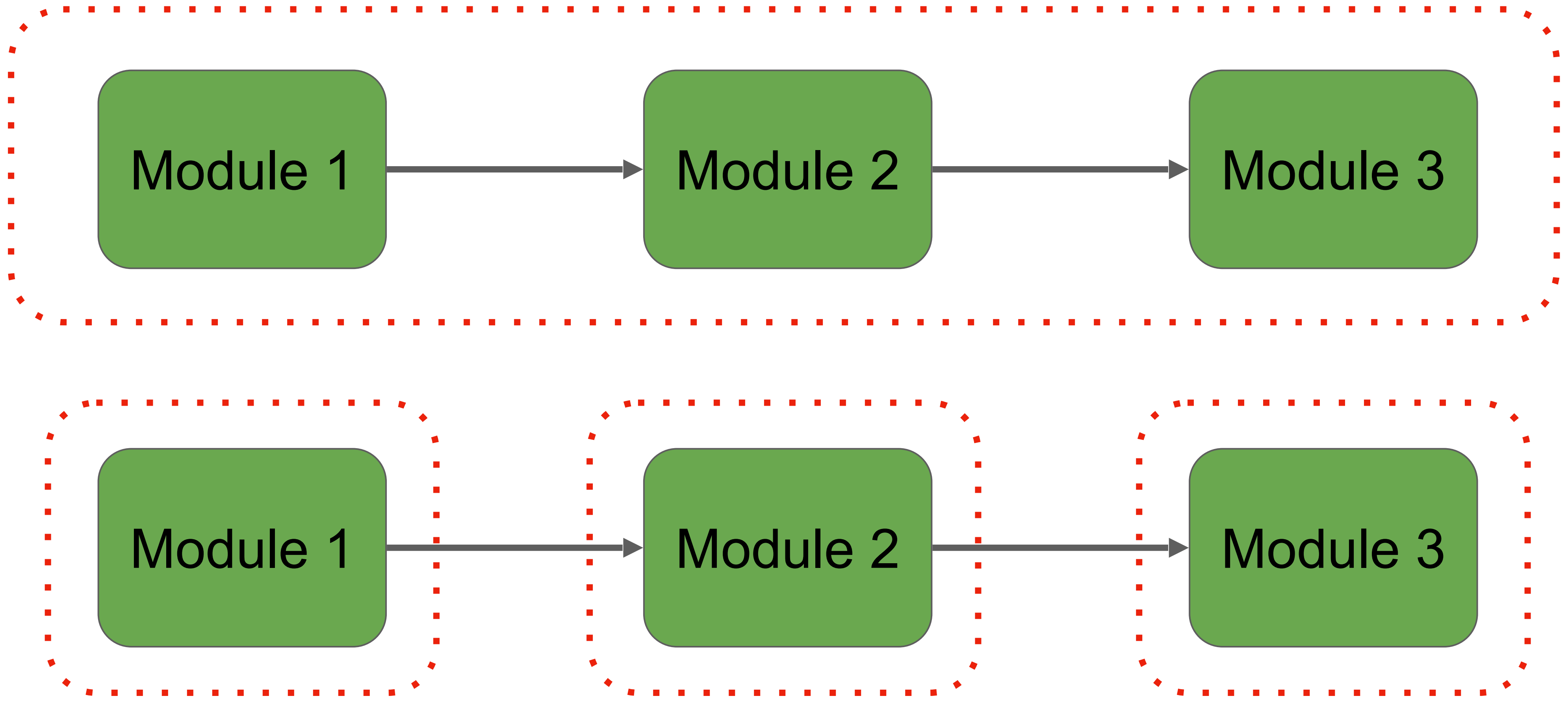
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level



Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).
Sanh, Victor, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning." *Advances in Neural Information Processing Systems* 33 (2020): 20378-20389.
Zhao, Mengjie, et al. "Masking as an efficient alternative to finetuning for pretrained language models." *arXiv preprint arXiv:2004.12406* (2020).

Weight Pruning Methods

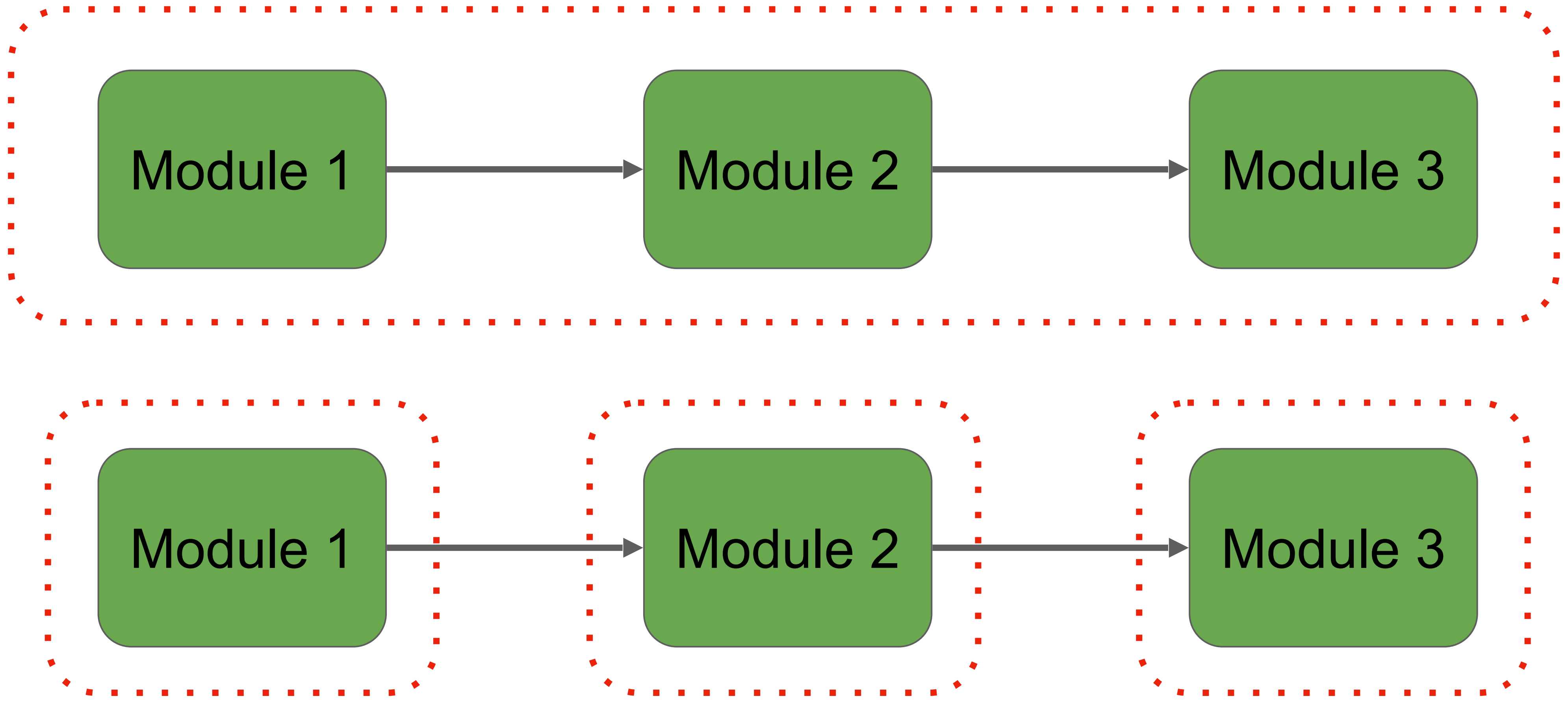
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training



Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).
Sanh, Victor, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning." *Advances in Neural Information Processing Systems* 33 (2020): 20378-20389.
Zhao, Mengjie, et al. "Masking as an efficient alternative to finetuning for pretrained language models." *arXiv preprint arXiv:2004.12406* (2020).

Weight Pruning Methods

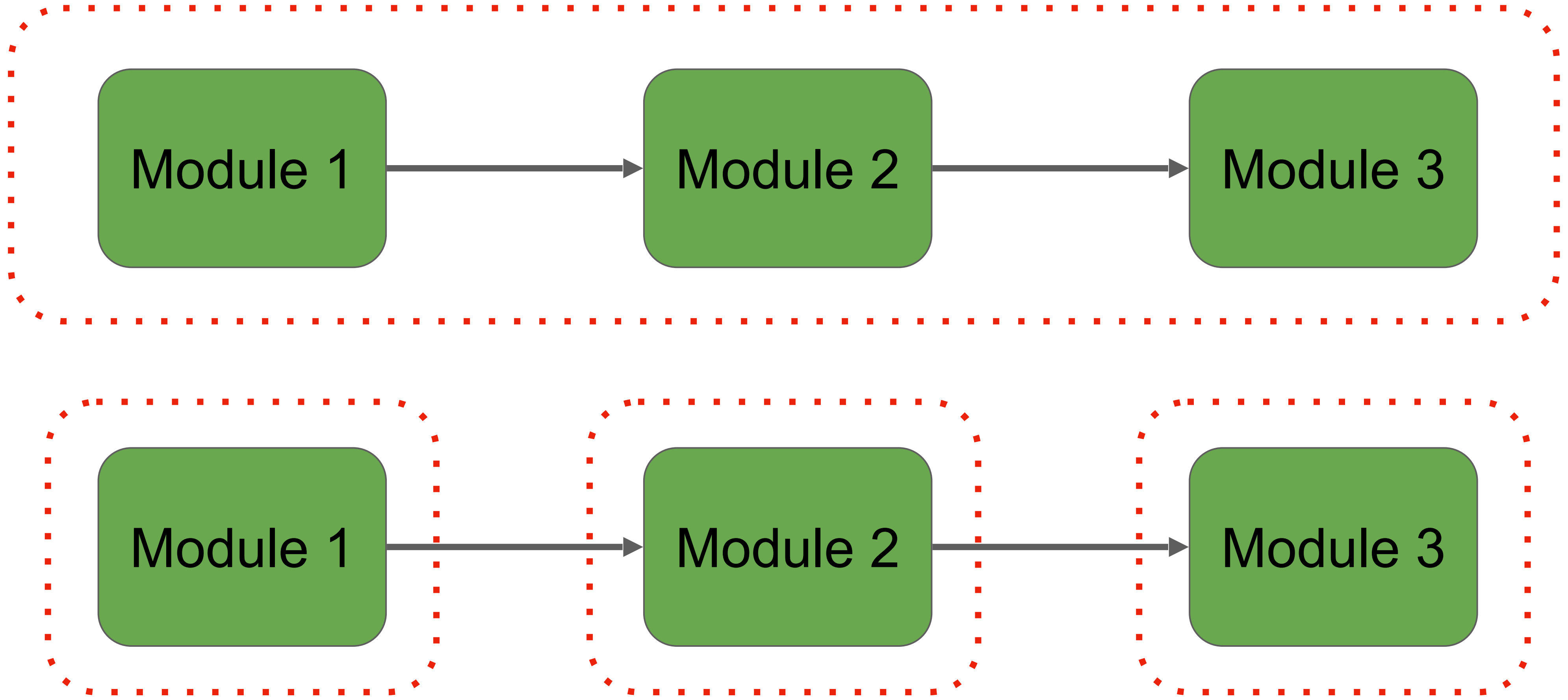
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training
- Movement pruning



Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).
Sanh, Victor, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning." *Advances in Neural Information Processing Systems* 33 (2020): 20378-20389.
Zhao, Mengjie, et al. "Masking as an efficient alternative to finetuning for pretrained language models." *arXiv preprint arXiv:2004.12406* (2020).

Weight Pruning Methods

- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training
- Movement pruning
- (Differentiable) masking as a pruning method
  - Example: attention head masking

