EMNLP 2021
7th – 11th November | Online and in the Dominican Republic

Less Is More : Domain Adaptation with Lottery Ticket for Reading Comprehension

Haichao Zhu, Zekun Wang, Heng Zhang, Ming Liu, Sendong Zhao, Bing Qin

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Reading Comprehension - Task

- Answering questions according to the context
  - Extractive answers are continuous context spans

Context: *Victorian schools are either publicly or privately funded. Public schools, also known as state or government schools, are funded and run directly by the Victoria Department of Education. Students do not pay tuition fees, but some extra costs are levied. Private fee-paying schools include parish schools run by the Roman Catholic Church and independent schools similar to British public schools. Independent schools are usually affiliated with Protestant churches. Victoria also has several private Jewish and Islamic primary and secondary schools.*

Question: *What organization runs the public schools in Victoria?*
Answer: *Victoria Department of Education*

Question: *Since students do not pay tuition, what do they have to pay for schooling in Victoria?*
Answer: *some extra costs*

Question: *What church runs some private schools in Victoria?*
Answer: *Roman Catholic Church*

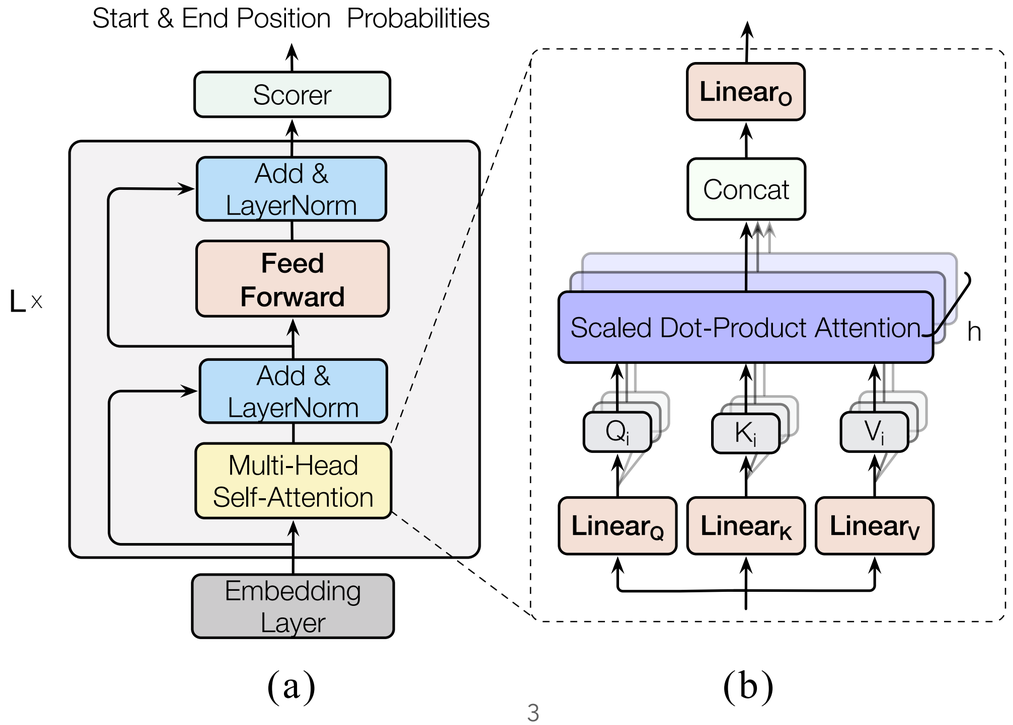
Reading Comprehension - SOTA

Reading Comprehension - SOTA

- Build on large pre-trained language models with at least **hundreds of millions** parameters

