Model Compression

Mohammadreza Banaei

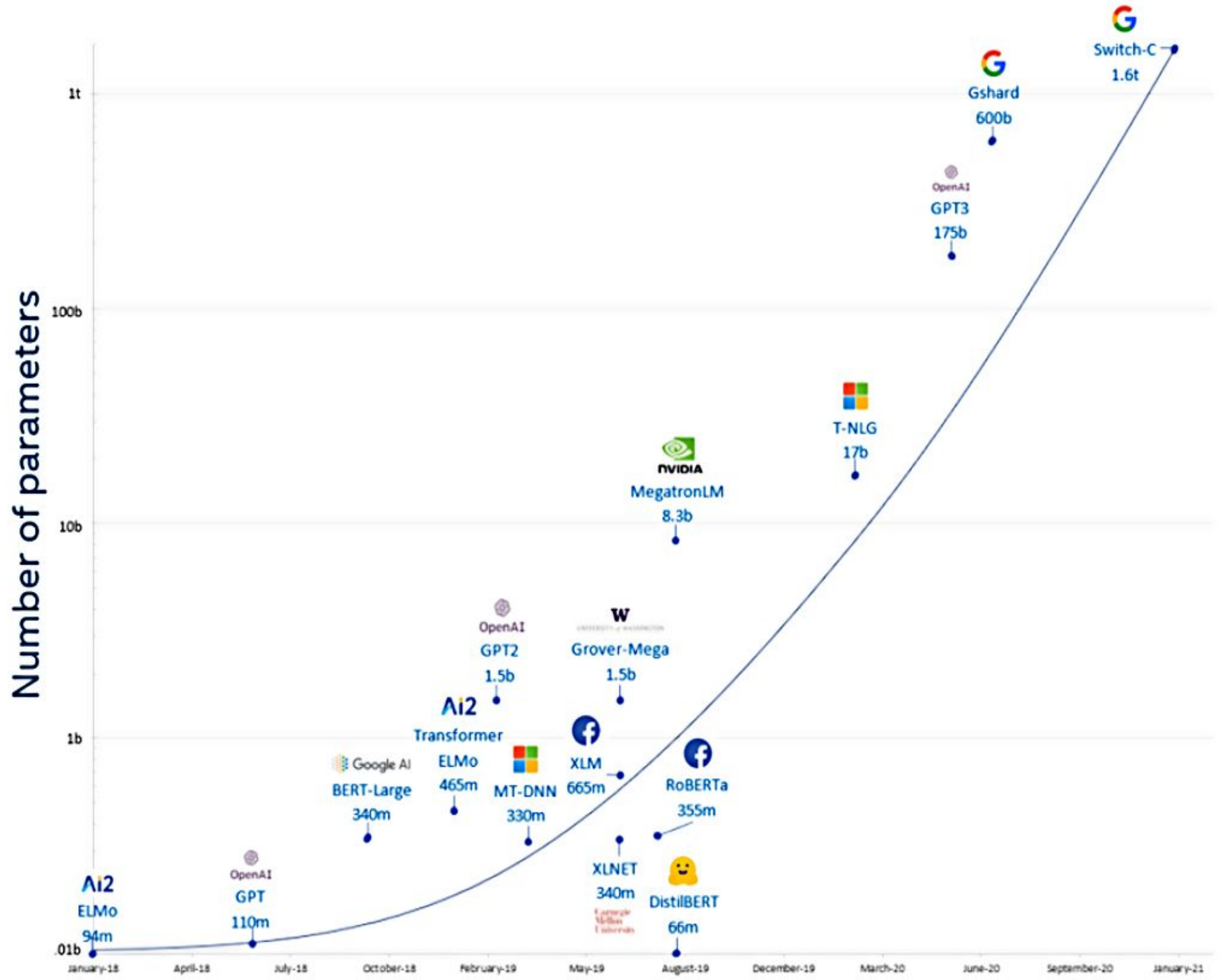
Outline

- Motivation
- Compression methods
  - Pruning
  - Quantization
  - Weight factorization
  - Knowledge distillation
  - Weight Sharing
- Sub-quadratic Transformers

Why do we need compression?

Growth of model parameters

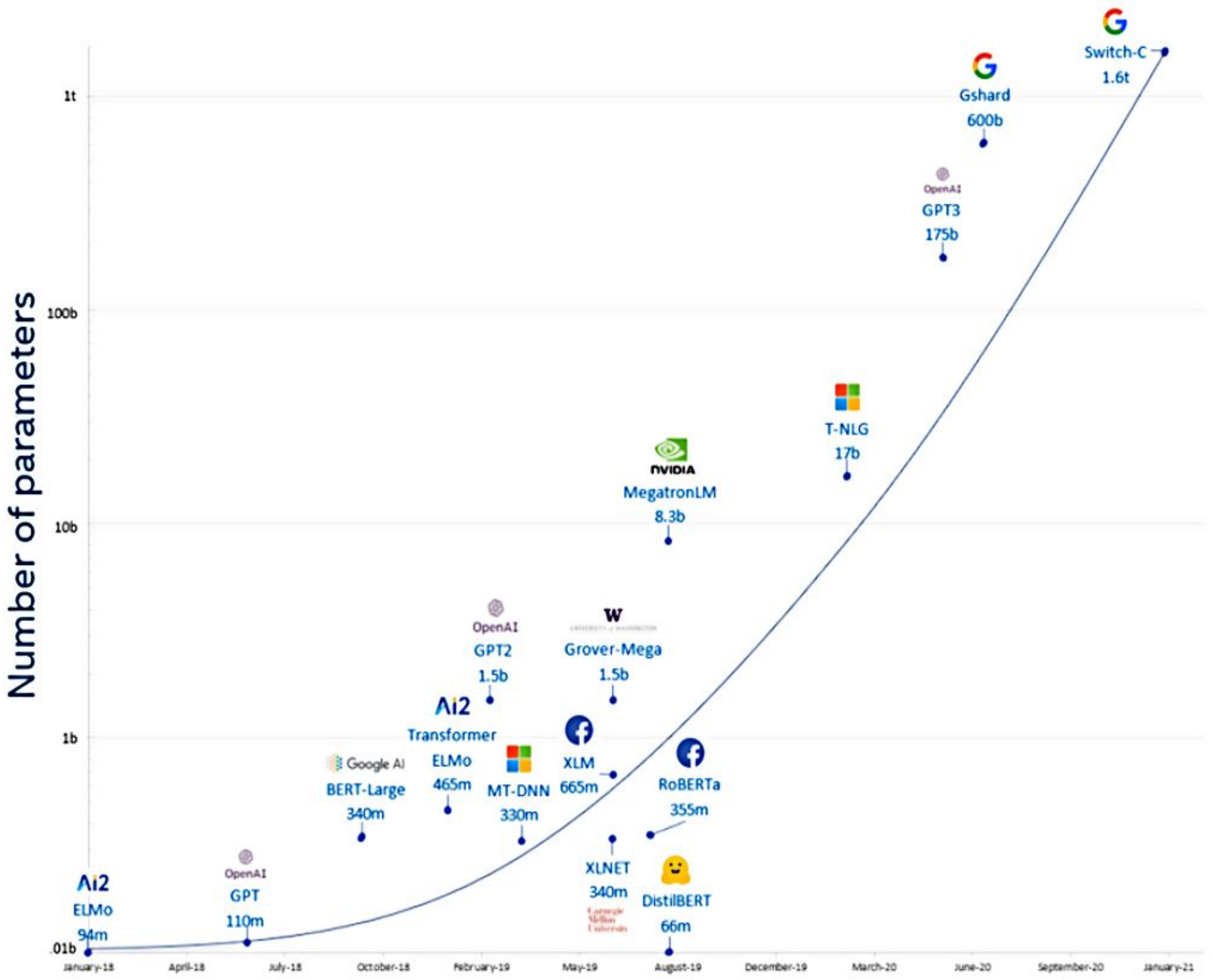
- Exponential growth in model parameters



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

Growth of model parameters

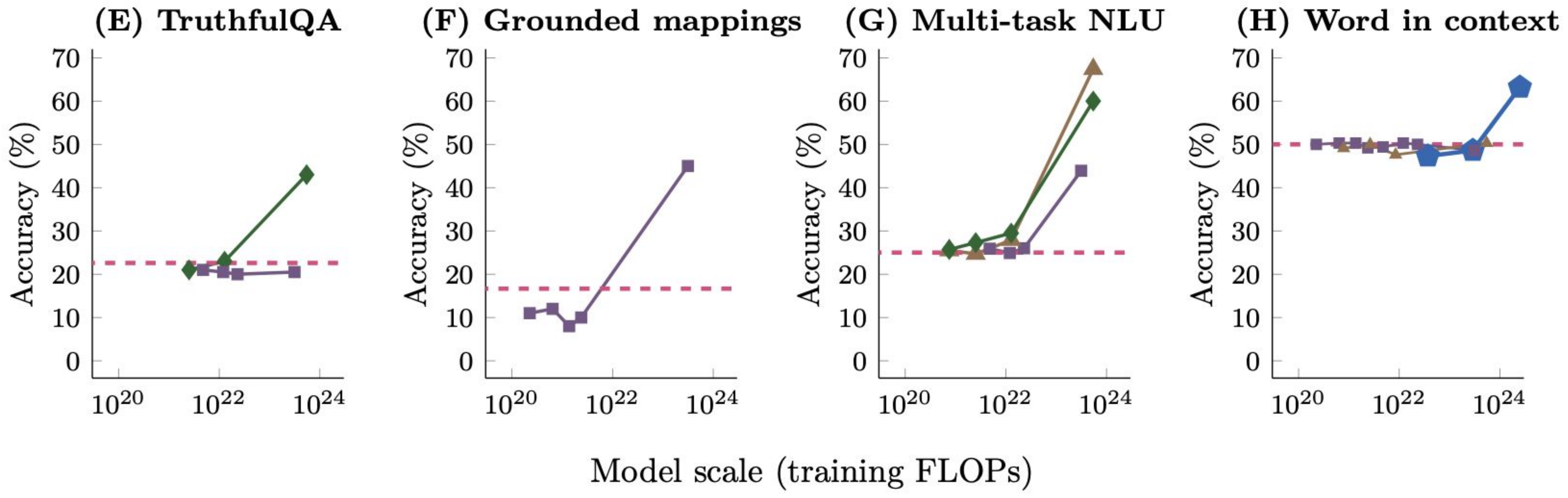
- Exponential growth in model parameters
  - Scaling laws

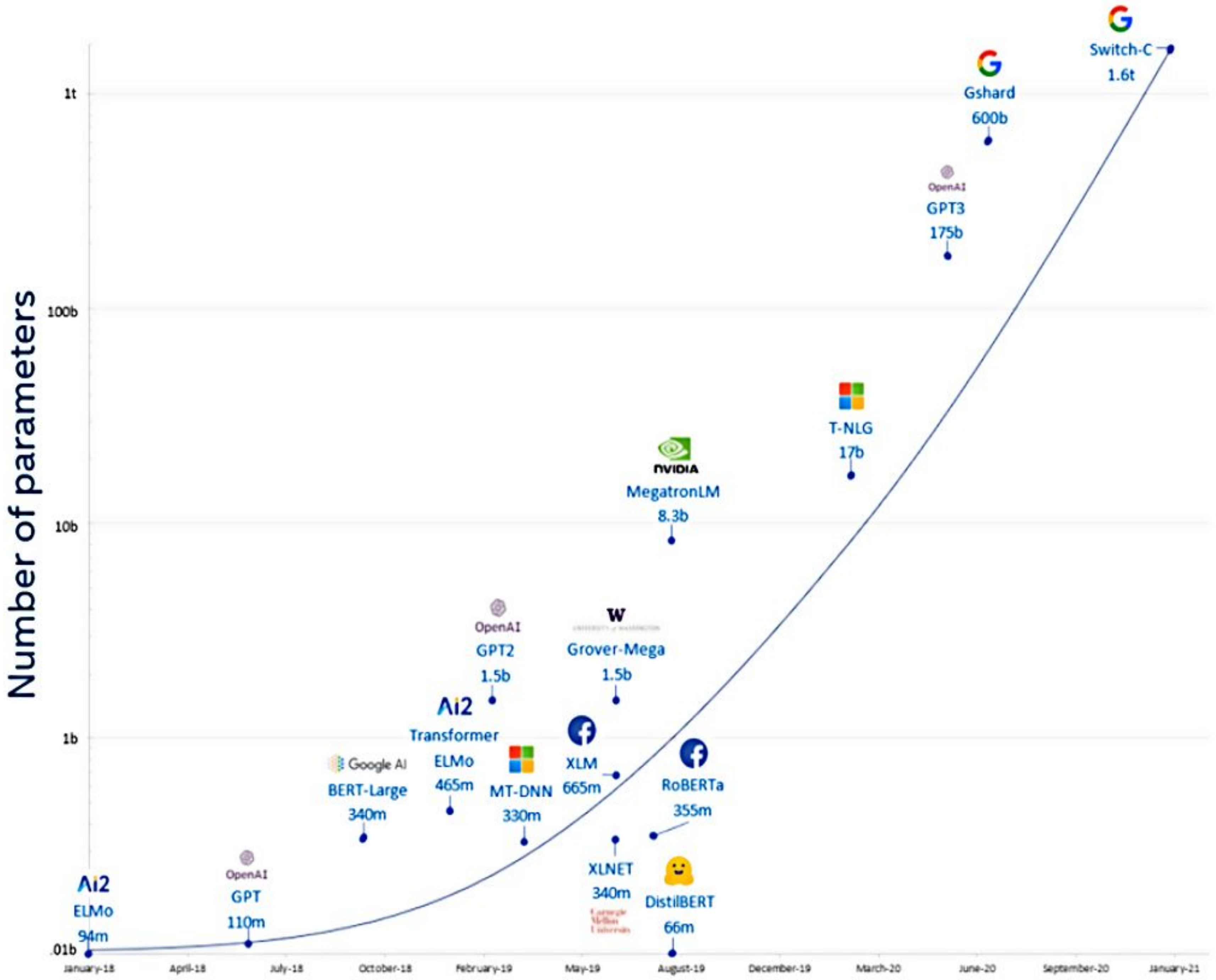


Kaplan, Jared, et al. "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361* (2020)
Wei, Jason, et al. "Emergent abilities of large language models." *arXiv preprint arXiv:2206.07682* (2022)

Growth of model parameters

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

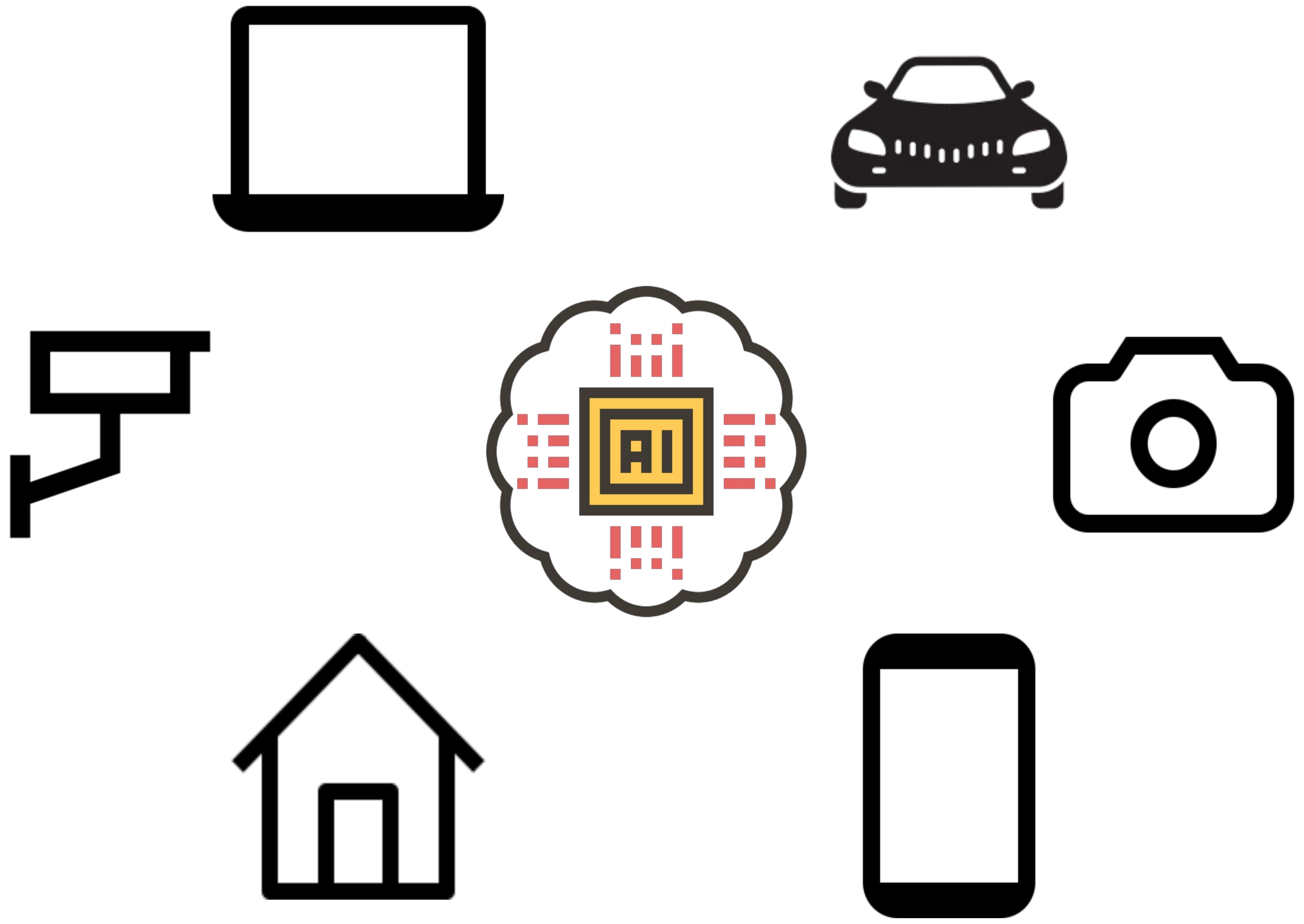




Kaplan, Jared, et al. "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361* (2020)
Wei, Jason, et al. "Emergent abilities of large language models." *arXiv preprint arXiv:2206.07682* (2022)