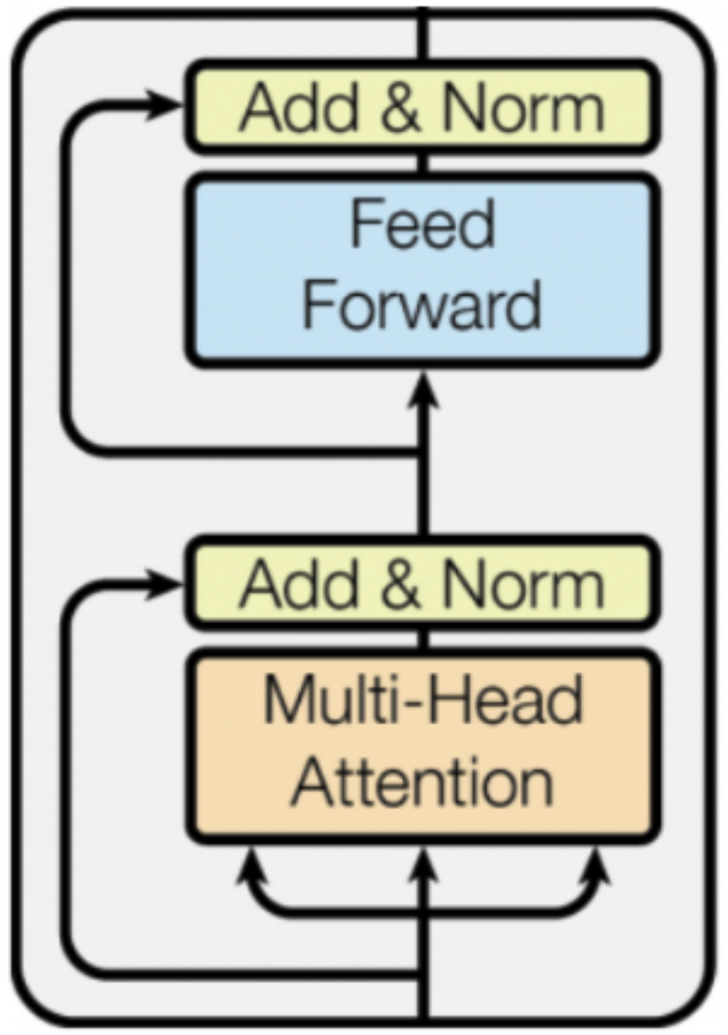


Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).
Sanh, Victor, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning." *Advances in Neural Information Processing Systems* 33 (2020): 20378-20389.
Zhao, Mengjie, et al. "Masking as an efficient alternative to finetuning for pretrained language models." *arXiv preprint arXiv:2004.12406* (2020).

What’s a shortcoming of unstructured pruning ?

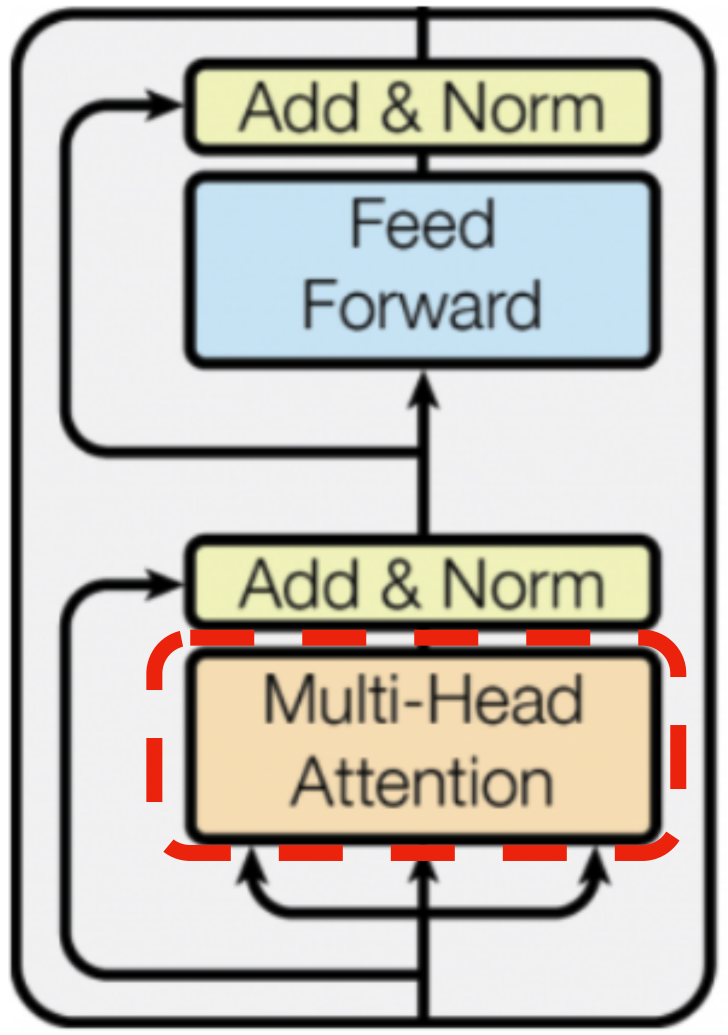
Structured Pruning

- Structured pruning for Transformer language models
  - Pruning neurons



Structured Pruning

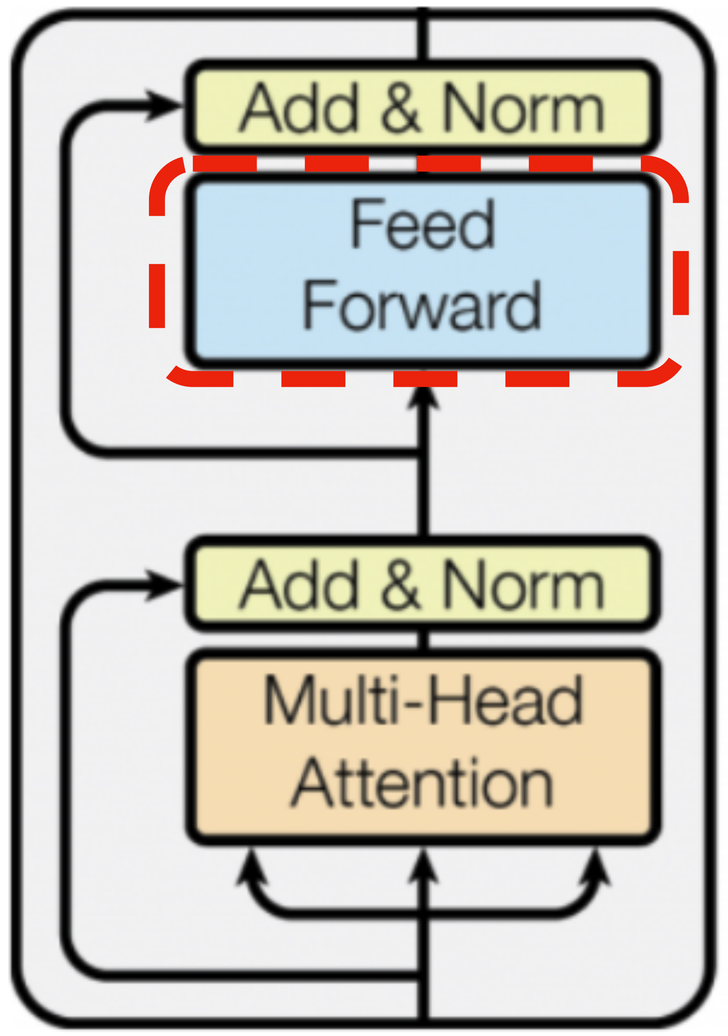
- Structured pruning for Transformer language models
  - Pruning neurons
  - Pruning attention heads



Michel, Paul, Omer Levy, and Graham Neubig. "Are sixteen heads really better than one?." *Advances in neural information processing systems* 32 (2019).
Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).