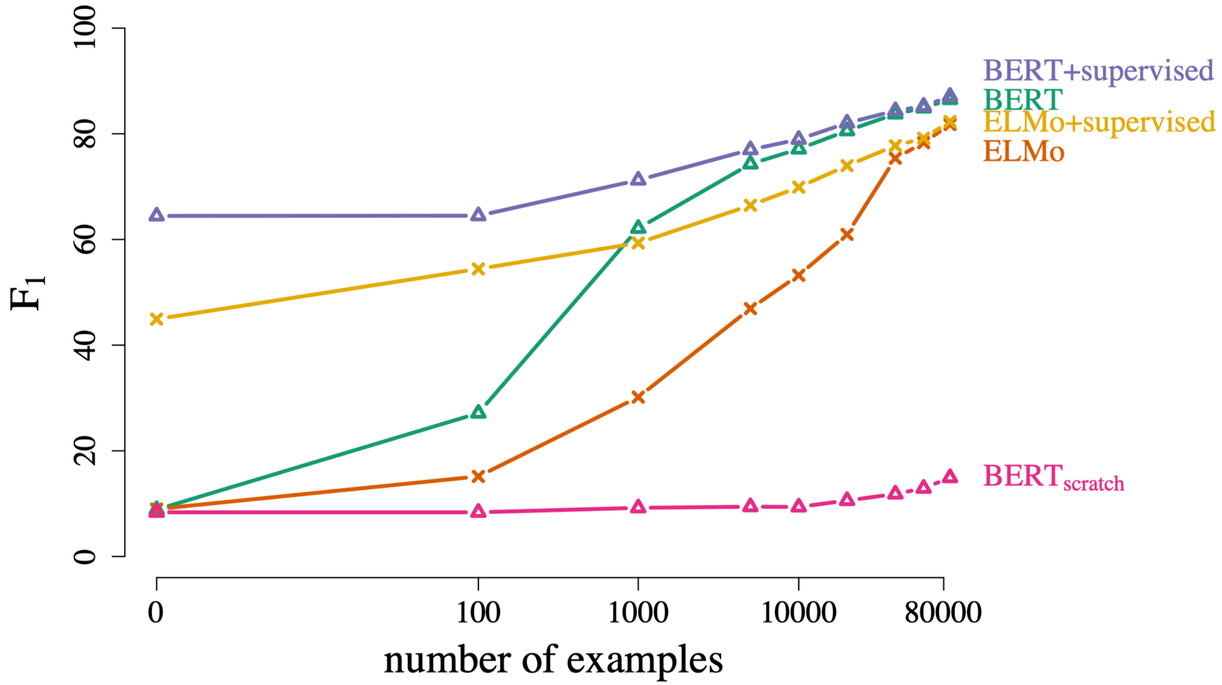
Reading Comprehension - SOTA

- Build on large pre-trained language models with at least **hundreds of millions** parameters
  - BERT, XLNET, RoBERTa, Albert, etc.



Reading Comprehension - SOTA

- Build on large pre-trained language models with at least **hundreds of millions** parameters
  - BERT, XLNET, RoBERTa, Albert, etc.
- Require large amounts of annotated data



Learning and evaluating general linguistic intelligence. Yogatama, Dani, et al., 2019.

Reading Comprehension - SOTA

- Build on large pre-trained language models with at least **hundreds of millions** parameters
  - BERT, XLNET, RoBERTa, Albert, etc.
- Require large amounts of annotated data
  - Generalize poorly

	SQuAD	NewsQA	SearchQA	TQA-G	TQA-W	HotpotQA
SQuAD	-	**31.8**	8.4	37.8	33.4	**11.8**
NewsQA	**60.4**	-	10.1	37.6	28.4	8.0
SearchQA	23.3	12.7	-	**53.2**	**35.4**	5.2
TQA-G	36.3	18.8	**39.2**	-	-	8.8
TQA-W	35.5	19.4	27.8	-	-	8.7
HotpotQA	54.5	25.6	19.6	37.3	34.9	-
Multi-75K	-	-	-	-	-	-
Self	78.0	46.0	52.2	60.7	50.1	24.2

MultiQA: An Empirical Investigation of Generalization and Transfer in, Talmor and Berant 2019

Reading Comprehension - Domain Adaptation

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)
  - Adapting language model (Nishida et al., 2020;)

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)
  - Adapting language model (Nishida et al., 2020;)
  - Adversarial training (Wang et al., 2019;Cao et al., 2020)

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)
  - Adapting language model (Nishida et al., 2020;)
  - Adversarial training (Wang et al., 2019;Cao et al., 2020)
- Supervised domain adaptation in few-shot settings

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)
  - Adapting language model (Nishida et al., 2020;)
  - Adversarial training (Wang et al., 2019;Cao et al., 2020)
- Supervised domain adaptation in few-shot settings
  - Large amount of annotated data in source domain 

Reading Comprehension - Domain Adaptation

- Unsupervised domain adaptation
  - Generate synthetic questions (Wang et al., 2019;Shakeri et al., 2020; Rennie et al., 2020)
  - Adapting language model (Nishida et al., 2020;)
  - Adversarial training (Wang et al., 2019;Cao et al., 2020)
- Supervised domain adaptation in few-shot settings
  - Large amount of annotated data in source domain 
  - Limited annotation in target domain 

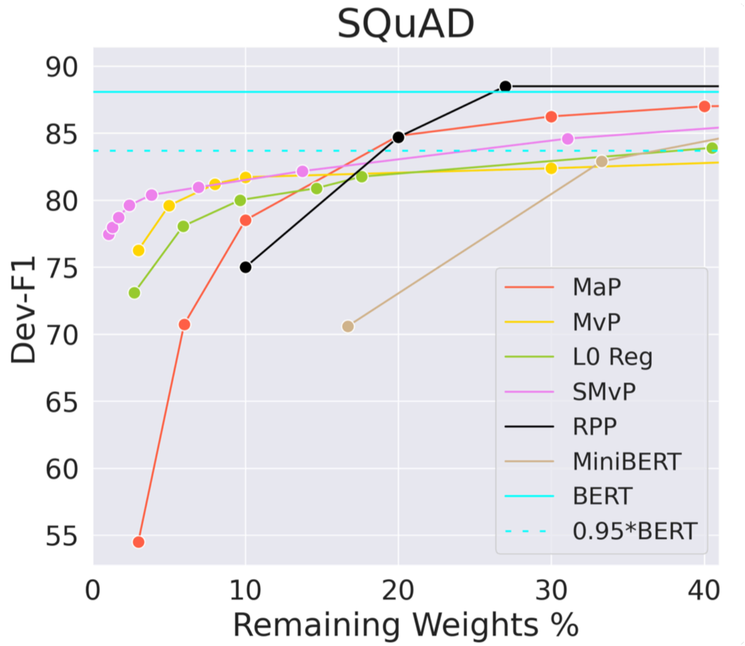
Over-parameterized Neural Models

Over-parameterized Neural Models

- Neural networks are over-parameterized, a great fraction of the parameters can be pruned with minimal or even no compromise in performance