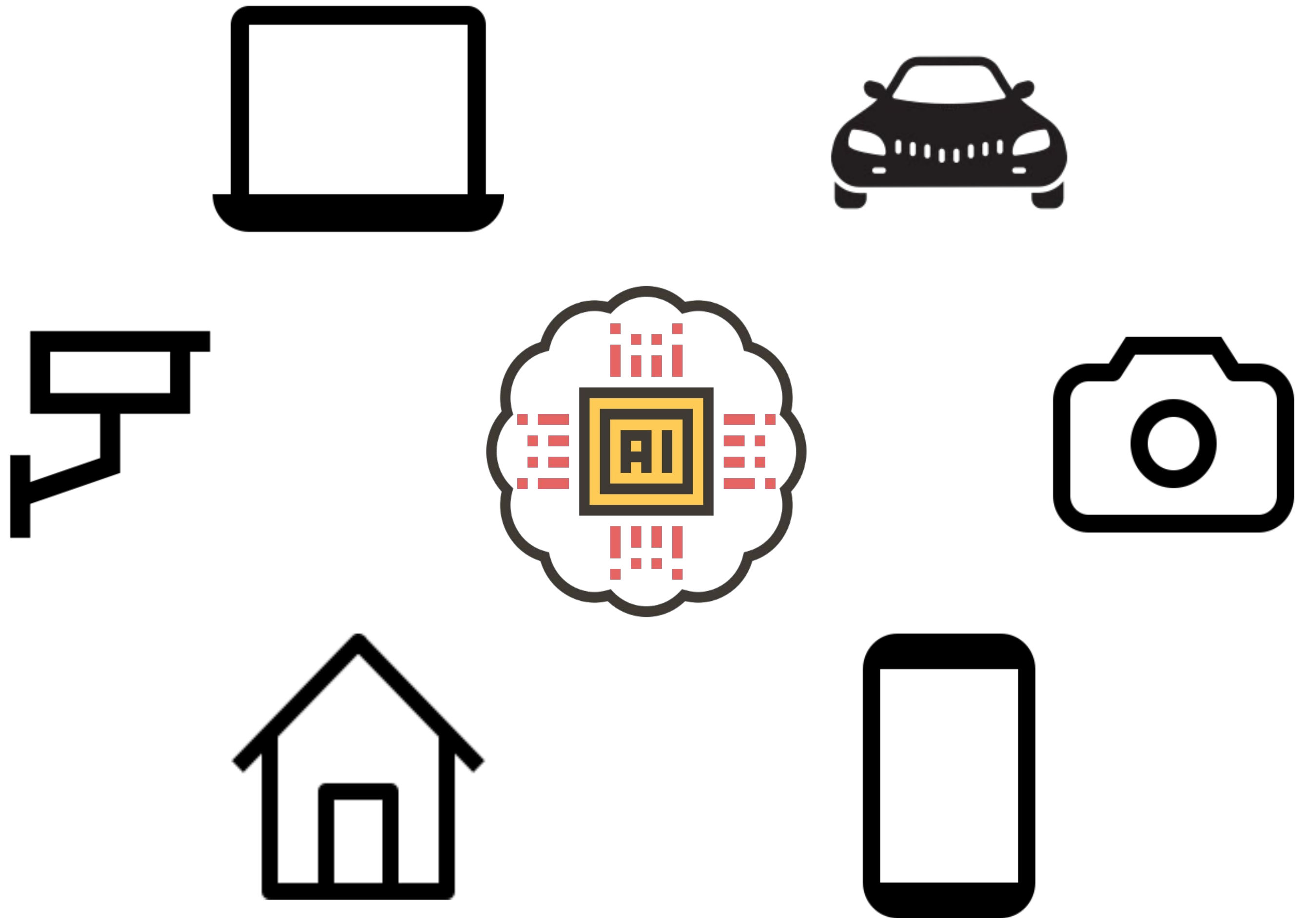
LLM Deployment in Production

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



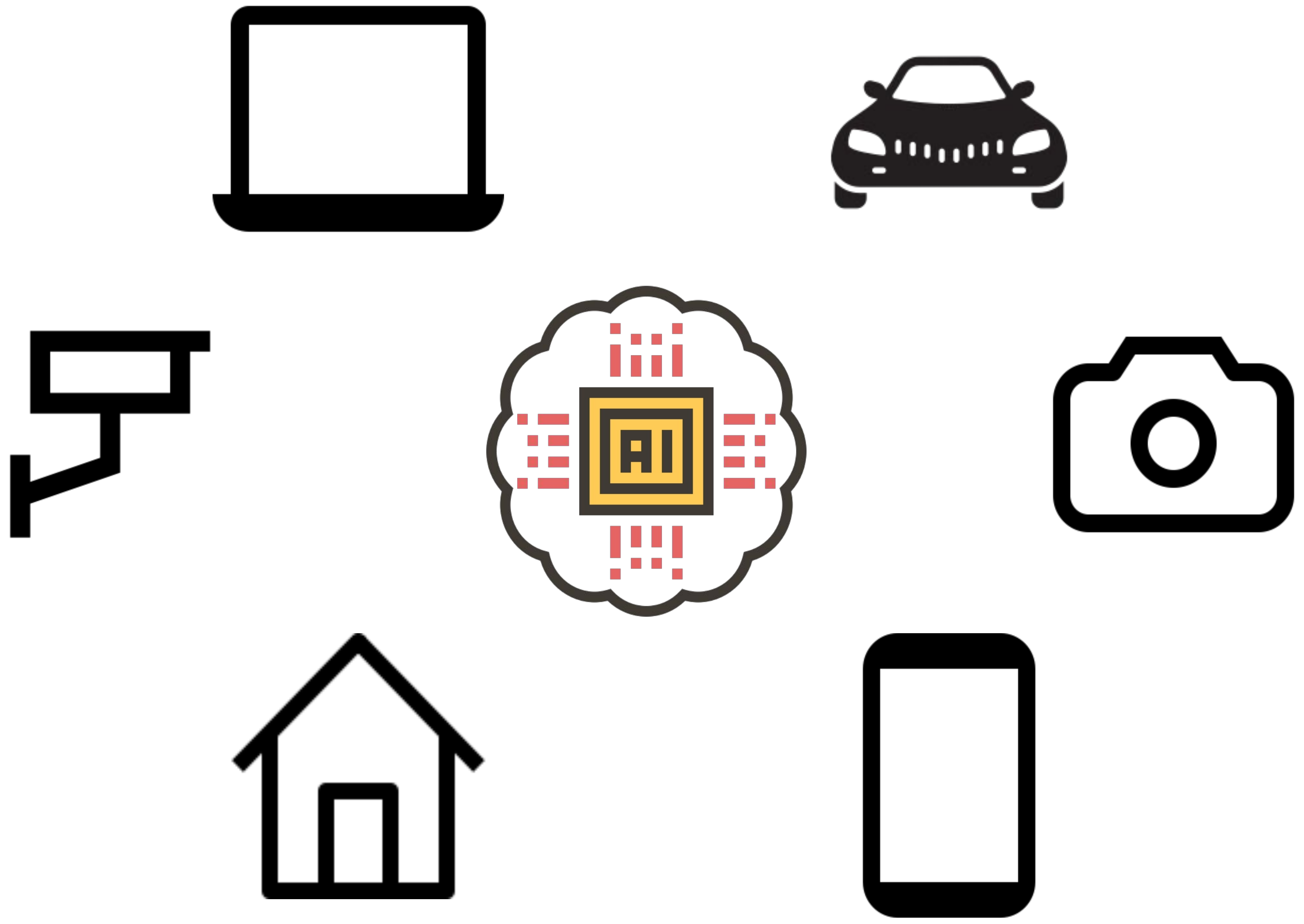
LLM Deployment in Production

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



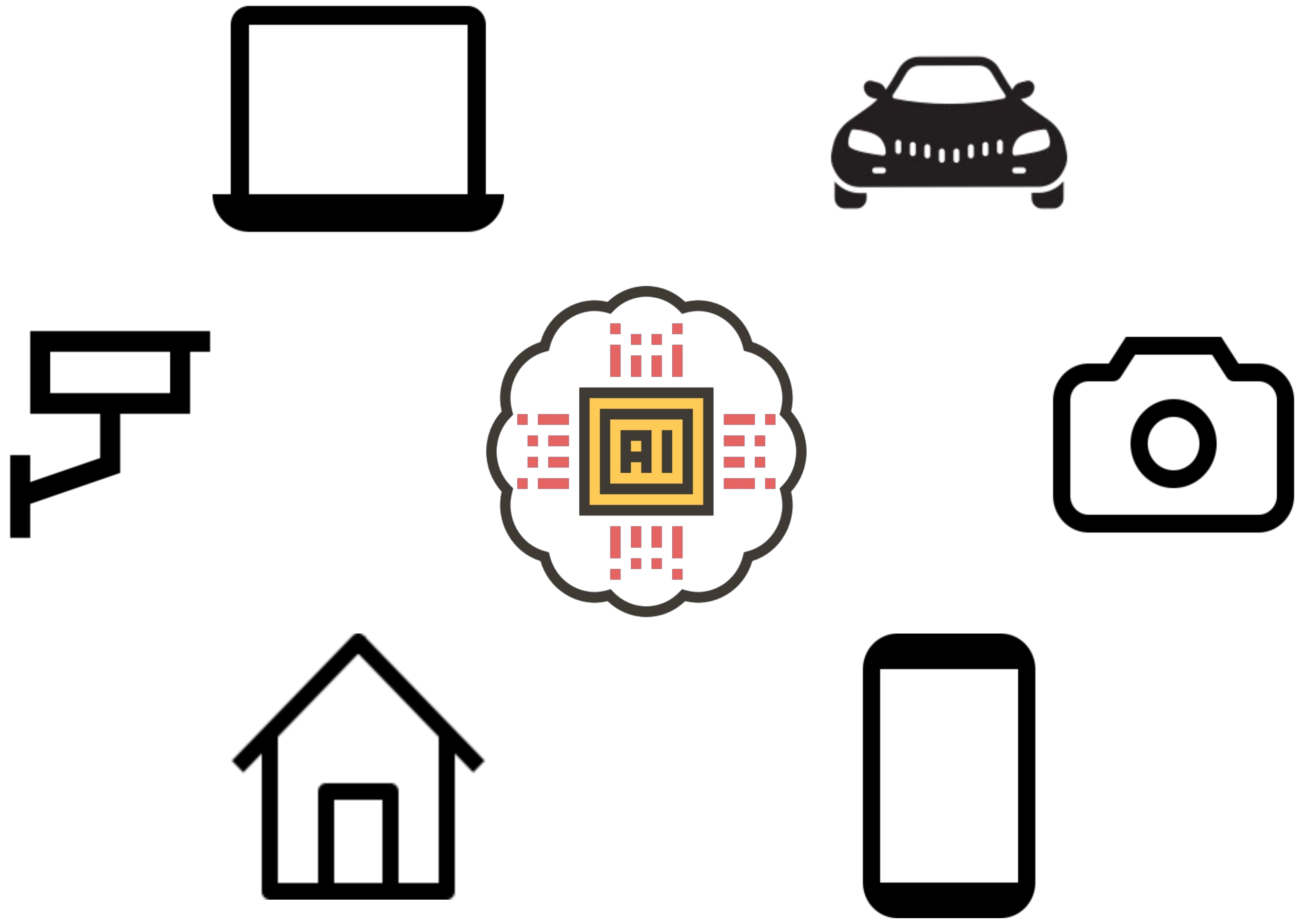
LLM Deployment in Production

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



LLM Deployment in Production

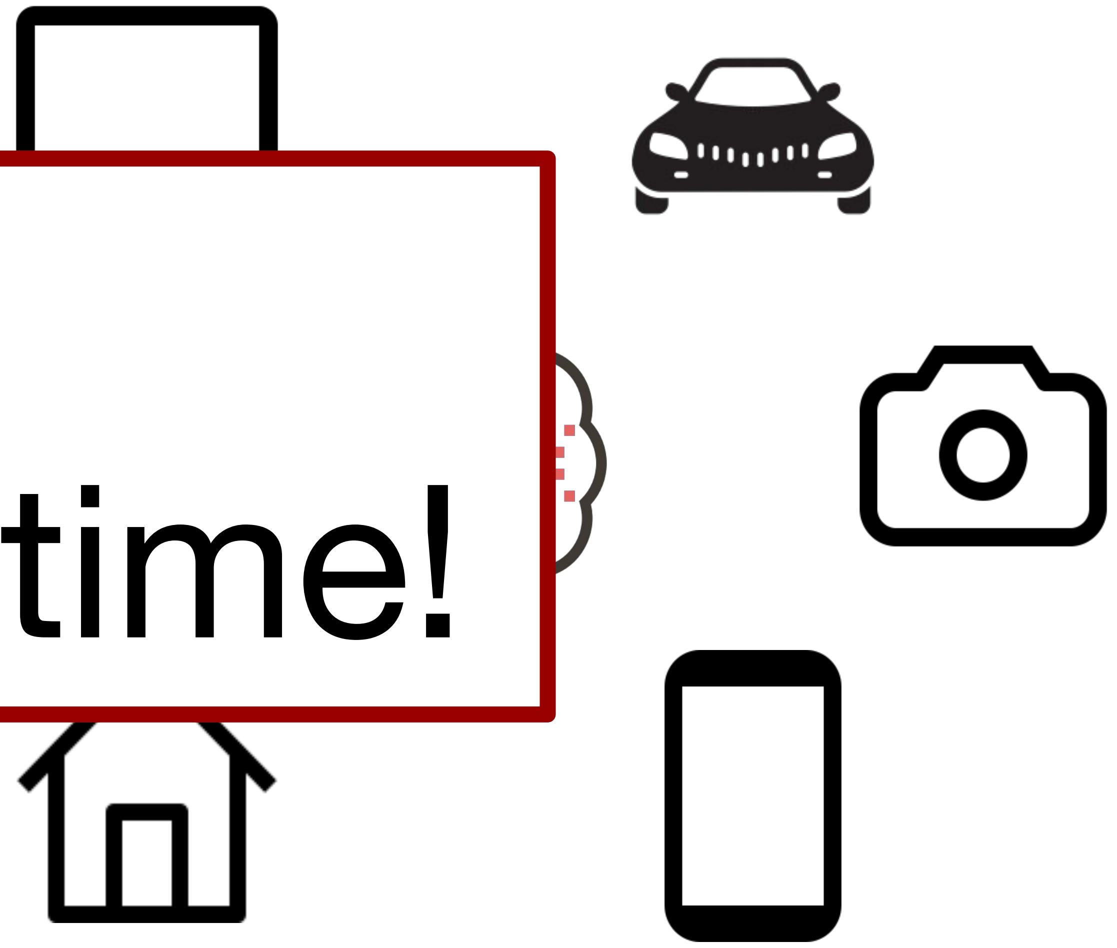
- Cloud processing not always possible
  - Latency issue
  - Data privacy
- Inference time for edge devices
- Memory issue
  - ~350 GB just for storing a LLM weights!
- Finetuning LLMs
  - Time-consuming
  - Expensive



LLM Deployment in Production

- Cloud processing not always possible
  - Latency issue
  - Data privacy
- Inference
- Memory
  - ~350 GB
- Finetuning LLMs
  - Time-consuming
  - Expensive

Compression could reduce #parameters and inference time!