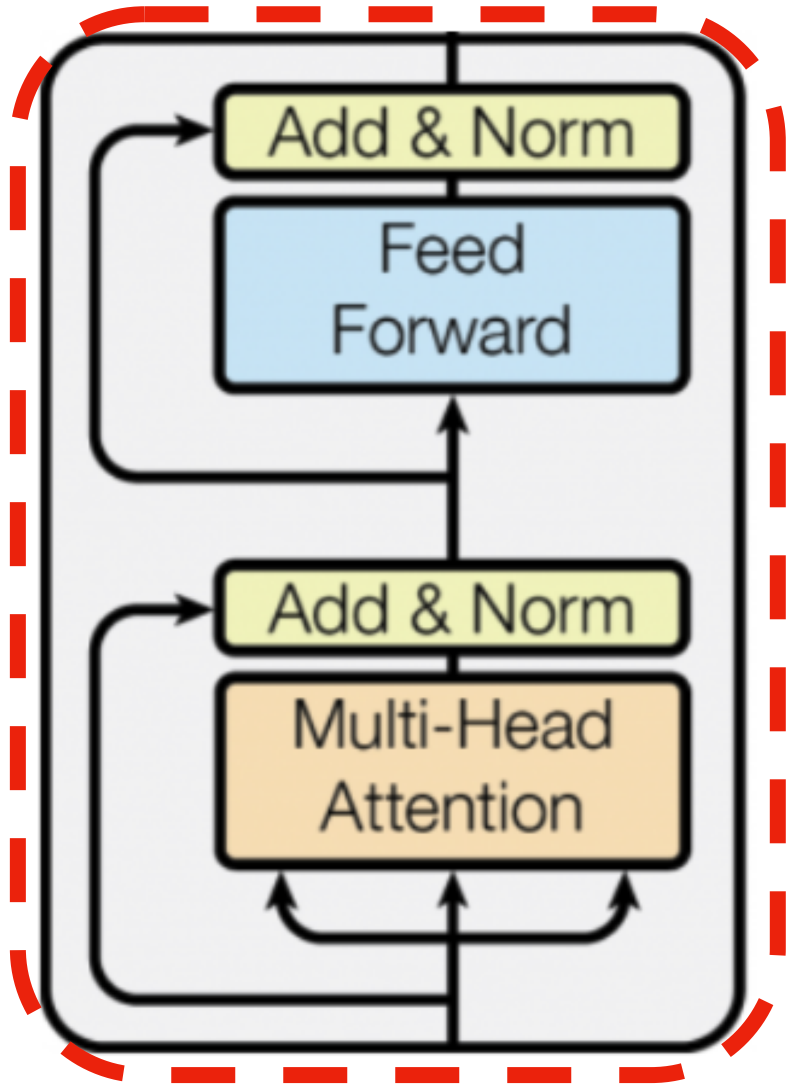
Structured Pruning

- Structured pruning for Transformer language models
  - Pruning neurons
  - Pruning attention heads
  - Pruning sub-layers
    - Example: pruning feed-forward sub-layer



Structured Pruning

- Structured pruning for Transformer language models
  - Pruning neurons
  - Pruning attention heads
  - Pruning sub-layers
    - Example: pruning feed-forward sub-layer
  - Pruning layers
    - Example: pruning the last K layers



Sajjad, Hassan, et al. "Poor man's bert: Smaller and faster transformer models." *arXiv preprint arXiv:2004.03844* 2.2 (2020).

Pruning Attention Heads

- How can we prune attention heads?

$$\text{MultiHead}(Q, K, V) = \text{Concat}_i(\text{head}_i)W^O$$

Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).
https://lena-voita.github.io/posts/acl19_heads.html

Pruning Attention Heads

- How can we prune attention heads?

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}_i(\mathrm{head}_i)W^O$$

$$\downarrow$$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}_i(\boxed{g_i} \cdot \mathrm{head}_i)W^O$$

Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).
https://lena-voita.github.io/posts/acl19_heads.html

Pruning Attention Heads

- How can we prune attention heads?

$$\text{MultiHead}(Q, K, V) = \text{Concat}_i(\text{head}_i)W^O$$

$$\downarrow$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}_i(\boxed{g_i} \cdot \text{head}_i)W^O$$

- L0 regularization over attention heads' mask parameters
  - Example: Translation task