Over-parameterized Neural Models

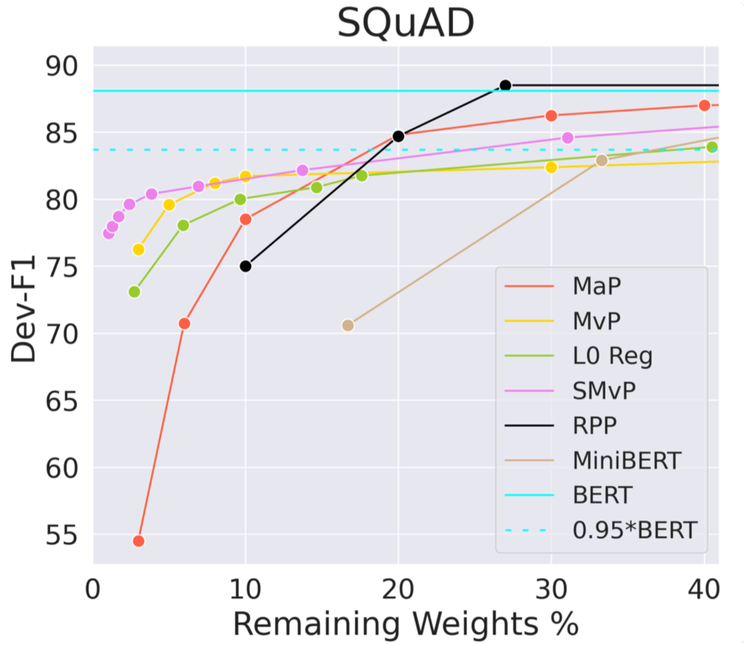
- Neural networks are over-parameterized, a great fraction of the parameters can be pruned with minimal or even no compromise in performance
  - Unstructured pruning (Frankle et al., 2020; Sanh et al., 2020; Gordon et al., 2020)

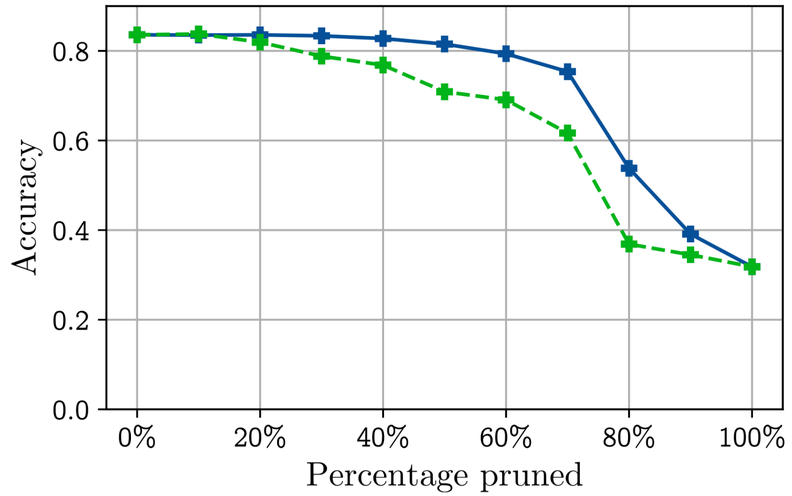


Movement pruning: Adaptive sparsity by finetuning. Sanh et al., 2020

Over-parameterized Neural Models

- Neural networks are over-parameterized, a great fraction of the parameters can be pruned with minimal or even no compromise in performance
  - Unstructured pruning (Frankle et al., 2020; Sanh et al., 2020; Gordon et al., 2020)
  - Structured pruning (Michel et al., 2019; Voita et al., 2019; McCarley et al., 2019; Fan et al., 2020)





(b) Evolution of accuracy on the MultiNLI-matched validation set when heads are pruned from BERT.

Movement pruning: Adaptive sparsity by finetuning. Sanh et al., 2020
Are sixteen heads really better than one? Michel et al., 2019