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)

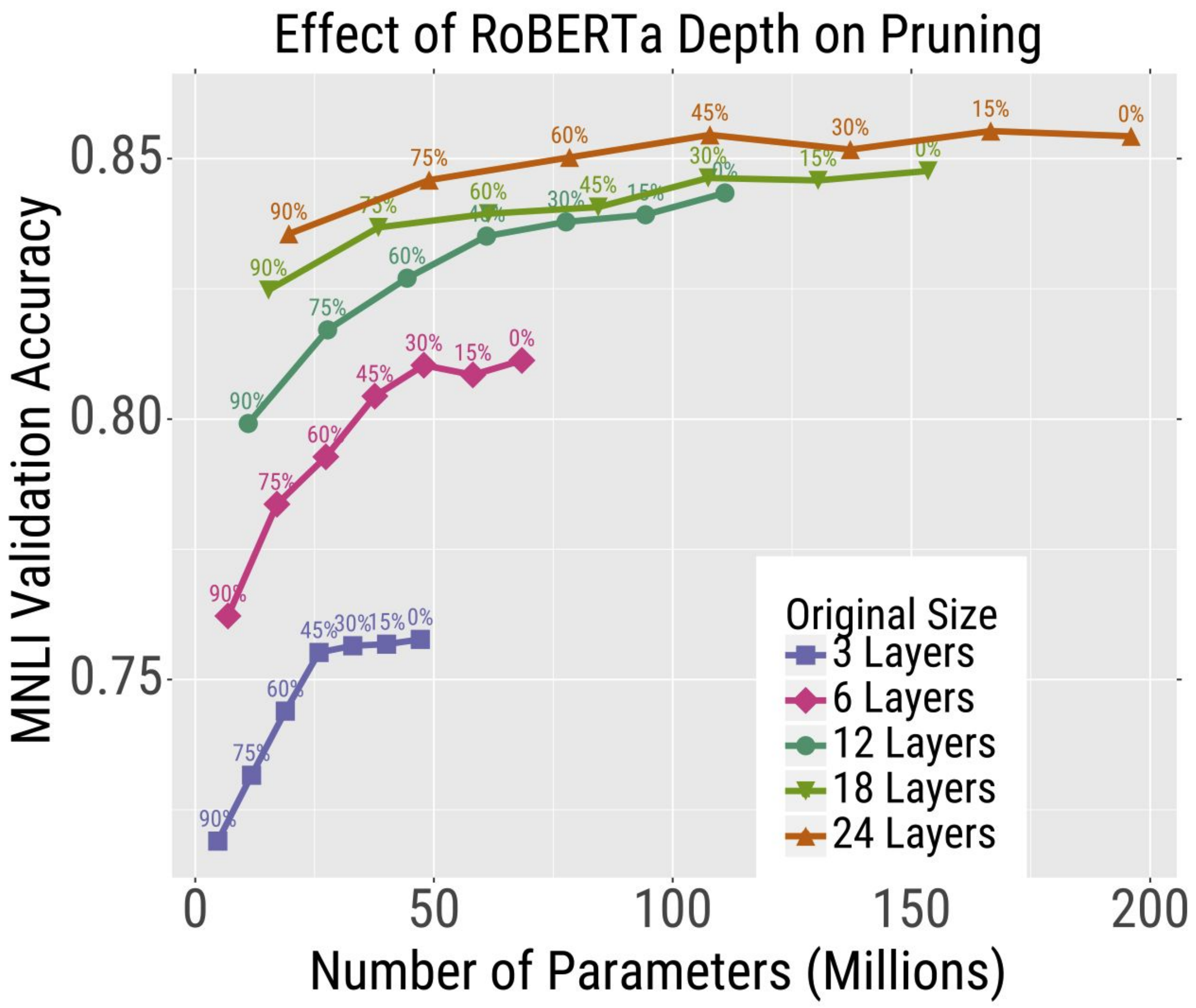
Train Large, then Compress!

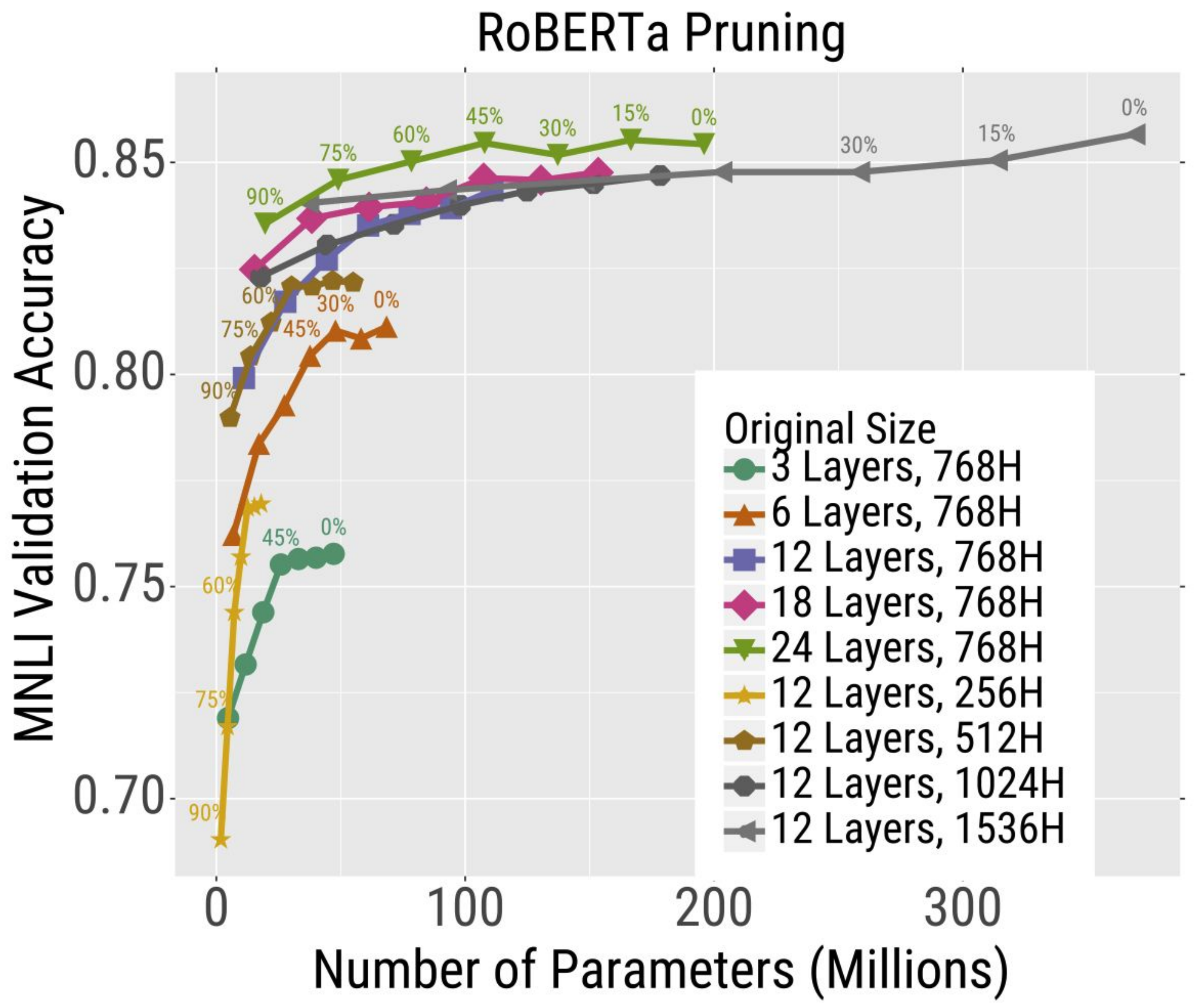
- Large models are more robust to compression techniques than small models
- For given test-time constraints (e.g., inference time, #parameter)
  - heavily compressed, large models > lightly compressed, small models

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 > lightly compressed, small models
- Comparing downstream task performance for discussed scenarios



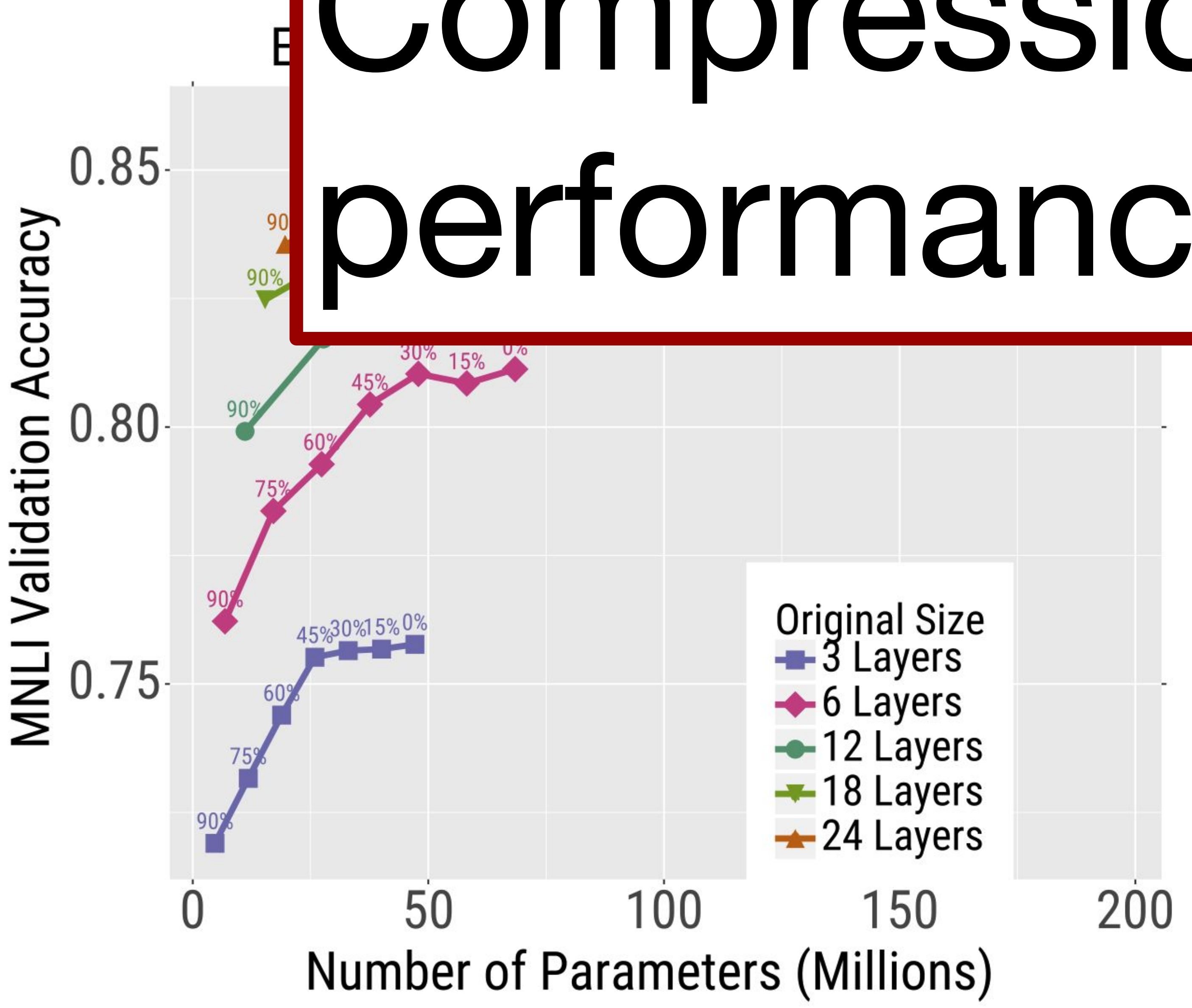


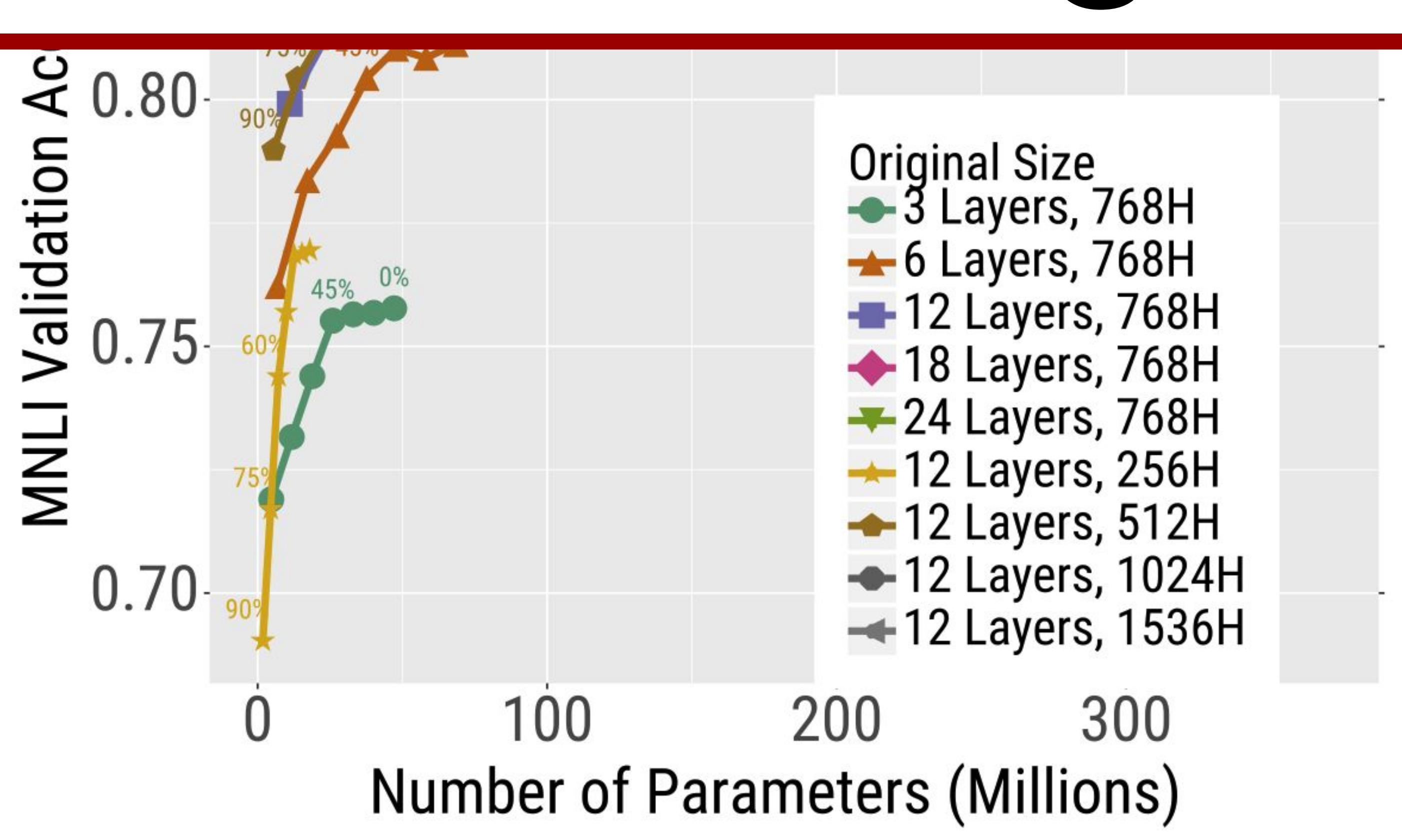
Li, Zhuohan et al. "Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers." *ArXiv* abs/2002.11794 (2020)

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 > lightly compressed, 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 the compression done?