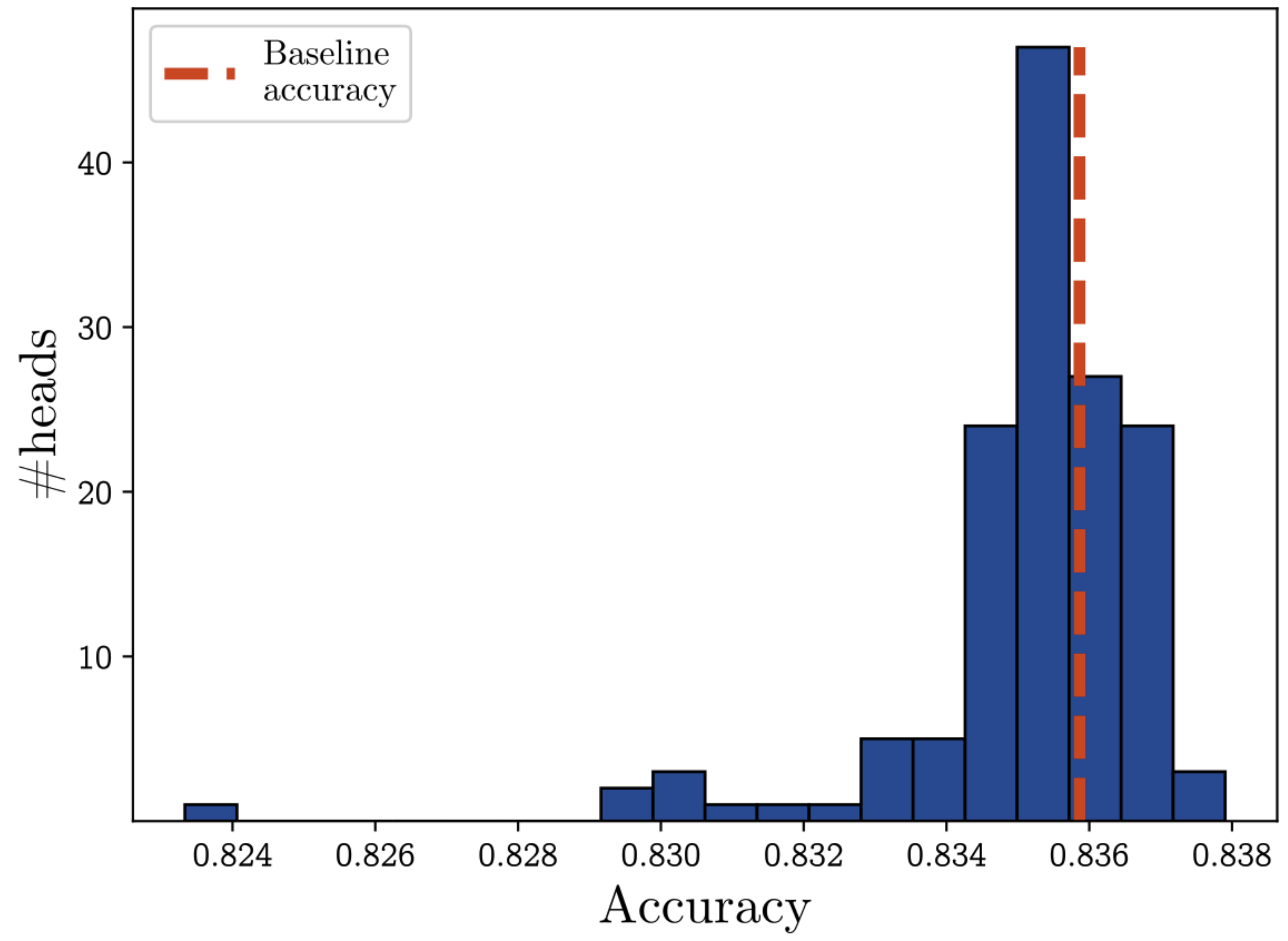
$$L = L_{xent} + \lambda L_C$$

$\lambda = 0.01$

Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).
https://lena-voita.github.io/posts/acl19_heads.html

Pruning Attention Heads

- Large fraction of Transformer attention heads can be removed at test time!



(BERT finetuned on MNLI dataset)

Layer		Layer	
1	-0.01%	7	0.05%
2	0.10%	8	-0.72%
3	-0.14%	9	-0.96%
4	-0.53%	10	0.07%
5	-0.29%	11	-0.19%
6	-0.52%	12	-0.12%

(Delta accuracy by layer when only one head is kept for MNLI BERT model)

Michel, Paul, Omer Levy, and Graham Neubig. "Are sixteen heads really better than one?." *Advances in neural information processing systems* 32 (2019).
Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).

Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning	Y/N	Y/N
Quantization		
Knowledge distillation		
Speculative Decoding		

Quantization

- How else can we compress a given neural module?

$$\underbrace{\begin{bmatrix} 1 & 2 & 0.01 & 3 \\ -0.01 & -1 & -2 & -0.01 \\ -10 & -20 & 0.01 & -0.01 \end{bmatrix}}_{\text{Linear Layer}} \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \end{bmatrix}$$

Quantization

- How else can we compress a given neural module?

32 bits

Reduce number of bits to store weights!

$$\underbrace{\begin{pmatrix} \boxed{1} & 2 & 0.01 & 3 \\ -0.01 & -1 & -2 & -0.01 \\ -10 & -20 & 0.01 & -0.01 \end{pmatrix}}_{\text{Linear Layer}} \begin{pmatrix} X1 \\ X2 \\ X3 \\ X4 \end{pmatrix}$$

Quantization

- How else can we compress a given neural module?

$$\underbrace{\begin{bmatrix} \boxed{1} & 2 & 0.01 & 3 \\ -0.01 & -1 & -2 & -0.01 \\ -10 & -20 & 0.01 & -0.01 \end{bmatrix}}_{\text{Linear Layer}} \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \end{bmatrix}$$

32 bits

Reduce number of bits to store weights!

- Number of parameters remains the same!
  - Improvement in memory footprint + inference time
- Quantization is mostly applied on a trained model

Binarized Network

- Essentially using 1 bit per parameter!
- Deterministic Binarization
  - c1 and c2 from K-means over the weights
  - c1 and c2 tuned on downstream task

$$w_b = \begin{cases} c_1 & \text{if } w \geq (c_1 + c_2)/2 \\ c_2 & \text{if } w < (c_1 + c_2)/2 \end{cases}$$

Gupta, Manish, and Puneet Agrawal. "Compression of deep learning models for text: A survey." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.4 (2022): 1-55.
Cheong, Robin, and Robel Daniel. "transformers. zip: Compressing Transformers with Pruning and Quantization." *Technical report, tech. rep., Stanford University, Stanford, California* (2019).

Binarized Network

- Essentially using 1 bit per parameter!
- Deterministic Binarization
  - c1 and c2 from K-means over the weights
  - c1 and c2 tuned on downstream task

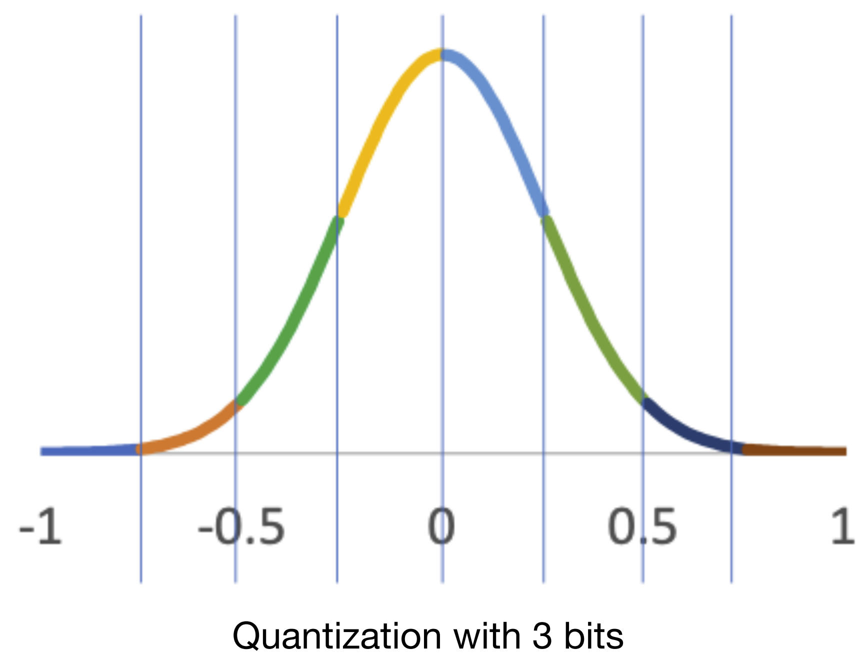
$$w_b = \begin{cases} c_1 & \text{if } w \geq (c_1 + c_2)/2 \\ c_2 & \text{if } w < (c_1 + c_2)/2 \end{cases}$$

- Question: How can we improve the binarized network performance?