Self-Attention Head Importance

Self-Attention Head Importance

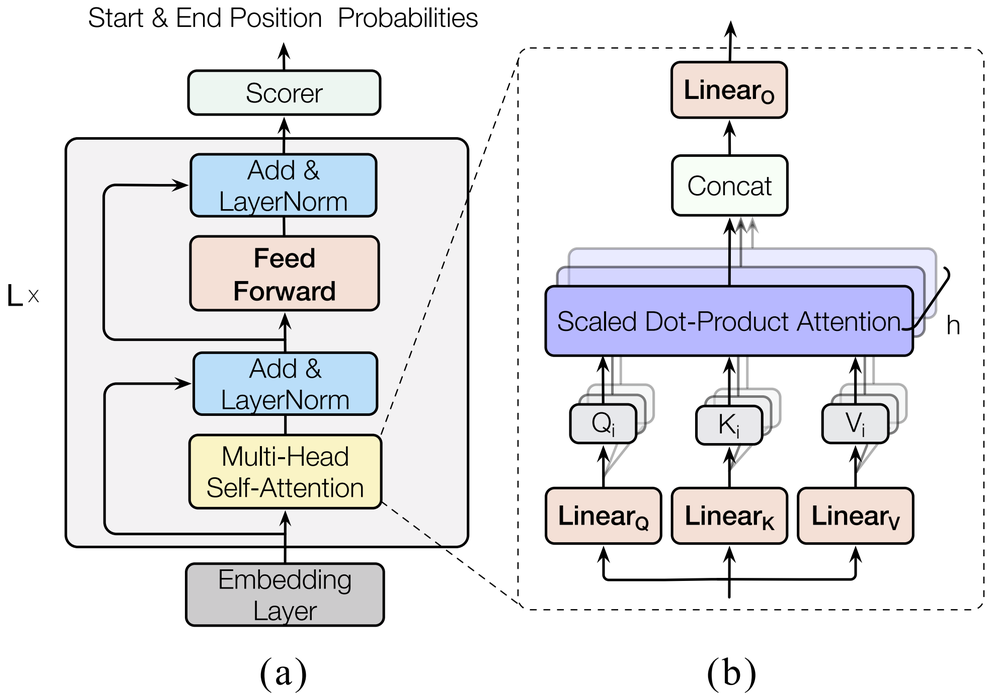
- Self-Attention Attribution (Hao et al. 2021)

Self-Attention Head Importance

- Self-Attention Attribution (Hao et al. 2021)

  1. Compute self-attention map of i-th head

$$\mathbf{A}_i = \text{softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d_k}})$$



Self-Attention Head Importance

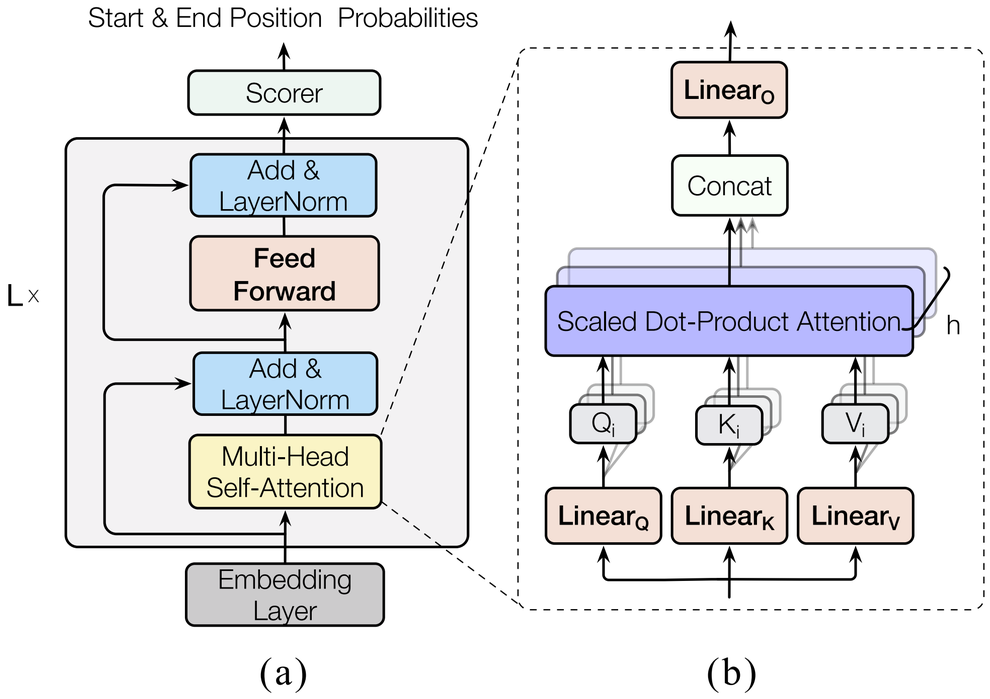
- Self-Attention Attribution (Hao et al. 2021)

1. Compute self-attention map of i-th head

$$\mathbf{A}_i = \text{softmax}(\frac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d_k}})$$

2. Compute gradient with manipulated map

$$\text{Attr}(\mathbf{A}_i) = \mathbf{A}_i \odot \int_{\alpha=0}^{1} \frac{\partial \mathcal{F}(\mathrm{x}, \alpha\mathbf{A})}{\partial \mathbf{A}_i} \mathrm{d}\alpha \in \mathbb{R}^{n\times n}$$



Self-Attention Head Importance

- Self-Attention Attribution (Hao et al. 2021)