Compression Methods

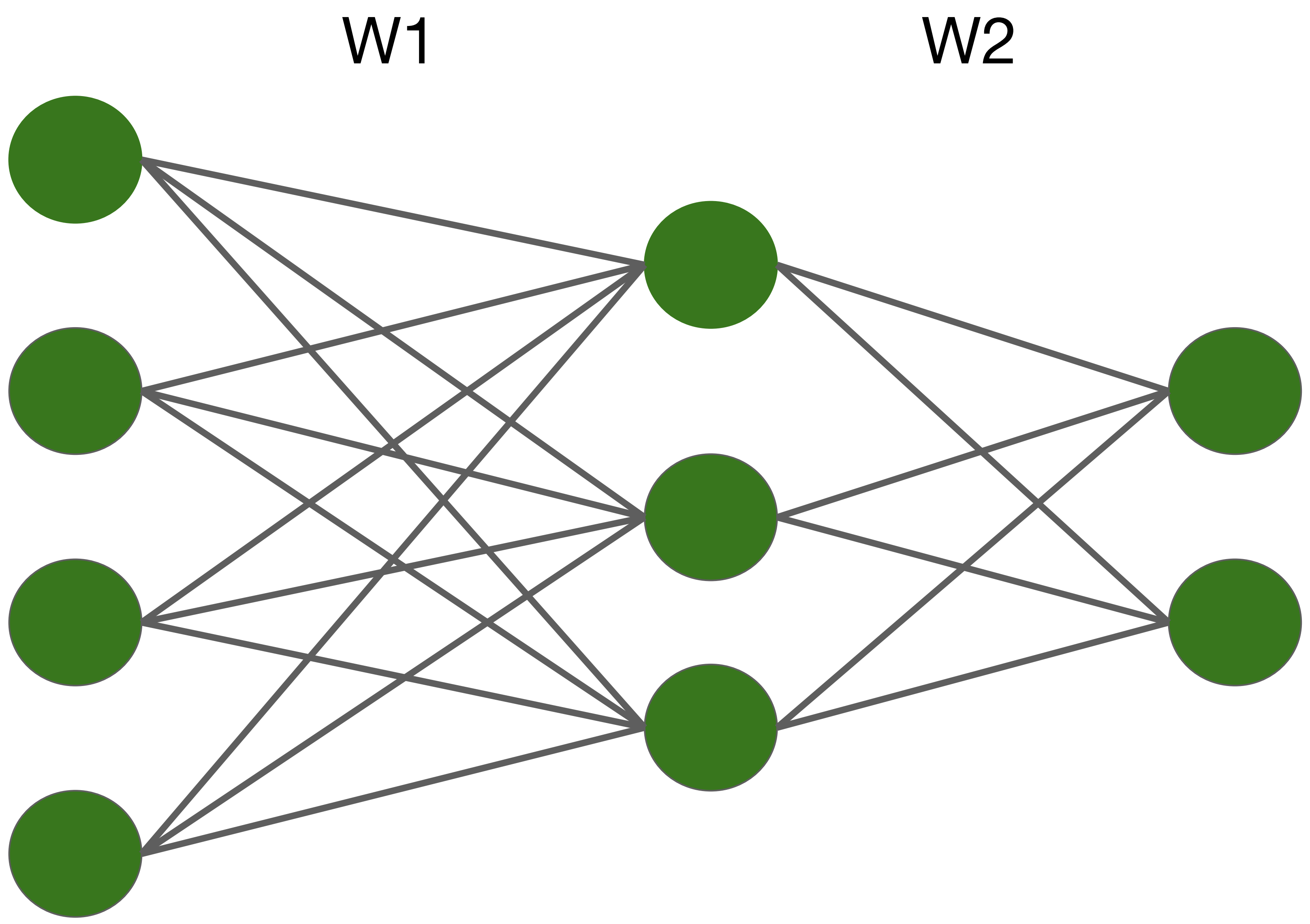
- Pruning
- Quantization
- Weight factorization
- Knowledge Distillation
- Weight sharing

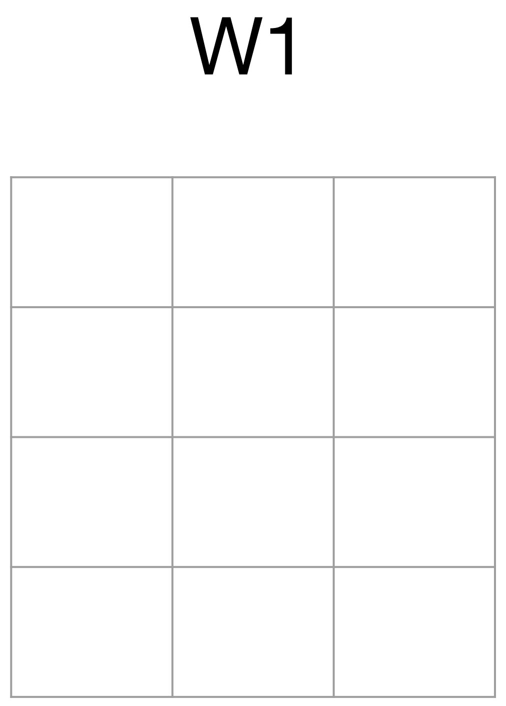
Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning		
Quantization		
Weight Factorization		
Weight Sharing		
Knowledge distillation		
Sub-quadratic Transformer		

Pruning

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

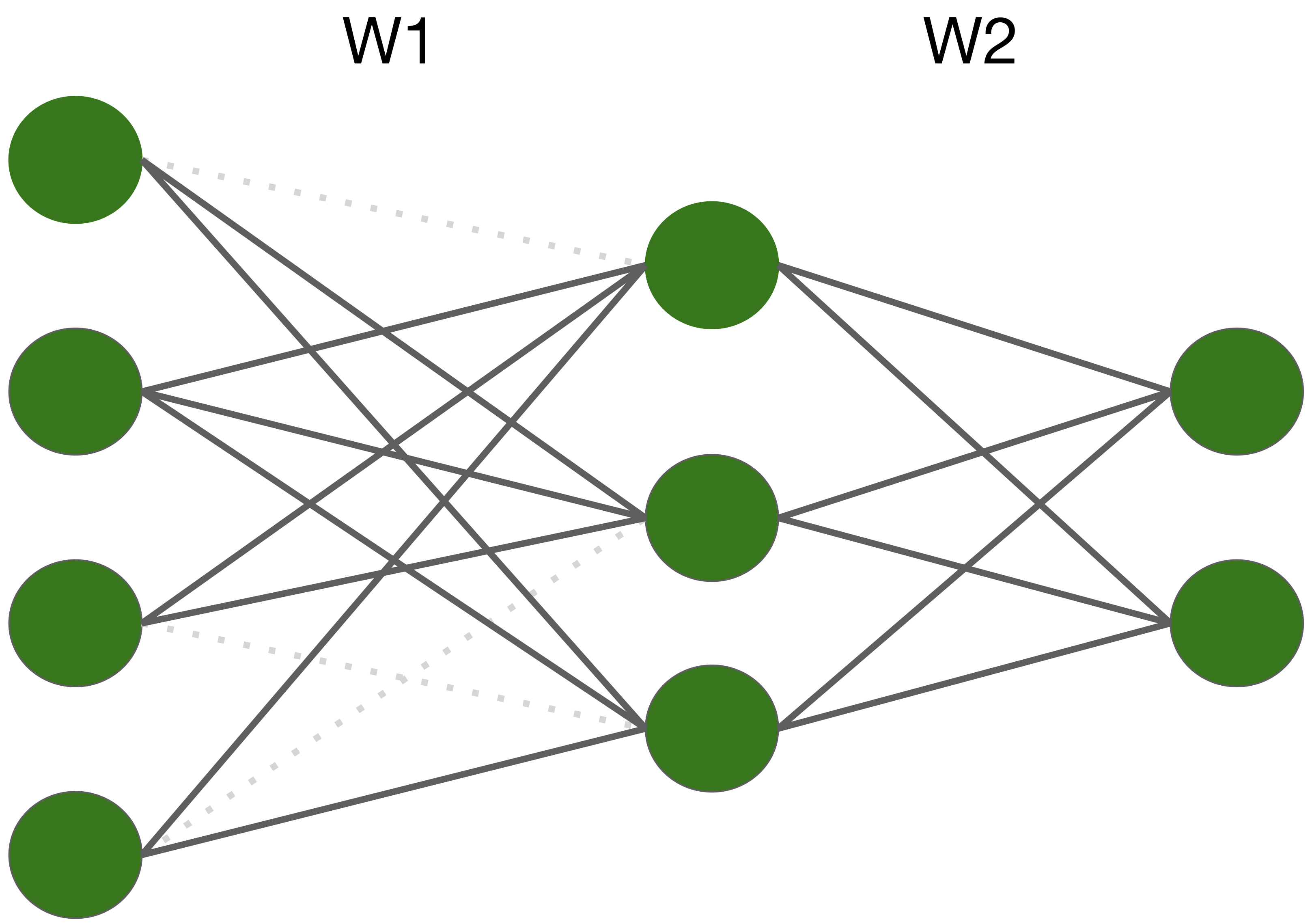




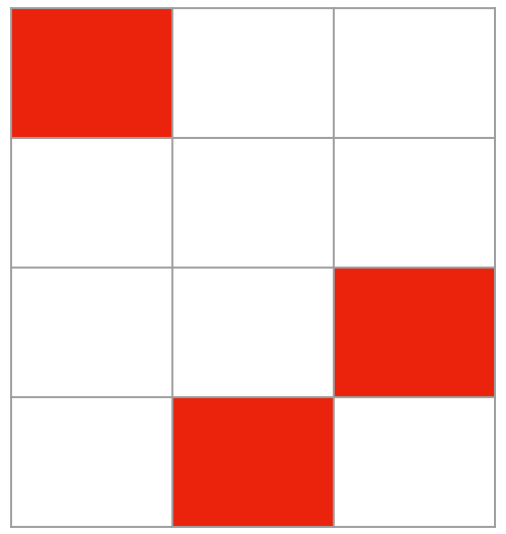
(No pruning)

Pruning

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



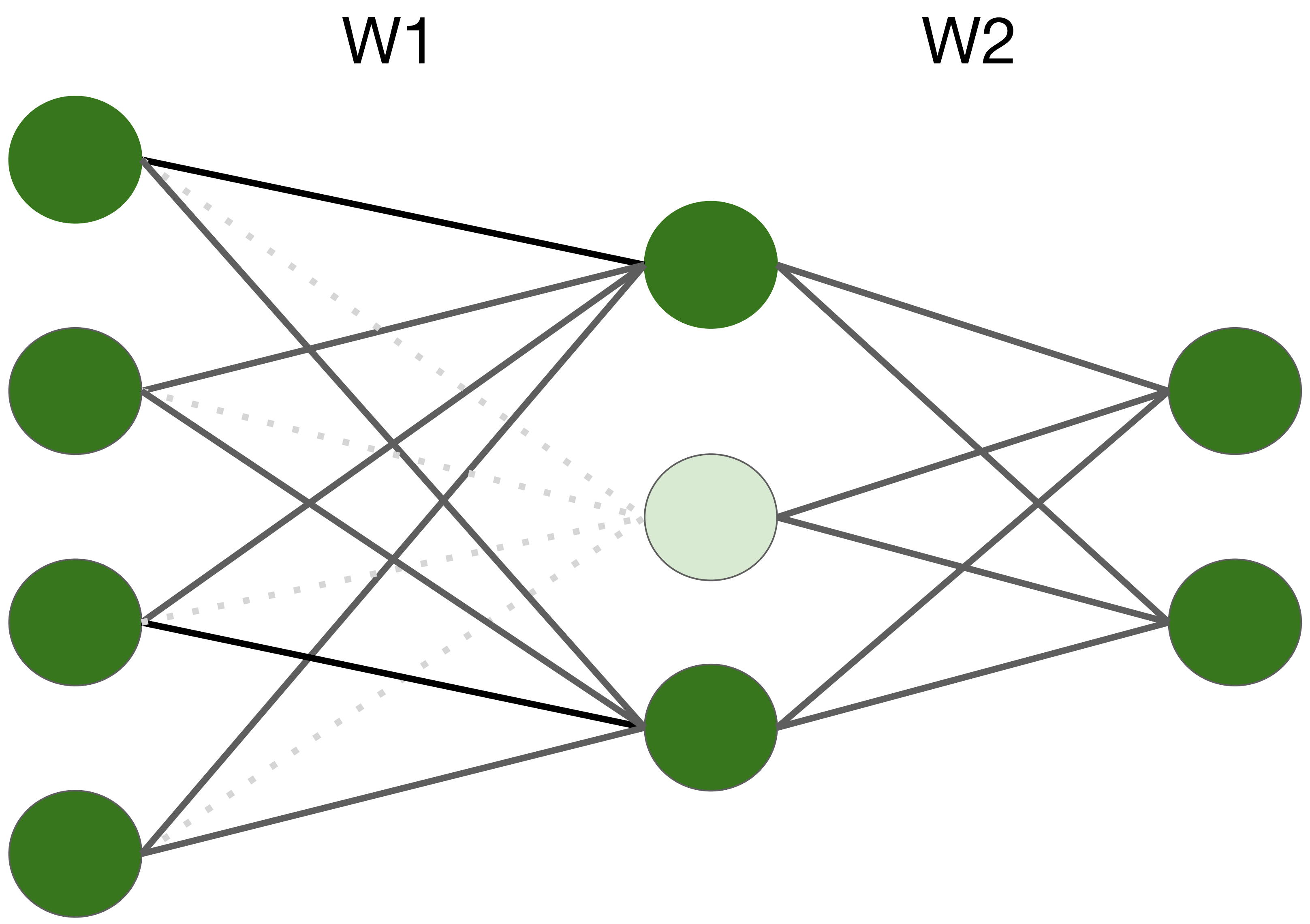
W1

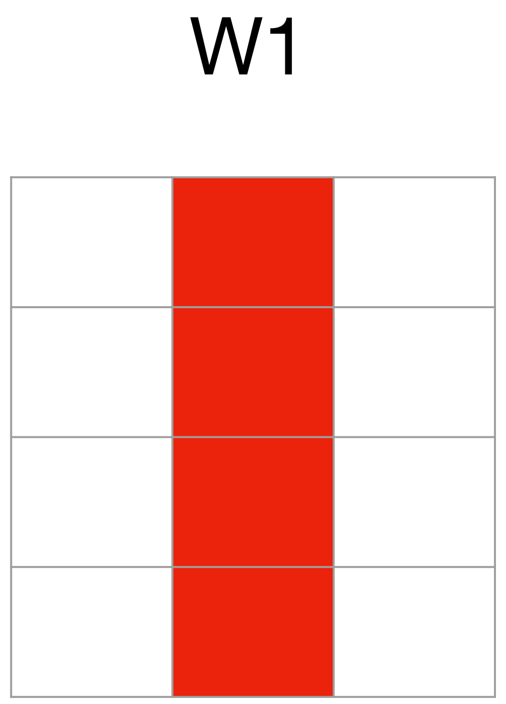


(Unstructured pruning)

Pruning

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

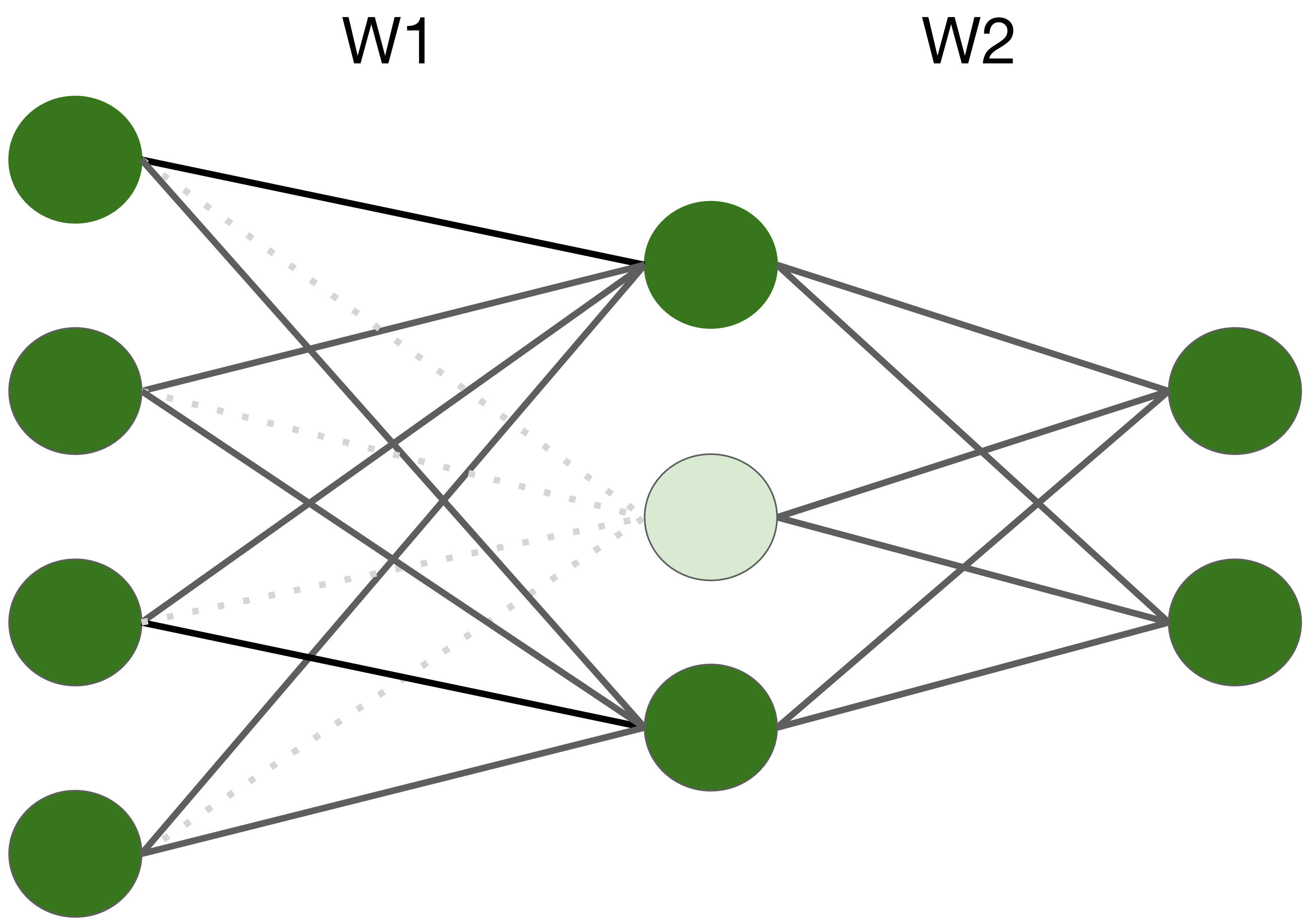


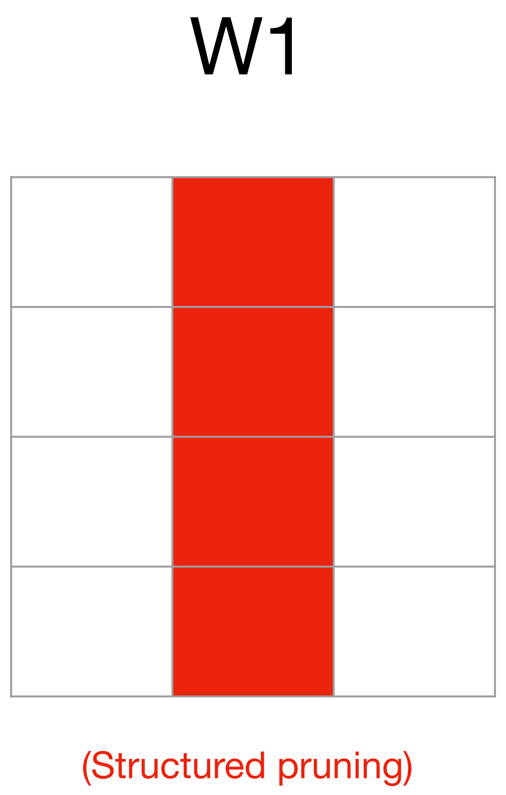


(Structured pruning)

Pruning

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



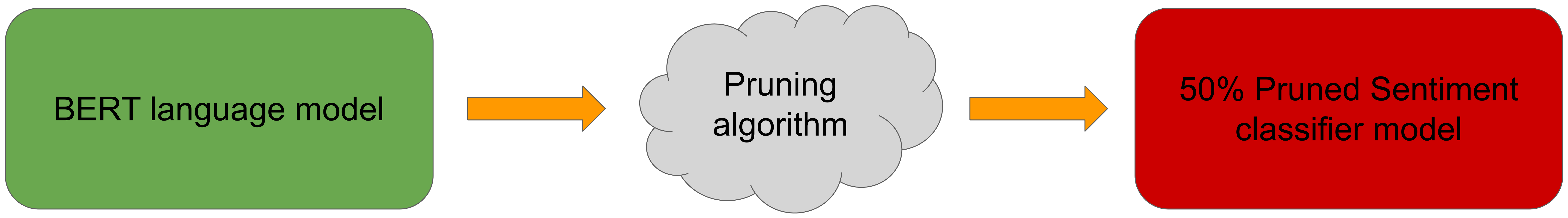


What is the potential benefit of structured pruning?