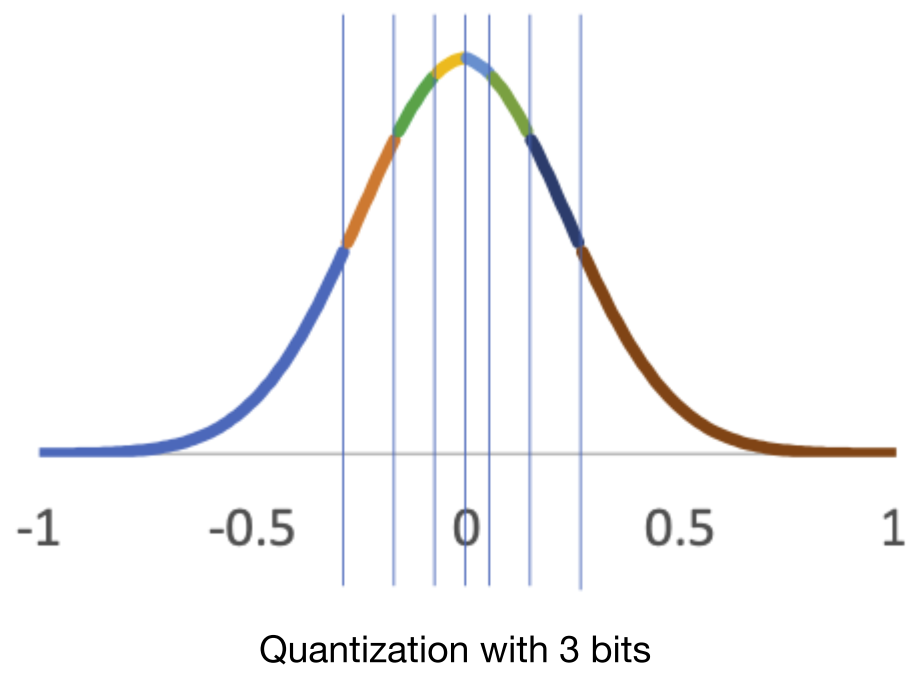
Gupta, Manish, and Puneet Agrawal. "Compression of deep learning models for text: A survey." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.4 (2022): 1-55.
Cheong, Robin, and Robel Daniel. "transformers. zip: Compressing Transformers with Pruning and Quantization." *Technical report, tech. rep., Stanford University, Stanford, California* (2019).

General Quantized Networks

- Uniform Quantization
  - Not necessarily optimal



- Balanced Quantization
  - Better fitted for non-uniform weights!
  - Example: Decide bin boundaries using clustering!



Gupta, Manish, and Puneet Agrawal. "Compression of deep learning models for text: A survey." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.4 (2022): 1-55..

Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning	Y/N	Y/N
Quantization	Yes	Yes
Knowledge Distillation		
Speculative Decoding		

Knowledge Distillation

- Training a smaller student network by distilling a large teacher model
    - The student's goal is to imitate teacher's behavior!
- Can we have the best of the two worlds?
    - Good performance of teacher model + faster & parameter-efficient student model

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- Training a smaller student network by distilling a large teacher model
  - The student's goal is to imitate teacher's behavior!
- Can we have the best of the two worlds?
  - Good performance of teacher model + faster & parameter-efficient student model
- Knowledge distillation Vs. Transfer learning
  - Transfer learning → deals with shared architecture/layers
  - Knowledge distillation → often the student model has a different smaller architecture

How can we distill the teacher's knowledge?

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- Intuition behind knowledge distillation

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- Intuition behind knowledge distillation
- Consider a 3-class sentiment analysis dataset
  - We pass the following 2 samples to the teacher model to get class probabilities

Sample #1

	Positive	Negative	Neutral
Groundtruth	1	0	0
Teacher prob.	0.94	0.01	0.05

Sample #2

Soft Labels

	Positive	Negative	Neutral
Groundtruth	1	0	0
Teacher prob.	0.67	0.02	0.31

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- How to leverage soft labels for the student model?
  - Additional cross-entropy to soft labels (soft loss)
  - Cross-entropy loss to ground-truth labels → hard loss

$$\mathcal{L} = \alpha \cdot \underbrace{\mathcal{L}_{\text{CE}}}_{\text{Hard Loss}} + (1-\alpha) \cdot \underbrace{\mathcal{L}_{\text{distill}}}_{\text{Soft Loss}}$$

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015)
Tang, Raphael, et al. "Distilling task-specific knowledge from bert into simple neural networks." *arXiv preprint arXiv:1903.12136* (2019)

Knowledge Distillation

- How to leverage soft labels for the student model?
  - Additional cross-entropy to soft labels (soft loss)
  - Cross-entropy loss to ground-truth labels → hard loss

$$\mathcal{L} = \alpha \cdot \underbrace{\mathcal{L}_{\text{CE}}}_{\text{Hard Loss}} + (1-\alpha) \cdot \underbrace{\mathcal{L}_{\text{distill}}}_{\text{Soft Loss}}$$

- Teacher making confident prediction for easy downstream tasks
  - Solution: increase softmax temperature to get suitably soft targets!

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015)
Tang, Raphael, et al. "Distilling task-specific knowledge from bert into simple neural networks." *arXiv preprint arXiv:1903.12136* (2019)

Knowledge Distillation

- How to leverage soft labels for the student model?
  - Additional cross-entropy to soft labels (soft loss)
  - Cross-entropy loss to ground-truth labels → hard loss

$$\mathcal{L} = \alpha \cdot \underbrace{\mathcal{L}_{\text{CE}}}_{\text{Hard Loss}} + (1-\alpha) \cdot \underbrace{\mathcal{L}_{\text{distill}}}_{\text{Soft Loss}}$$

- Teacher making confident prediction for easy downstream tasks
  - Solution: increase softmax temperature to get suitably soft targets!