1. Compute self-attention map of i-th head

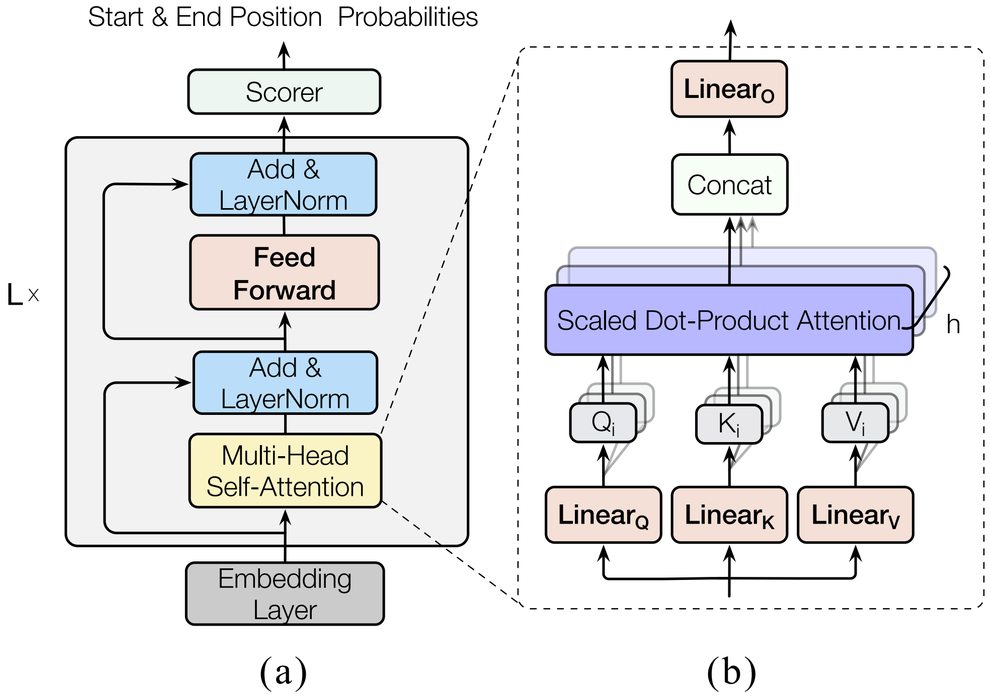
$$\mathbf{A}_i = \text{softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d_k}})$$

2. Compute gradient with manipulated map

$$\text{Attr}(\mathbf{A}_i) = \mathbf{A}_i \odot \int_{\alpha=0}^{1} \frac{\partial \mathcal{F}(\mathrm{x}, \alpha \mathbf{A})}{\partial \mathbf{A}_i} \mathrm{d}\alpha \in \mathbb{R}^{n \times n}$$

3. Estimate the head importance score

$$I_i = E_{\mathrm{x}} \left[\max(\text{Attr}(\mathbf{A}_i)) \right]$$

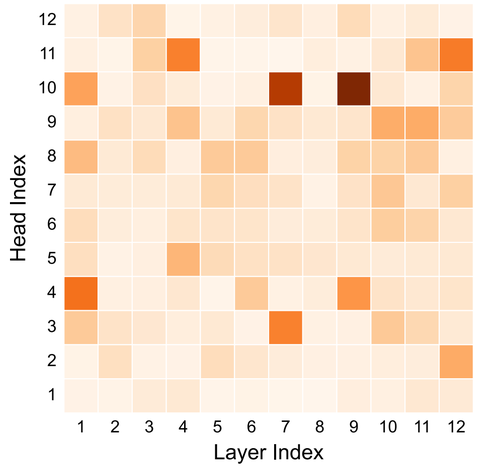


Self-Attention Head Importance Analysis

- Important attention heads are **highly correlated** across various domains
  - e.g., SQuAD, NewsQA, NQ

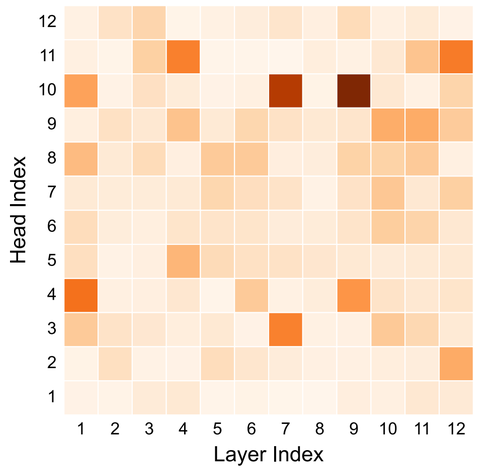
Self-Attention Head Importance Analysis

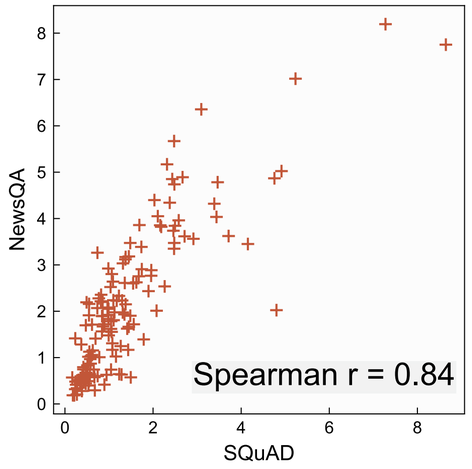
- Important attention heads are **highly correlated** across various domains
  - e.g., SQuAD, NewsQA, NQ