How to choose pruned weights?

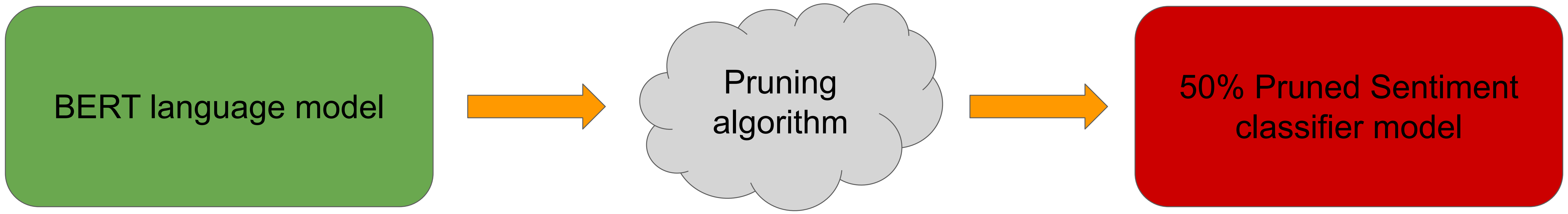
Pruning: case study

- 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

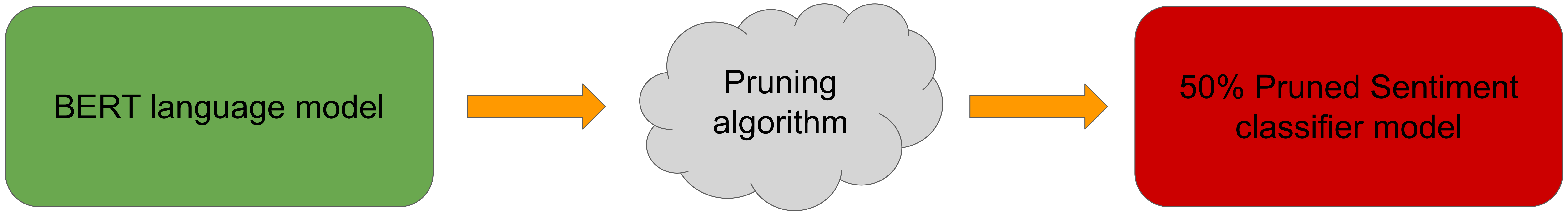


- which weights should be pruned?

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

Pruning: case study

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



- which weights should be pruned?

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

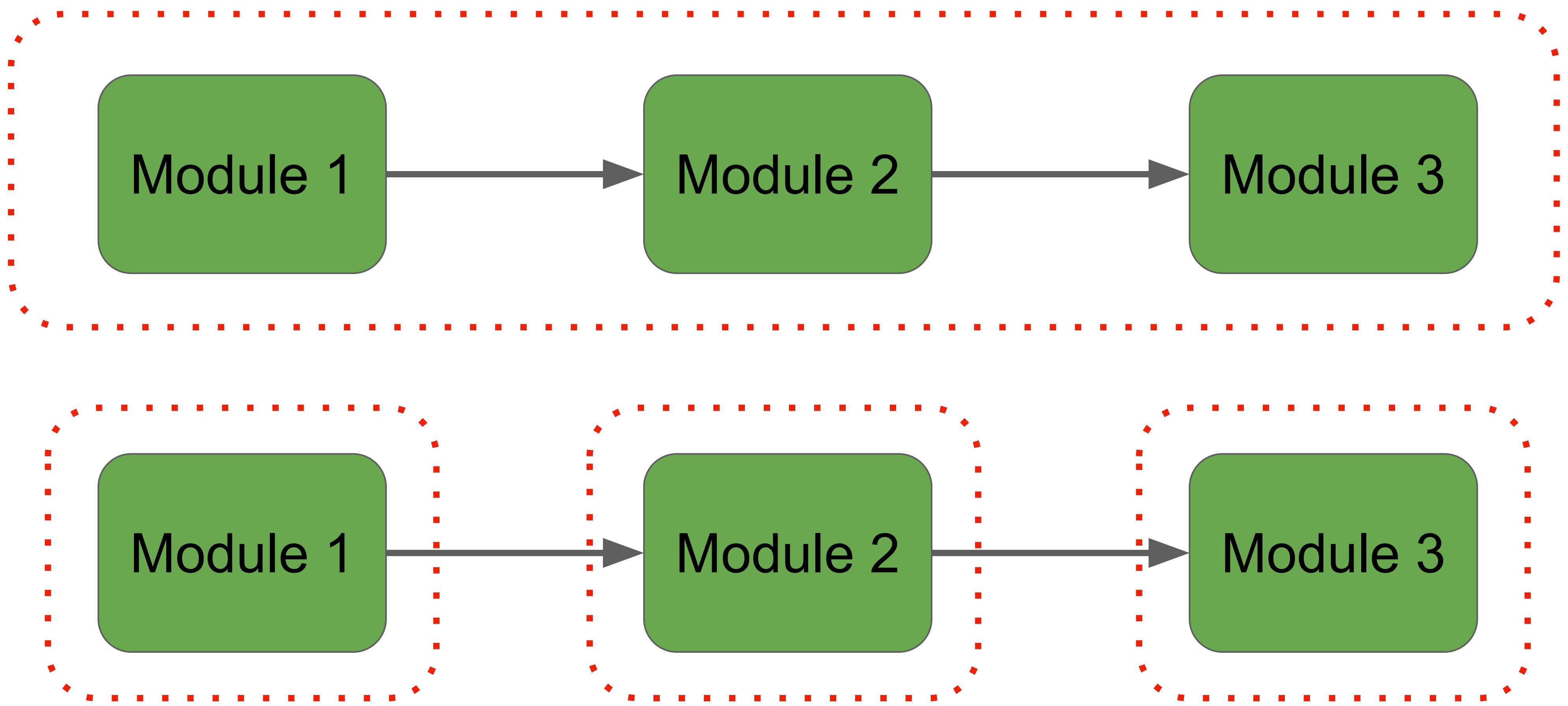
Weight Pruning Methods

- 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

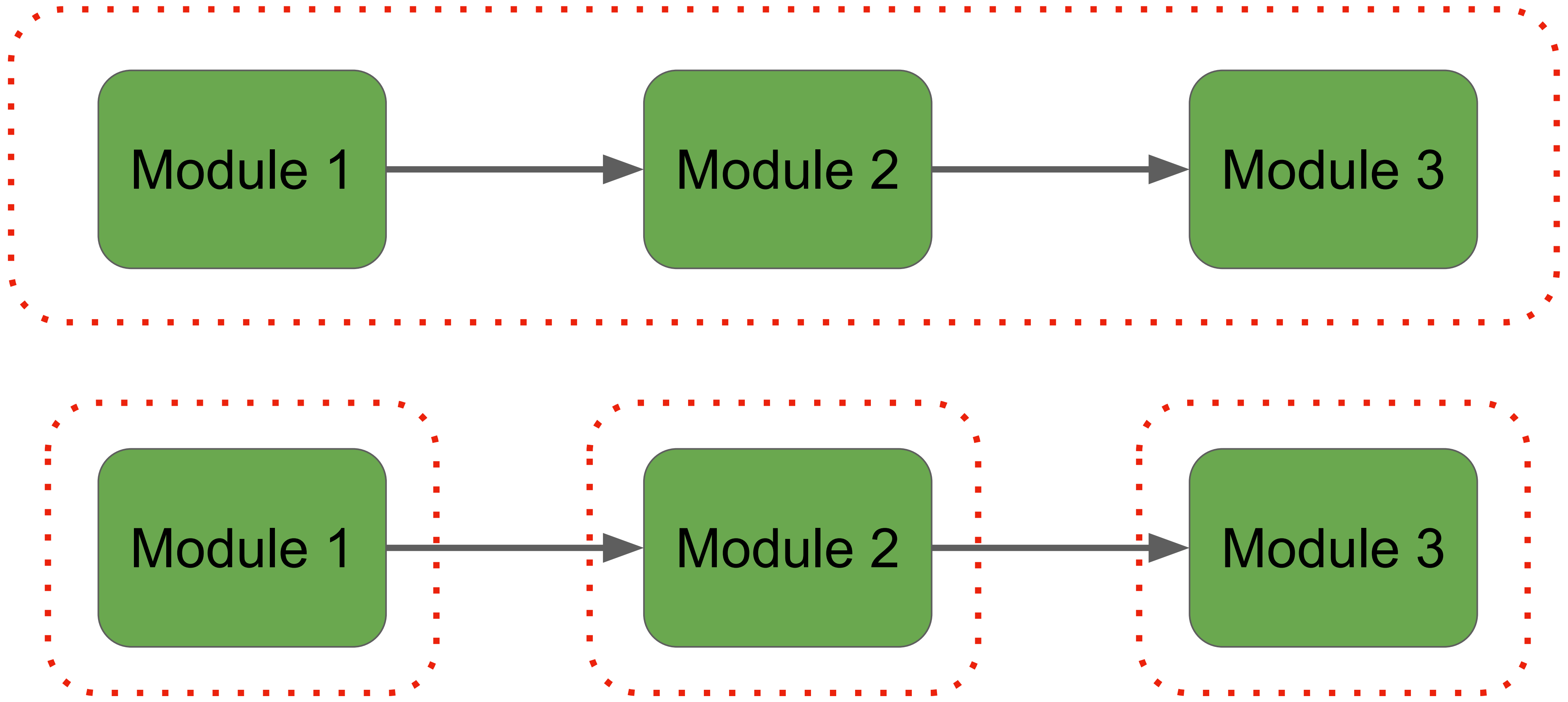
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level



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

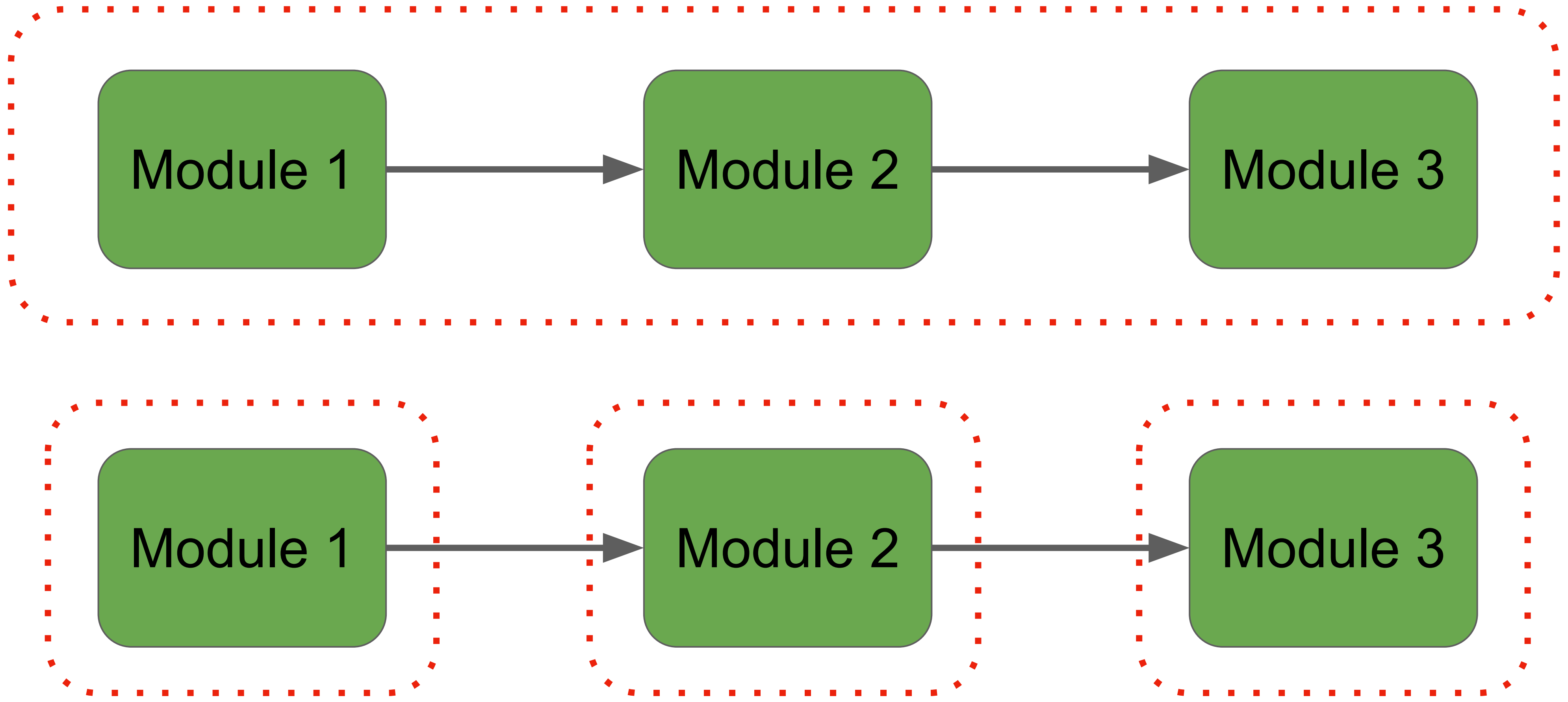
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training



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

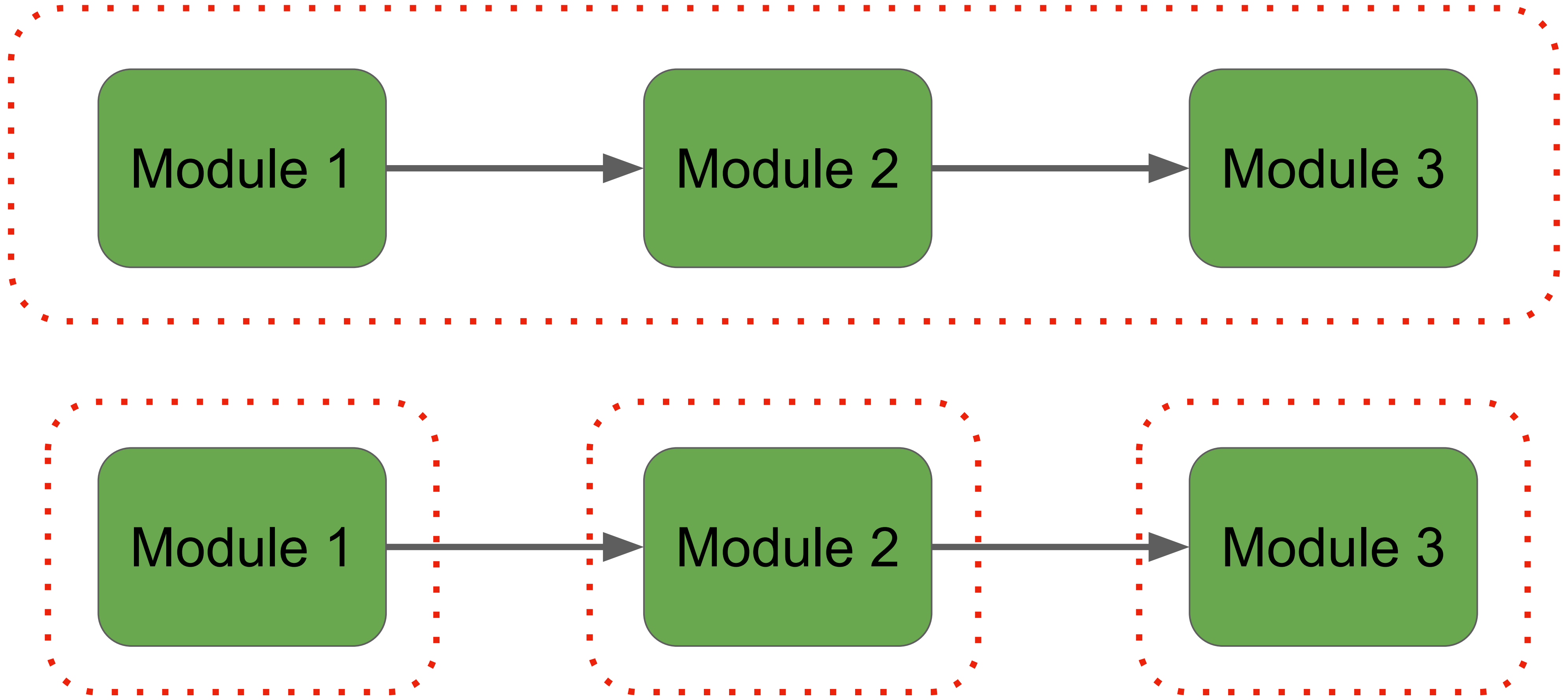
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training
- Movement pruning