#	Model	SST-2	QQP	MNLI-m	MNLI-mm
		Acc	F_1/Acc	Acc	Acc
1	$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2018)	94.9	72.1/89.3	86.7	85.9
2	$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018)	93.5	71.2/89.2	84.6	83.4
3	OpenAI GPT (Radford et al., 2018)	91.3	70.3/88.5	82.1	81.4
4	BERT ELMo baseline (Devlin et al., 2018)	90.4	64.8/84.7	76.4	76.1
5	GLUE ELMo baseline (Wang et al., 2018)	90.4	63.1/84.3	74.1	74.5
6	Distilled $\text{BiLSTM}_{\text{SOFT}}$	**90.7**	**68.2/88.1**	**73.0**	**72.6**
7	BiLSTM (our implementation)	86.7	63.7/86.2	68.7	68.3

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015)
Tang, Raphael, et al. "Distilling task-specific knowledge from bert into simple neural networks." *arXiv preprint arXiv:1903.12136* (2019)

Case Study: distilBERT

- 6-layer student model distilled from BERT-base (i.e., teacher)
  - Initialize the student from the teacher by taking one layer out of two

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Case Study: distilBERT

- 6-layer student model distilled from BERT-base (i.e., teacher)
  - Initialize the student from the teacher by taking one layer out of two
- Distillation on MLM loss
  - Improving LM generalization

I absolutely [MASK] natural language processing field. $\xrightarrow{\text{BERT-base}}$

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Case Study: distilBERT

- 6-layer student model distilled from BERT-base (i.e., teacher)
  - Initialize the student from the teacher by taking one layer out of two
- Distillation on MLM loss
  - Improving LM generalization

I absolutely [MASK] natural language processing field.

BERT-base

```
0.241 I absolutely hate natural language processing field.
0.154 I absolutely love natural language processing field.
0.045 I absolutely need natural language processing field.
0.041 I absolutely mean natural language processing field.
0.040 I absolutely missed natural language processing field.
0.034 I absolutely hated natural language processing field.
0.032 I absolutely understand natural language processing field.
0.024 I absolutely loved natural language processing field.
0.023 I absolutely like natural language processing field.
0.020 I absolutely miss natural language processing field.
```

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Case Study: distilBERT

- 6-layer student model distilled from BERT-base (i.e., teacher)
  - Initialize the student from the teacher by taking one layer out of two
- Distillation on MLM loss
  - Improving LM generalization

I absolutely [MASK] natural language processing field. $\xrightarrow{\text{BERT-base}}$

```
0.241 I absolutely hate natural language processing field.
0.154 I absolutely love natural language processing field.
0.045 I absolutely need natural language processing field.
0.041 I absolutely mean natural language processing field.
0.040 I absolutely missed natural language processing field.
0.034 I absolutely hated natural language processing field.
0.032 I absolutely understand natural language processing field.
0.024 I absolutely loved natural language processing field.
0.023 I absolutely like natural language processing field.
0.020 I absolutely miss natural language processing field.
```

- Proposed Loss: MLM + distilling BERT ML

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Case Study: distilBERT

- 6-layer student model distilled from BERT-base (i.e., teacher)
  - Initialize the student from the teacher by taking one layer out of two
- Distillation on MLM loss
  - Improving LM generalization

I absolutely [MASK] natural language processing field. → BERT-base →

```
0.241 I absolutely hate natural language processing field.
0.154 I absolutely love natural language processing field.
0.045 I absolutely need natural language processing field.
0.041 I absolutely mean natural language processing field.
0.040 I absolutely missed natural language processing field.
0.034 I absolutely hated natural language processing field.
0.032 I absolutely understand natural language processing field.
0.024 I absolutely loved natural language processing field.
0.023 I absolutely like natural language processing field.
0.020 I absolutely miss natural language processing field.
```

- Proposed Loss: MLM + distilling BERT ML
- Competitive performance to the teacher

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning	Y/N	Y/N
Quantization	Yes	Yes
Knowledge distillation	Yes…	Yes…
Speculative Decoding		