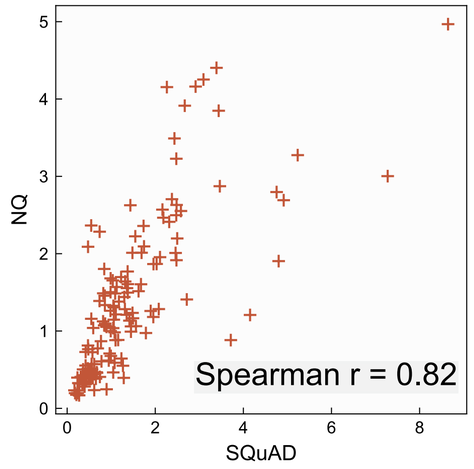


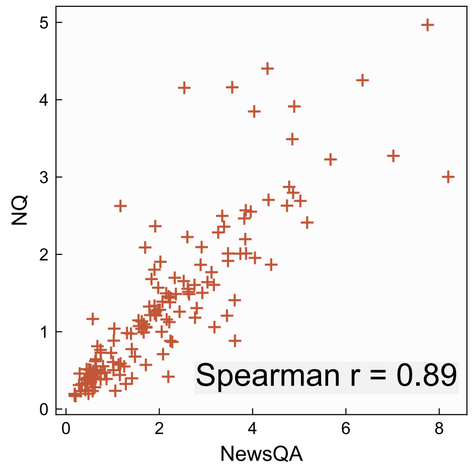
Self-Attention Head Importance Analysis

- Important attention heads are **highly correlated** across various domains
  - e.g., SQuAD, NewsQA, NQ

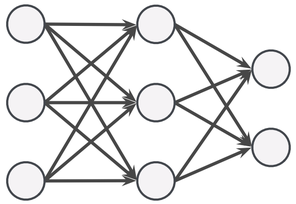








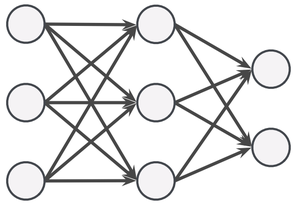
Our Work



Source Domain Model

Our Work

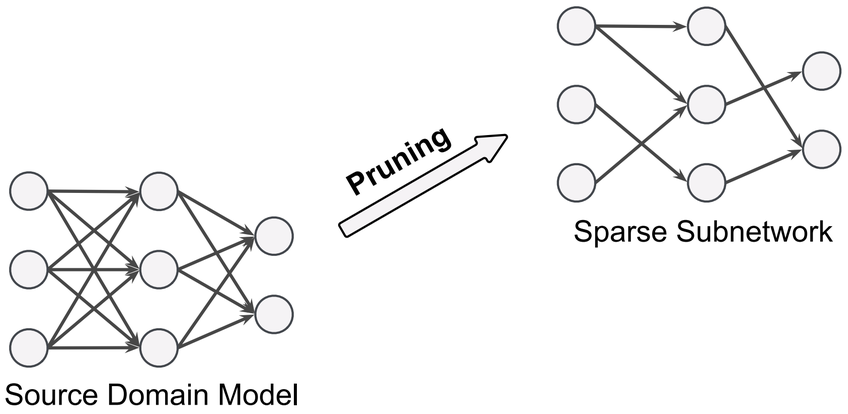
- Exploit a small fraction of deliberately selected parameters for adaptation



Source Domain Model

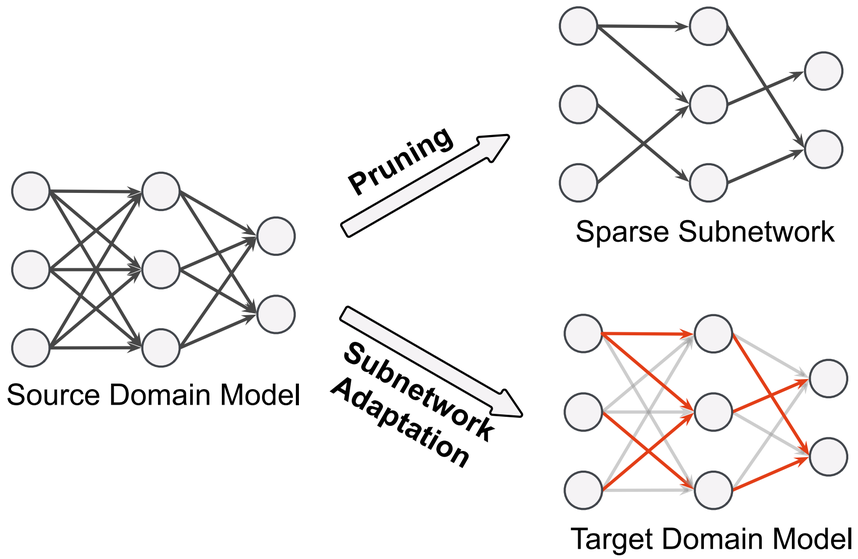
Our Work

- Exploit a small fraction of deliberately selected parameters for adaptation
  1. Identify lottery subnetworks of the source domain model with self-attention head importance



Our Work

- Exploit a small fraction of deliberately selected parameters for adaptation
  1. Identify lottery subnetworks of the source domain model with self-attention head importance
  2. Adapt to the target domain by only updating the parameters of the subnetworks



Identifying the Lottery Network

Identifying the Lottery Network

1. Estimate and normalize attention head importance

$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

λ indicates the intensity of head importance intervention

Identifying the Lottery Network

1. Estimate and normalize attention head importance

$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

2. Trim magnitudes with normalized importance score

$$\hat{\theta}_{(n-1)\nabla t} \leftarrow AttrMagnitude(\theta_{(n-1)\nabla t}, \hat{I}_n)$$