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

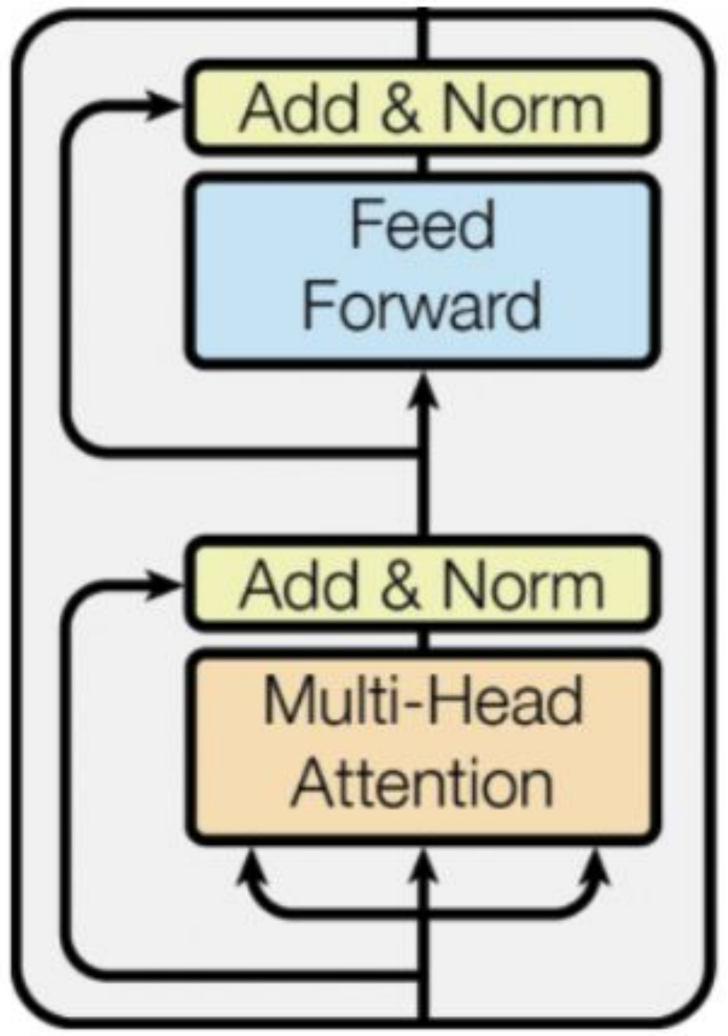
- Magnitude pruning
  - Pruning weights with small magnitude
  - Pruning x% at global Vs. Module level
- Iterative magnitude pruning
  - pruning gradually during training
- Movement pruning
- (Differentiable) masking as a pruning method
  - Example: attention head masking



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).

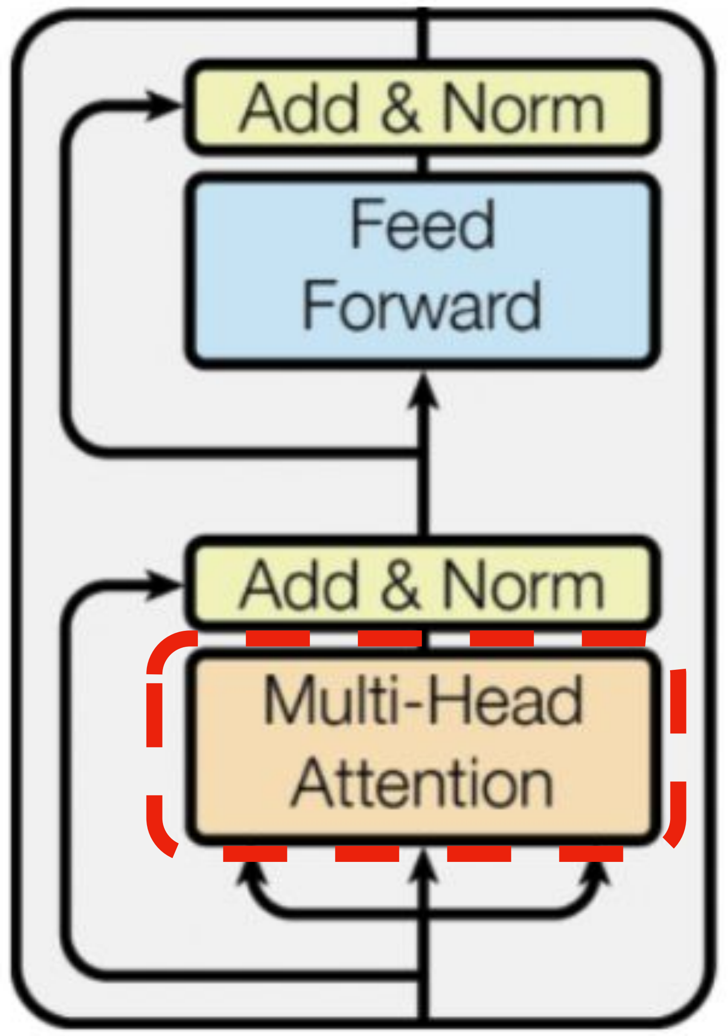
Structured Pruning

- Structured pruning for Transformer language models
  - Pruning neurons



Structured Pruning

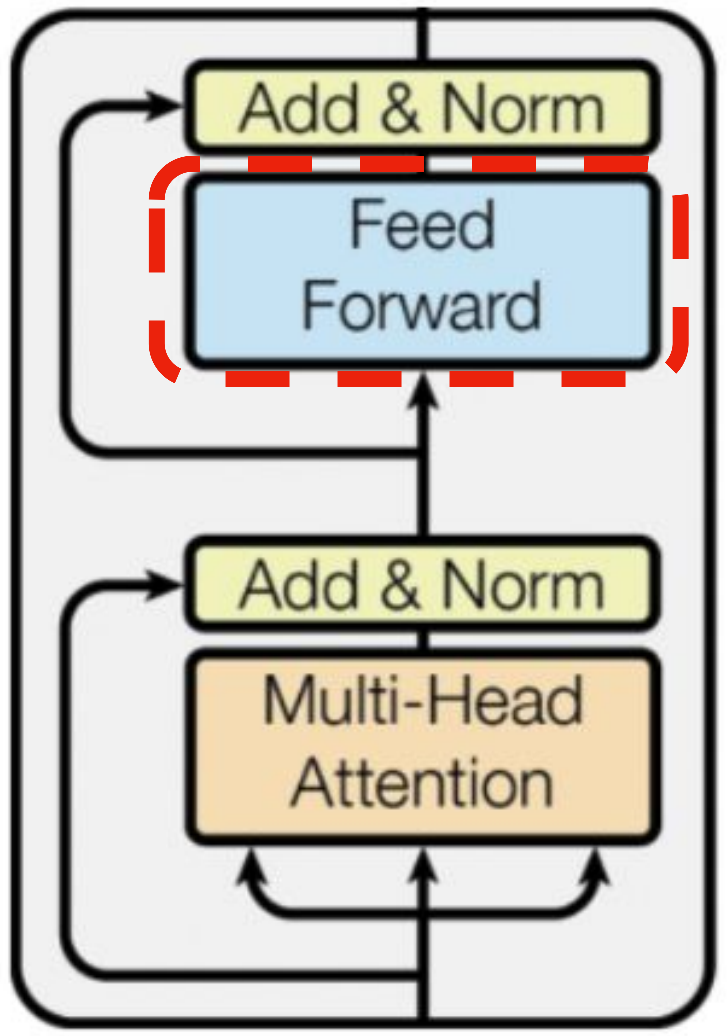
- Structured pruning for Transformer language models
  - Pruning neurons
  - Pruning attention heads



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).

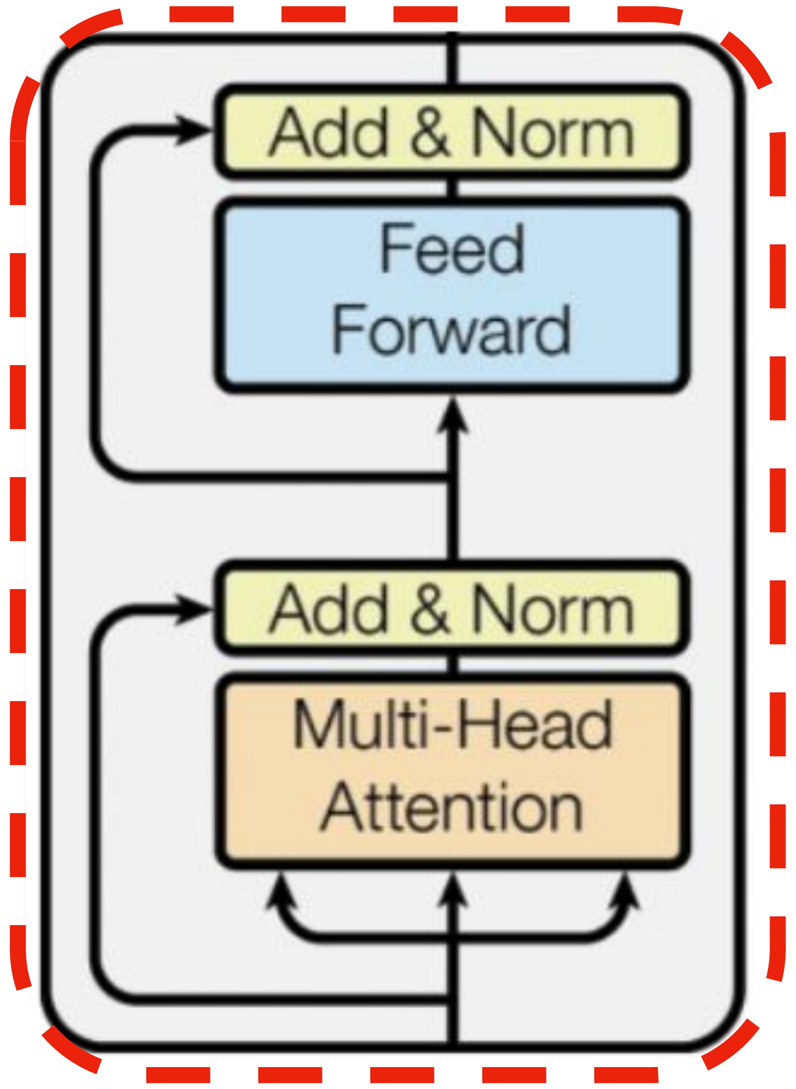
Structured Pruning

- 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?

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

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?

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

$$\downarrow$$

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

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?

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

$$\downarrow$$

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

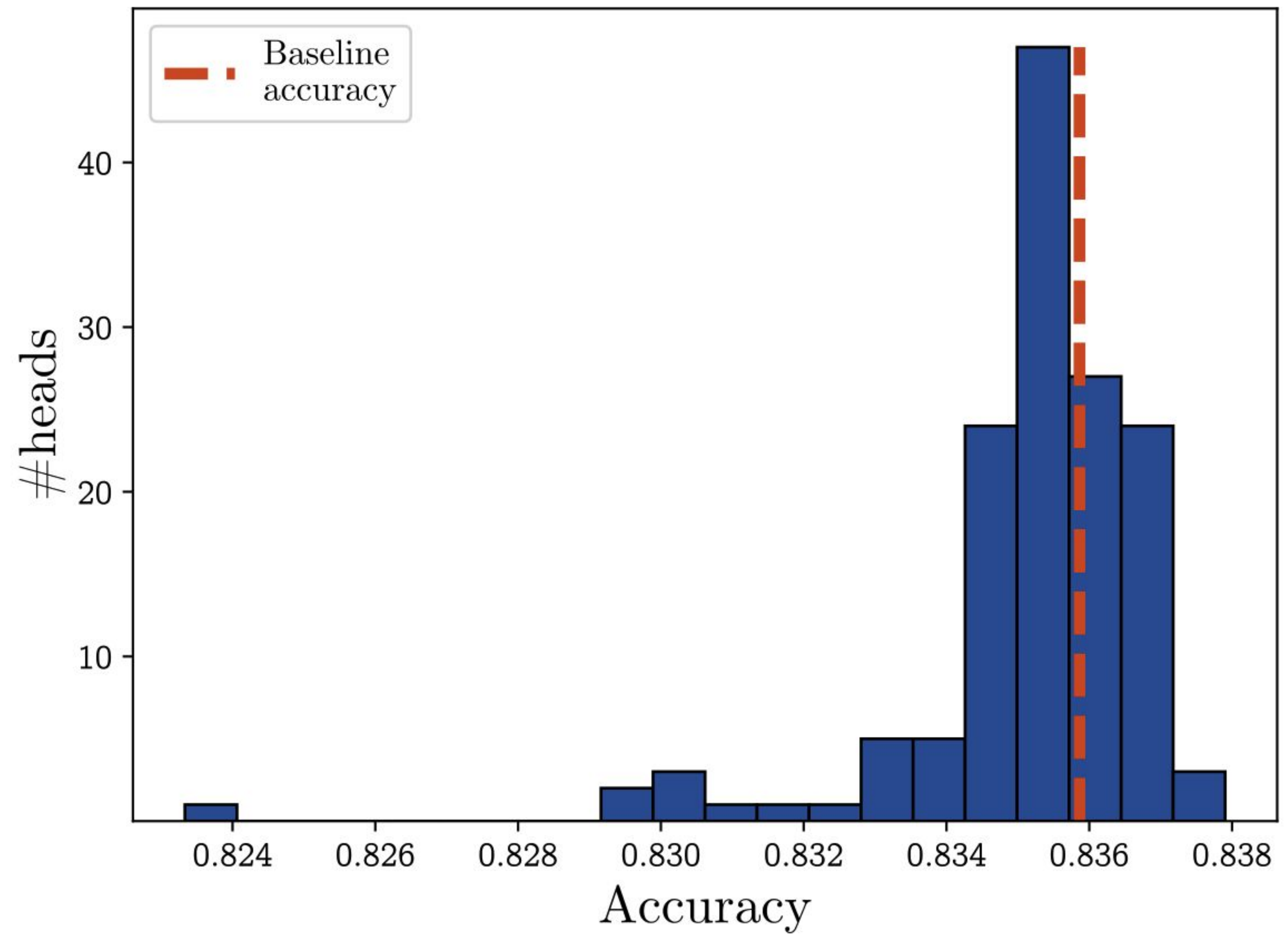
- L0 regularization over attention heads' mask parameters
  - Example: Translation task

$$L = L_{xent} + \lambda L_C \qquad \lambda = 0.01$$

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		
Weight Factorization		
Weight Sharing		
Knowledge distillation		
Sub-quadratic Transformer		

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!

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

Quantization

- How else can we compress a given neural module?

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

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

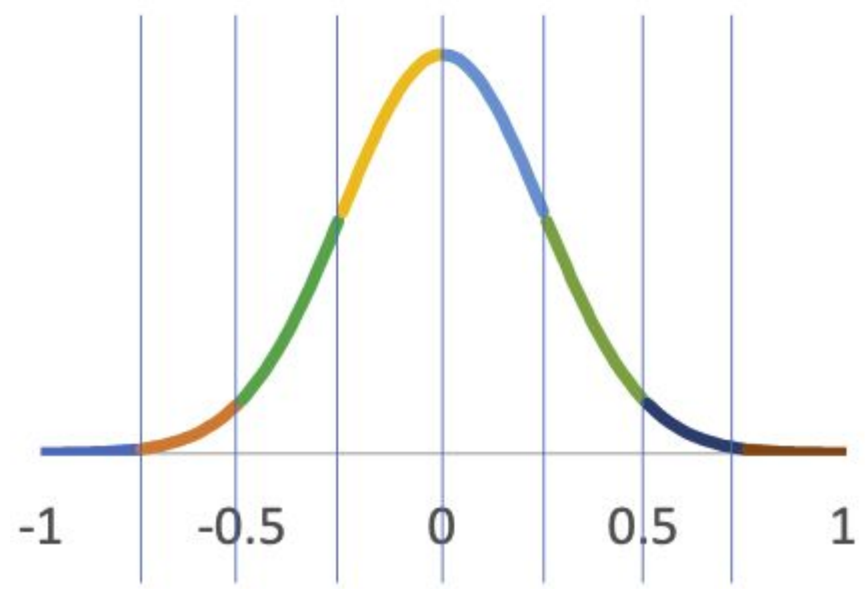
$$w_b = \begin{cases} c_1 & \text{if } w \geq (c_1 + c_2)/2 \\ c_2 & \text{if } w < (c_1 + c_2)/2 \end{cases}$$

- 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).