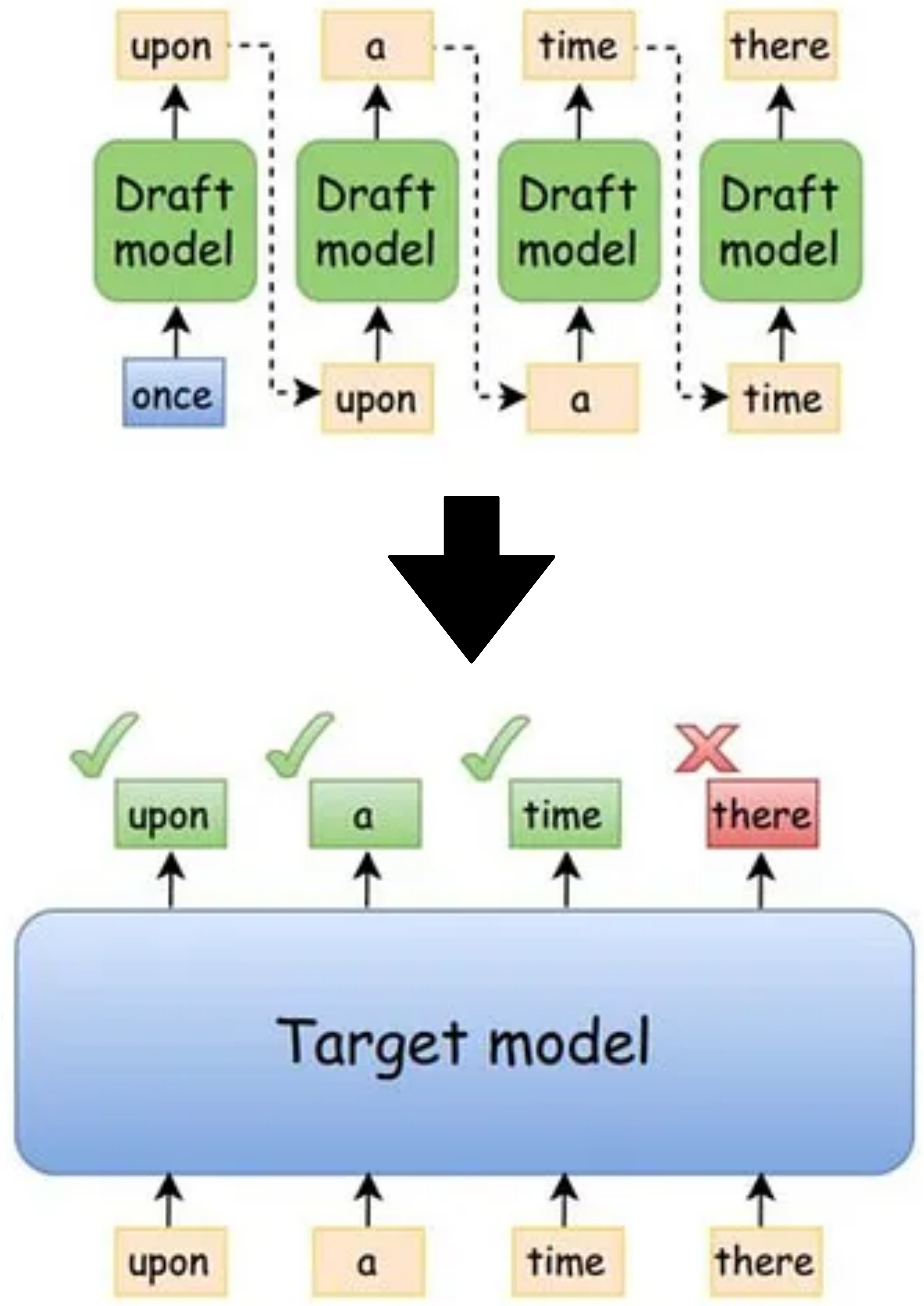
Speculative Decoding

- Large models have a much higher decoding cost during inference
  - full forward pass for every token generated !
- **Solution:** use a small model to generate candidate sequences, and **verify** that the large model would have also generated the same sequences
  - smaller model performs full forward pass for every token generated, and larger model only does forward pass in verification steps.

Leviathan et al., 2022

Speculative Decoding

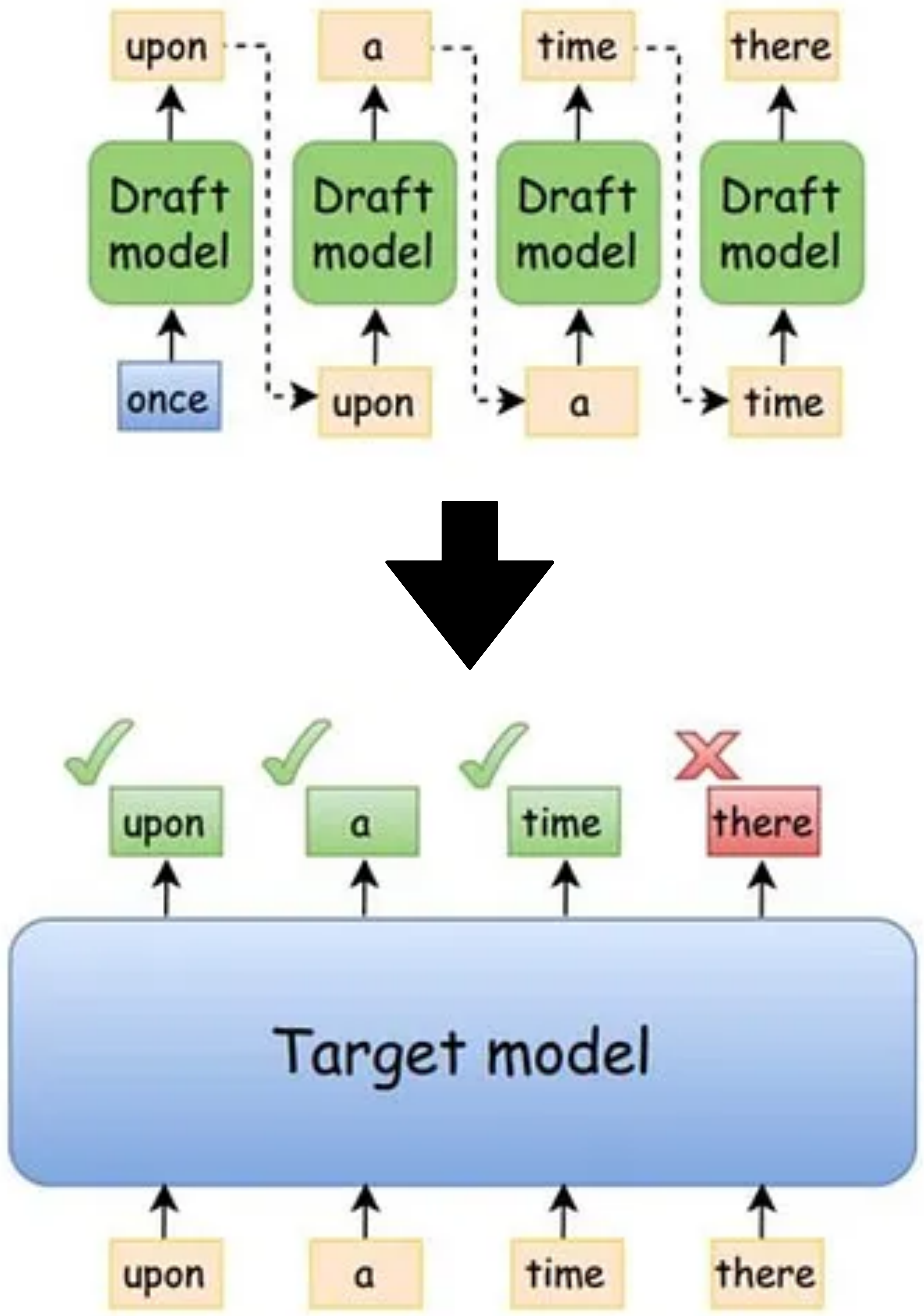
- Step 1: Generate from smaller draft model
  - Typically set window *k* as a hyper parameter of the number of tokens to generated !
- Step 2: Verify generated tokens in parallel using larger target model
  - If generated tokens are "in distribution" of target model, keep the generated tokens.
  - If not, reject draft tokens and decode from target model at first generated token that is not "in distribution"



Leviathan et al., 2022

Speculative Decoding

- What does "in distribution" mean?
  - **Greedy decoding**: same max-probability token
  - **Sampling**: probability of token within some bound of max-probability token of the target model
    - More details: https://arxiv.org/abs/2211.17192
- Considerations
  - Large *k*: many rejections (draft model wasted computations)
  - Small *k*: target model verifies more often (large computation)



Leviathan et al., 2022

Speculative Decoding

Leviathan et al., 2022

https://research.google/blog/looking-back-at-speculative-decoding/

Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning	Y/N	Y/N
Quantization	Yes	Yes
Weight Factorization	Yes	No
Weight Sharing	Yes	No
Knowledge distillation	Yes…	Yes…
Speculative Decoding	No	Yes

Recap

- Compression leads to improving:
  - Number of parameters
  - Inference time
- Different compression techniques
  - Pruning, quantization, factorization, weight sharing, knowledge distillation
- Size-performance trade-off
  - Heavily compressed large models > lightly compressed small models