Identifying the Lottery Network

1. Estimate and normalize attention head importance

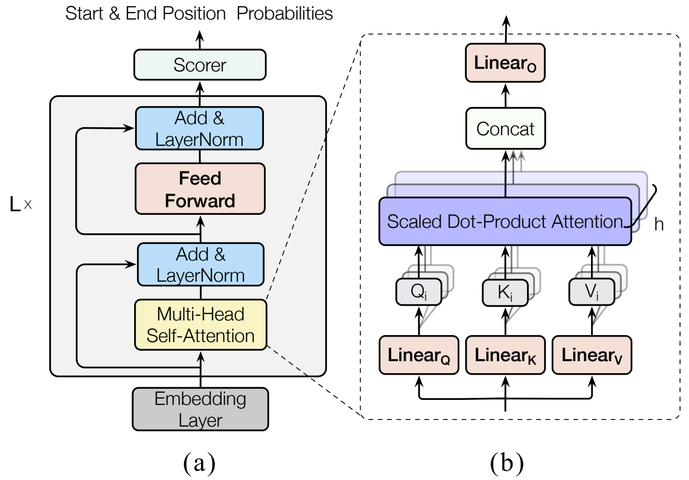
$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

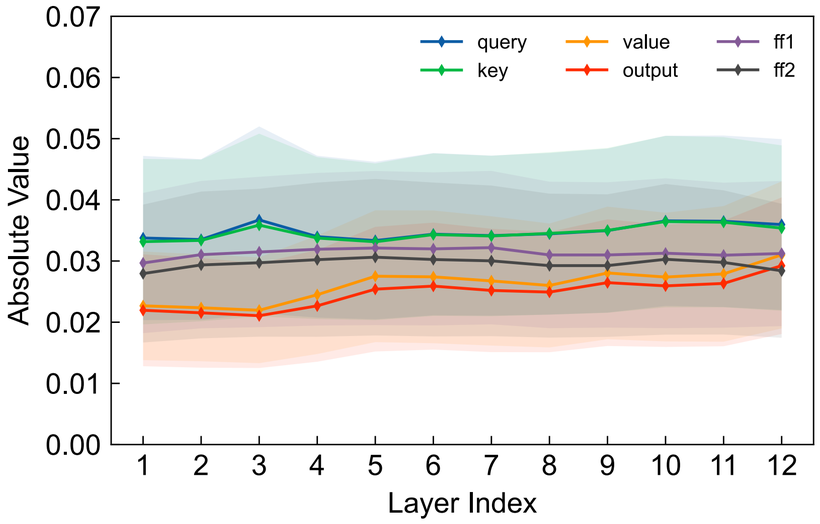
2. Trim magnitudes with normalized importance score

$$\hat{\theta}_{(n-1)\nabla t} \leftarrow AttrMagnitude(\theta_{(n-1)\nabla t}, \hat{I}_n)$$

3. Prune the lowest magnitudes parameters in group from $\hat{\theta}_{(n-1)\nabla t}$ to sparsity s_n

$$s_n \leftarrow s - s(1 - \frac{n}{N})^2$$





Identifying the Lottery Network

1. Estimate and normalize attention head importance

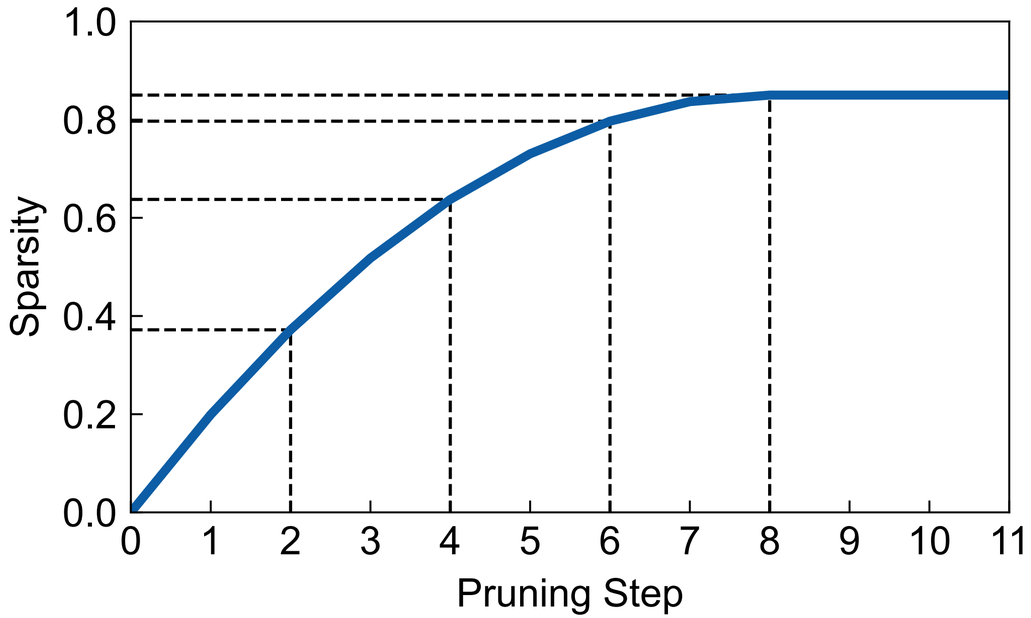
$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

2. Trim magnitudes with normalized importance score

$$\hat{\theta}_{(n-1)\nabla t} \leftarrow AttrMagnitude(\theta_{(n-1)\nabla t}, \hat{I}_n)$$