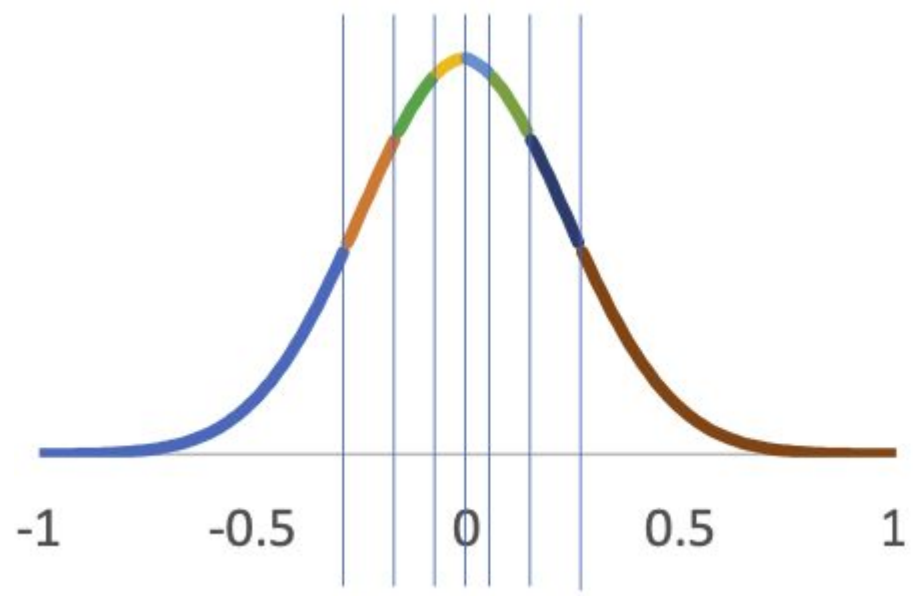
General Quantized Networks

- Uniform Quantization
  - Not necessarily optimal



Quantization with 3 bits

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



Quantization with 3 bits

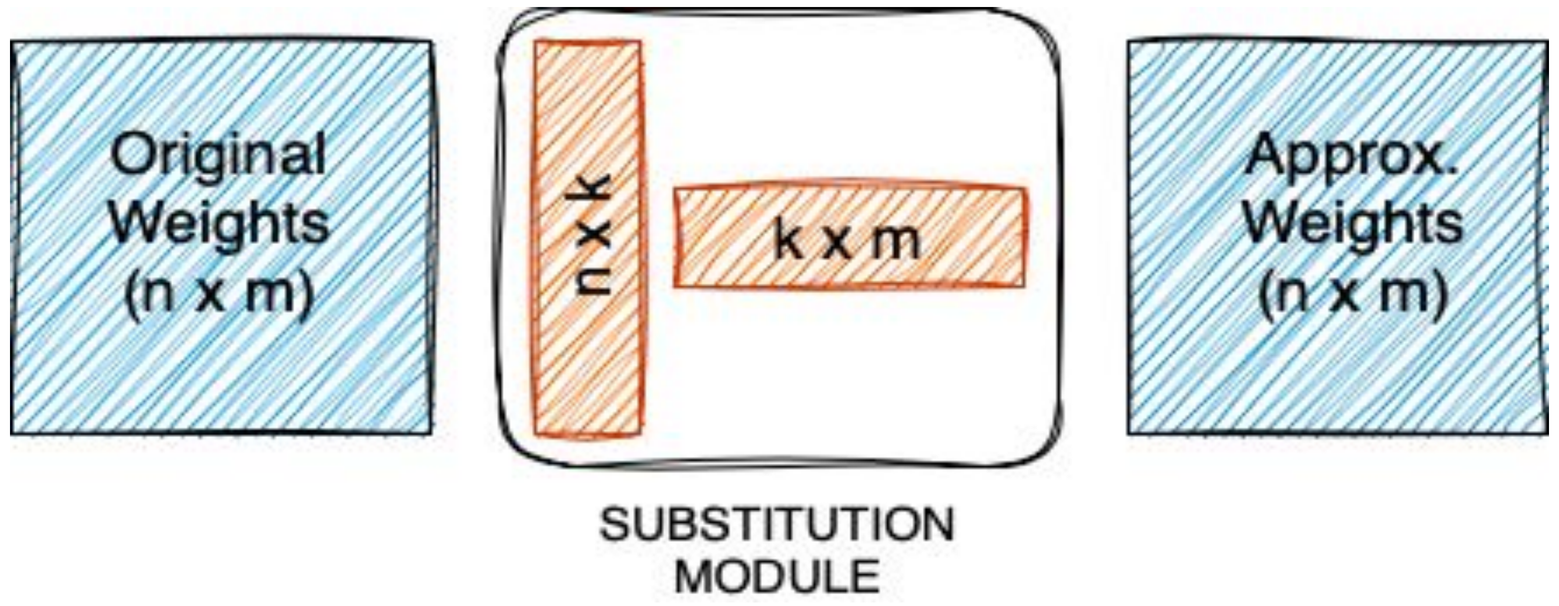
Gupta, Manish, and Puneet Agrawal. "Compression of deep learning models for text: A survey." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.4 (2022): 1-55..

Methods Overview

Approach	Improvement on memory footprint	Improvement on inference time
Pruning	Y/N	Y/N
Quantization	Yes	Yes
Weight Factorization		
Weight Sharing		
Knowledge distillation		
Sub-quadratic Transformer		

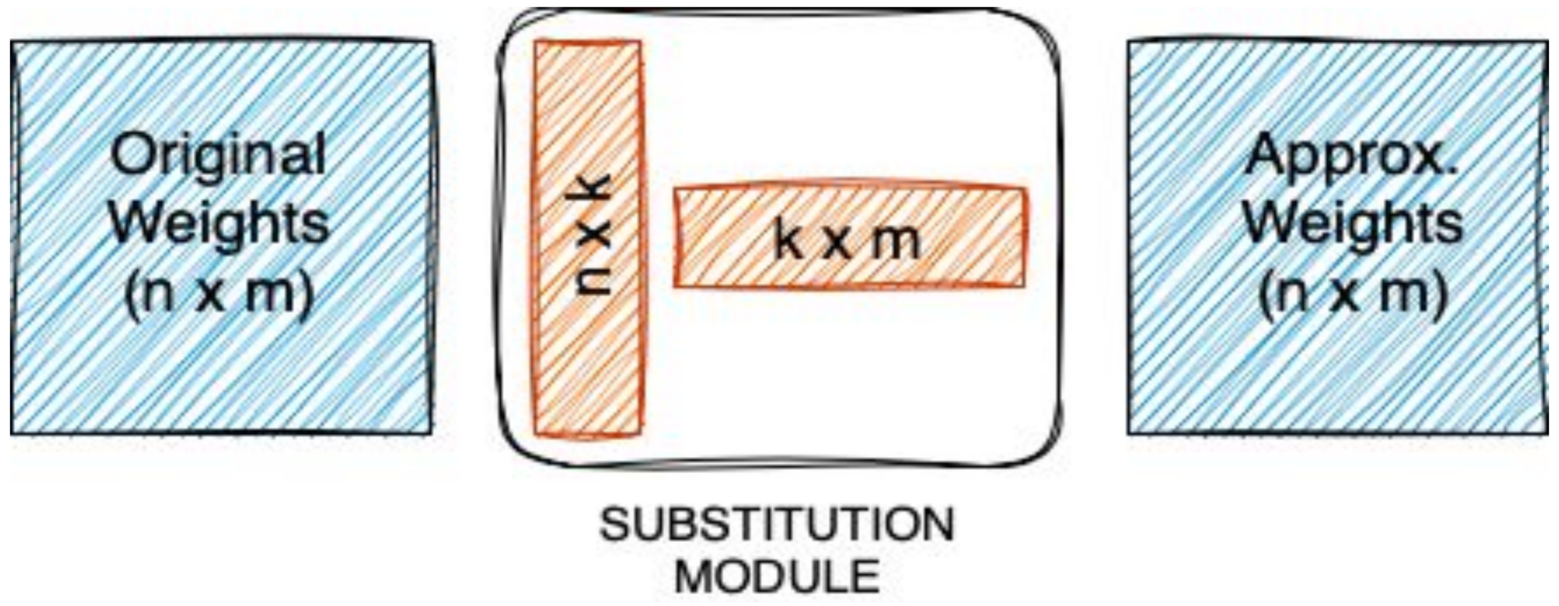
Weight Factorization

- The weight modules are replaced by their factorized matrices



Weight Factorization

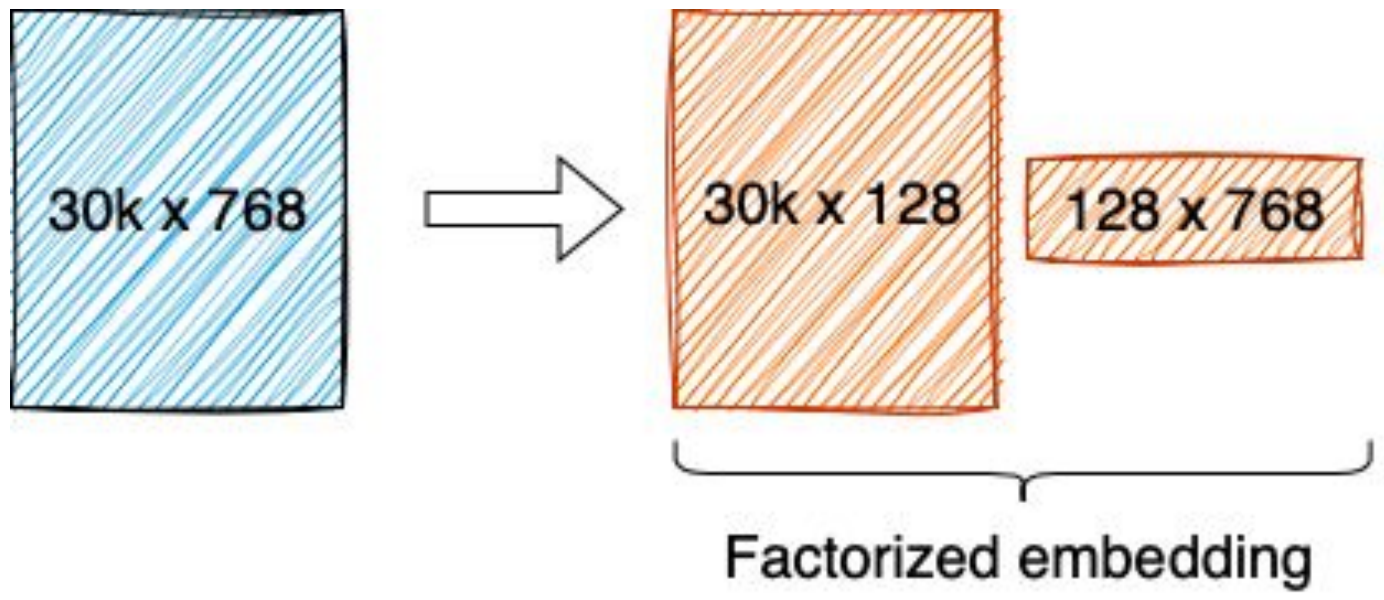
- The weight modules are replaced by their factorized matrices



- Factorization methods
  - Two low-rank matrices (similar to SVD)
  - Tensor decomposition
  - Non-linear factorization by using Auto-encoders

Case Study: ALBERT

- #parameters issue in token embedding matrix
  - ~23M out of 110M parameters in BERT-base
  - More than half of the parameters in mBERT
- Token embedding dimension are generally tied to the hidden dimension
  - What if we disentangle them by factorizing the embedding matrix?



Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).

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		
Knowledge distillation		
Sub-quadratic Transformer		

Weight Sharing

- A common example of parameter compression/efficiency
    - Finding weight blocks that can share the same weight

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.

Weight Sharing

- A common example of parameter compression/efficiency
  - Finding weight blocks that can share the same weight
- Examples of weight sharing
  - Sharing token embedding and LM decoder head
  - Parameter sharing in the embedding matrix
  - Cross-layer parameter sharing (e.g., ALBERT)

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
Weight Factorization	Yes	No
Weight Sharing	Yes	No
Knowledge distillation		
Sub-quadratic Transformer		

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

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

- 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. $\xrightarrow{\text{BERT-base}}$

```
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.
```

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 MLM + distilling hidden states

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 MLM + distilling hidden states
- 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
Weight Factorization	Yes	No
Weight Sharing	Yes	No
Knowledge distillation	Yes…	Yes…
Sub-quadratic Transformer		

Processing Long Contexts

- Issue with trained position embeddings
  - Example: BERT model

Token embedding $\boldsymbol{x}_i + \boldsymbol{p}_i$ positional embedding

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Processing Long Contexts

- Issue with trained position embeddings
  - Example: BERT model
- Sinusoidal position embedding
  - Example: (original) Transformer paper

Token embedding　　positional embedding

$$\boldsymbol{x}_i + \boldsymbol{p}_i$$

$$\omega_k = \frac{1}{10000^{2k/d}} \qquad \vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Processing Long Contexts

- Issue with trained position embeddings
  - Example: BERT model
- Sinusoidal position embedding
  - Example: (original) Transformer paper
- Relative positional encoding
  - Rotary Position Embedding (RoPE)
  - Attention with Linear Biases (ALiBi)

Token embedding

positional embedding

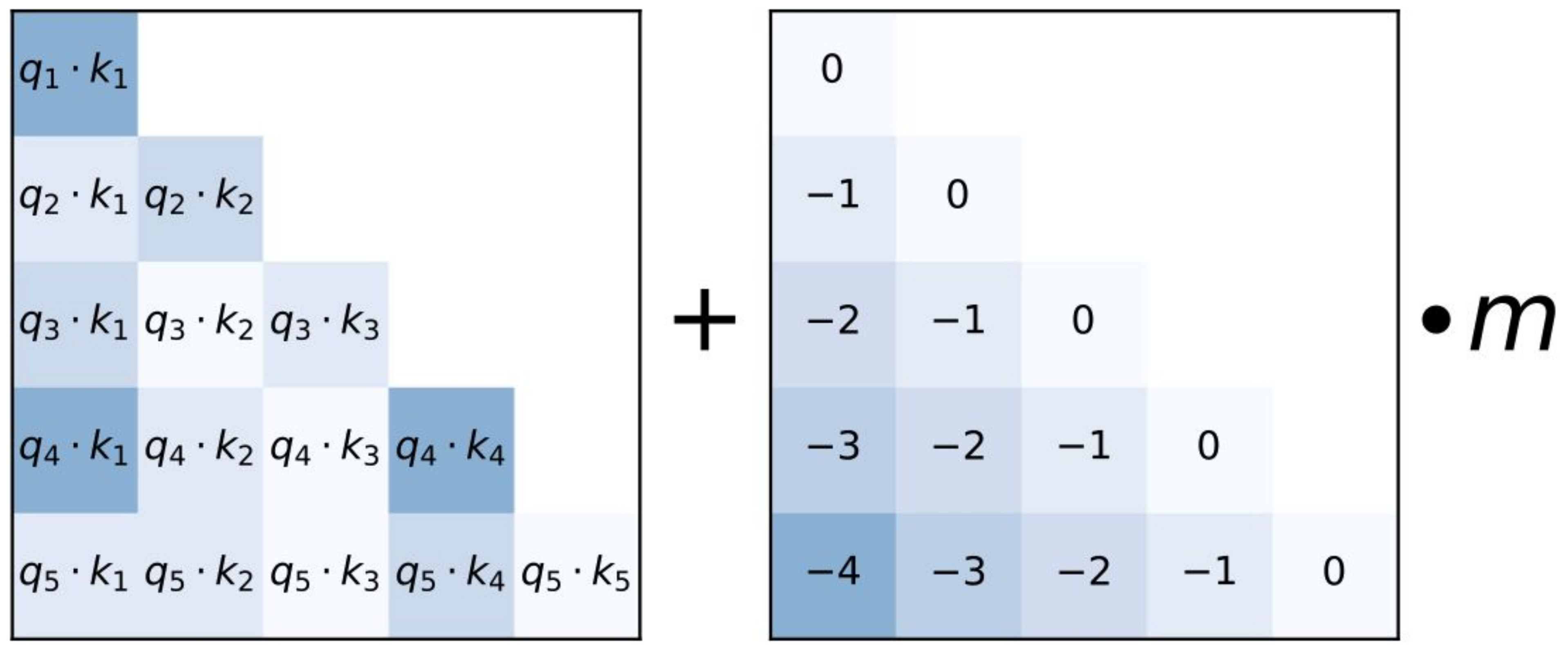
$$\boldsymbol{x}_i + \boldsymbol{p}_i$$

$$\omega_k = \frac{1}{10000^{2k/d}} \qquad \vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
Su, Jianlin, et al. "Roformer: Enhanced transformer with rotary position embedding." *arXiv preprint arXiv:2104.09864* (2021).
Press, Ofir, Noah A. Smith, and Mike Lewis. "Train short, test long: Attention with linear biases enables input length extrapolation." *arXiv preprint arXiv:2108.12409* (2021).

ALiBi

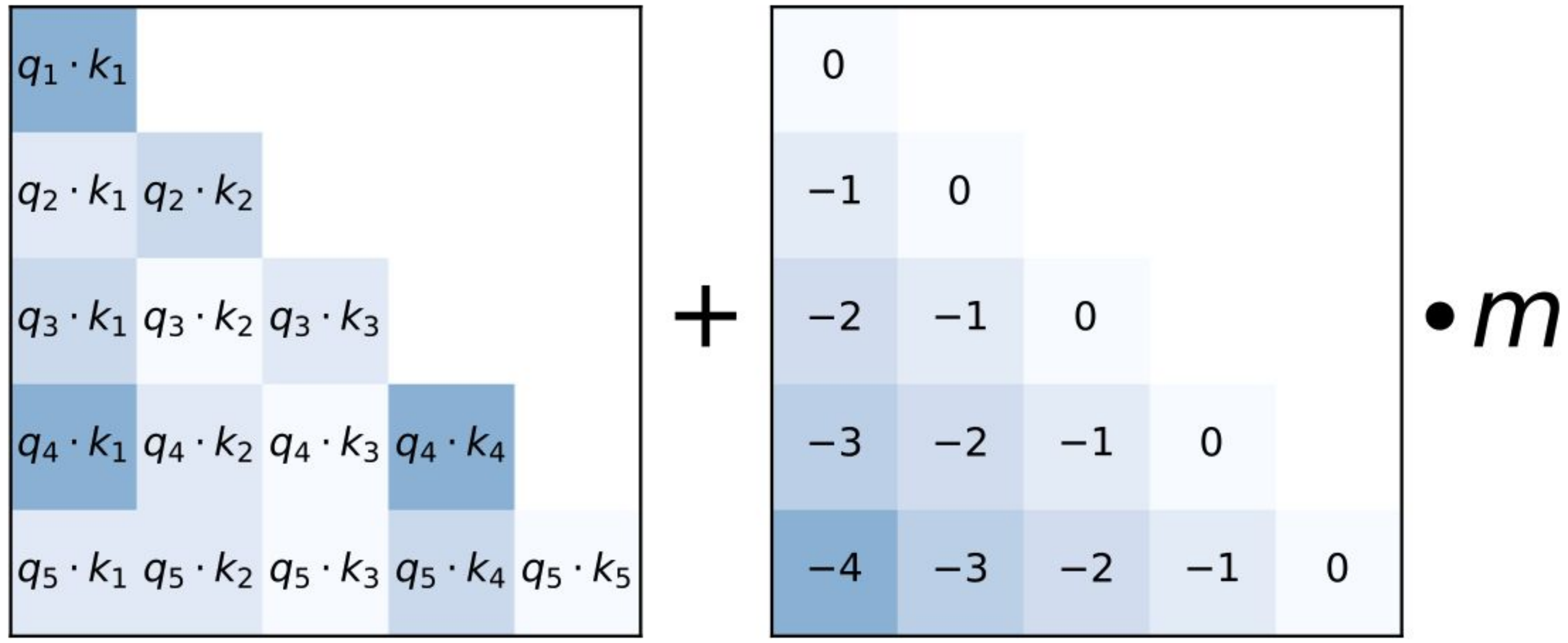
- No additive position embeddings in the input layer
- adding a linear bias to each attention score

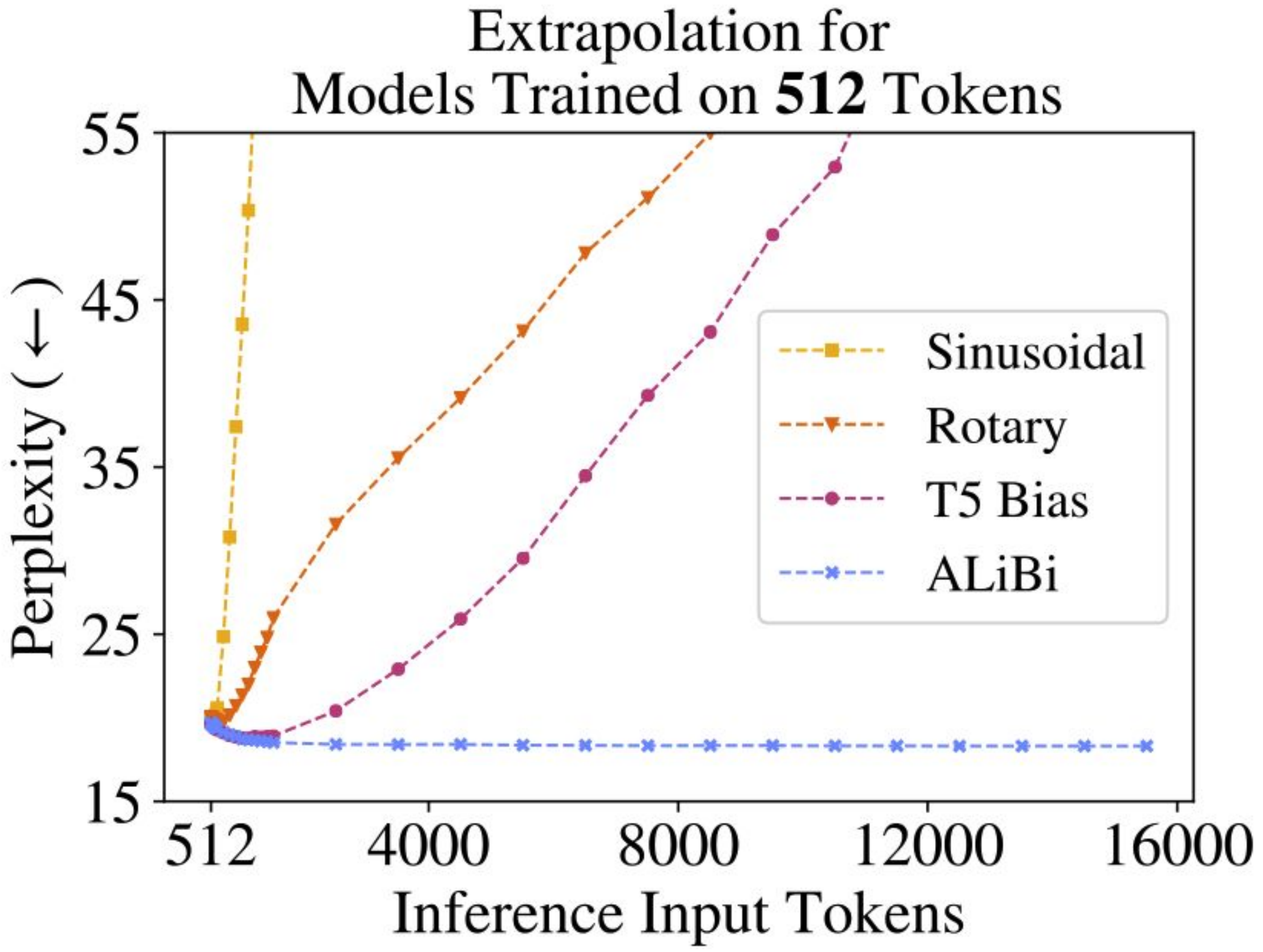


Press, Ofir, Noah A. Smith, and Mike Lewis. "Train short, test long: Attention with linear biases enables input length extrapolation." *arXiv preprint arXiv:2108.12409* (2021).

ALiBi

- No additive position embeddings in the input layer
- adding a linear bias to each attention score





Press, Ofir, Noah A. Smith, and Mike Lewis. "Train short, test long: Attention with linear biases enables input length extrapolation." *arXiv preprint arXiv:2108.12409* (2021).

Inference time for long inputs?

Sub-quadratic Transformers

- Time and activation memory grows quadratically with the sequence length
    - Especially important for long sequences
    - Potentially limiting the maximum sequence length

Qiu, Jiezhong, et al. "Blockwise self-attention for long document understanding." *arXiv preprint arXiv:1911.02972*(2019).

Sub-quadratic Transformers

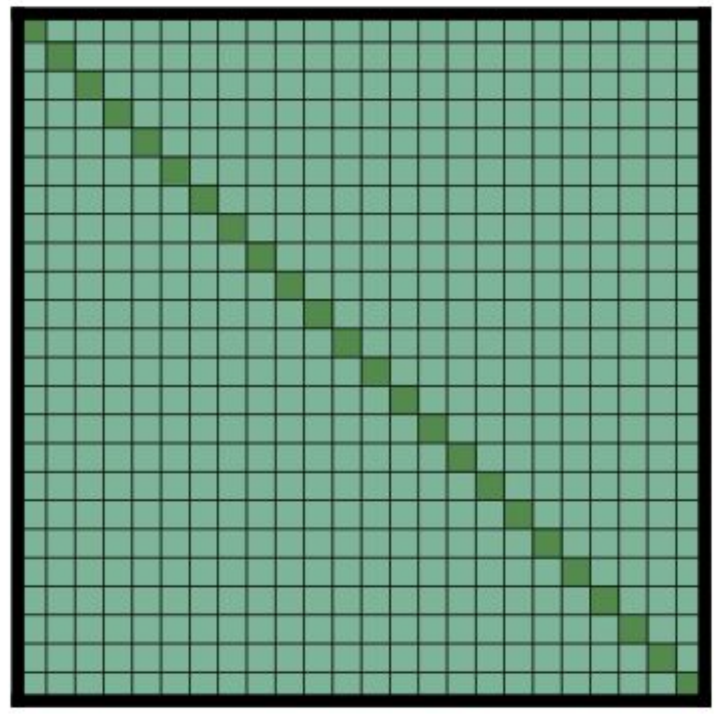
- Time and activation memory grows quadratically with the sequence length
  - Especially important for long sequences
  - Potentially limiting the maximum sequence length
- Do tokens need to directly attend to every other token?
  - What if attention is performed more locally! → Longformer
    - Masking attention between far tokens (using M matrix)

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{M}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}} \odot \boldsymbol{M}\right)\boldsymbol{V}$$

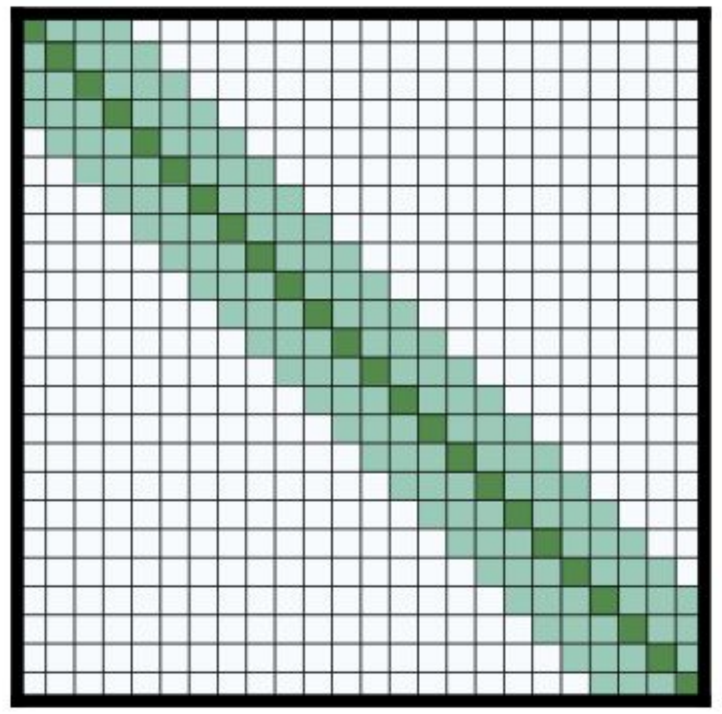
Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150* (2020).

Longformer

- Every token should attend to its neighbor tokens



Full n^2 attention

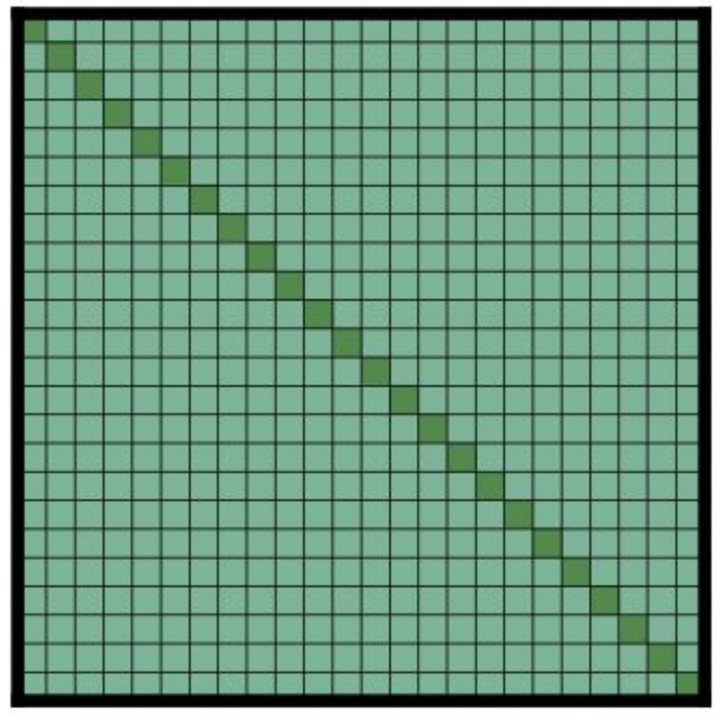


Sliding window attention

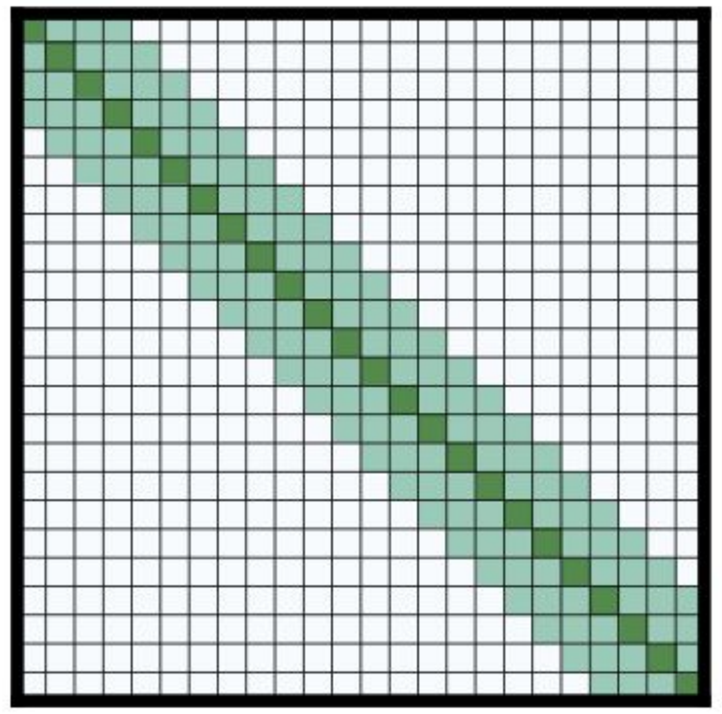
Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150* (2020).

Longformer

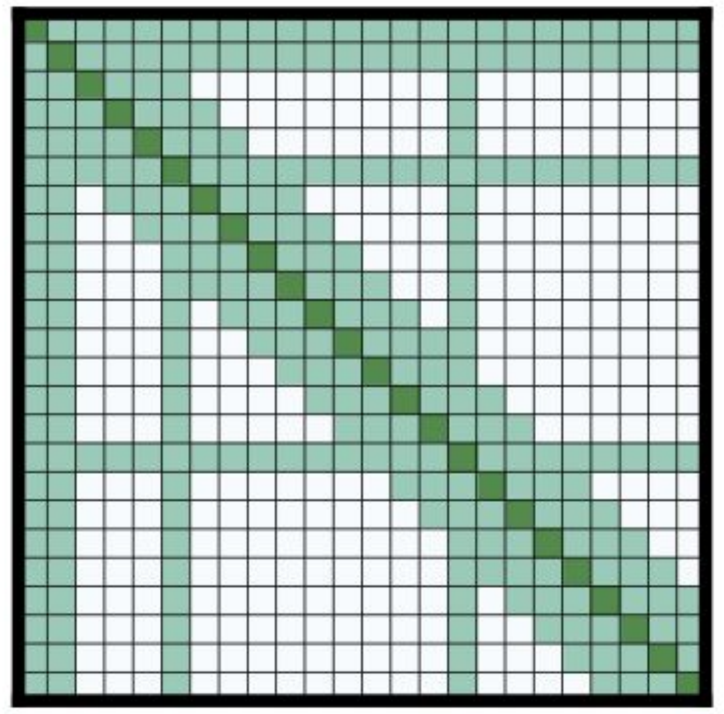
- Every token should attend to its neighbor tokens
- Need for some global tokens to bridge across the sequence
  - [CLS] for text classification
  - Question tokens for QA



Full n^2 attention



Sliding window attention



Global+sliding window

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150* (2020).

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…
Sub-quadratic Transformer	No	Yes

Recap

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