3. Prune the lowest magnitudes parameters in group from $\hat{\theta}_{(n-1)\nabla t}$ to sparsity s_n

$$s_n \leftarrow s - s(1 - \frac{n}{N})^2$$



Identifying the Lottery Network

1. Estimate and normalize attention head importance

$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

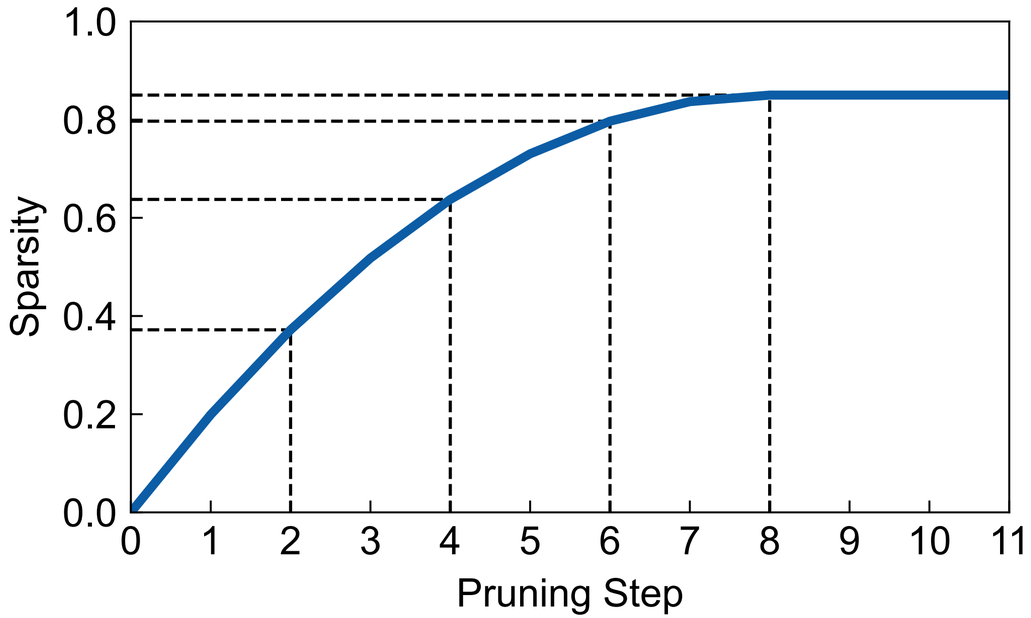
2. Trim magnitudes with normalized importance score

$$\hat{\theta}_{(n-1)\nabla t} \leftarrow AttrMagnitude(\theta_{(n-1)\nabla t}, \hat{I}_n)$$

3. Prune the lowest magnitudes parameters in group from $\hat{\theta}_{(n-1)\nabla t}$ to sparsity s_n

$$s_n \leftarrow s - s(1 - \frac{n}{N})^2$$

4. Train the model for ∇t steps



Identifying the Lottery Network

1. Estimate and normalize attention head importance

$$\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$$

2. Trim magnitudes with normalized importance score

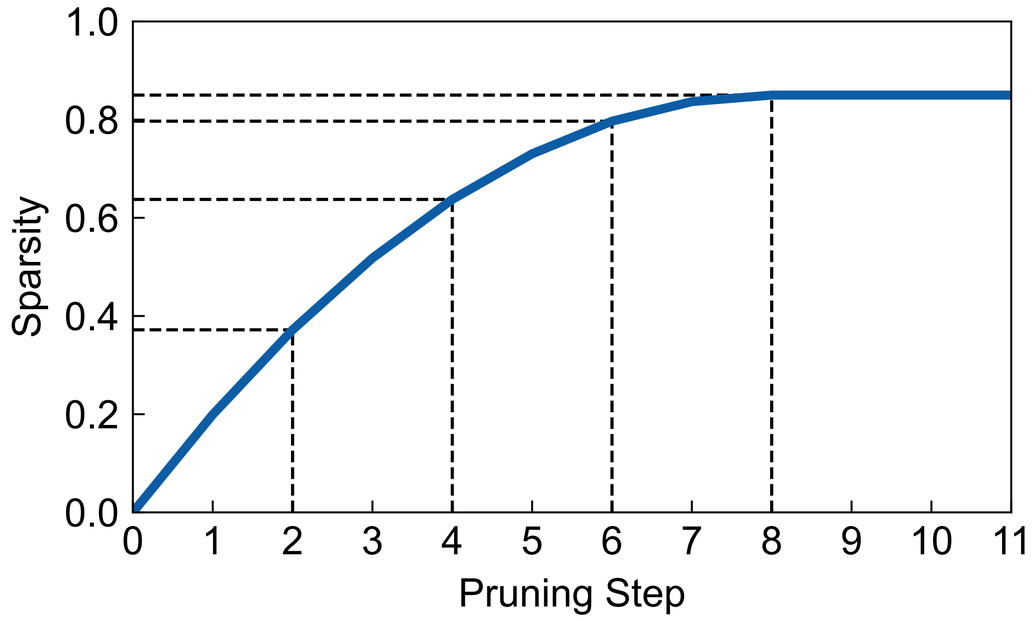
$$\hat{\theta}_{(n-1)\nabla t} \leftarrow AttrMagnitude(\theta_{(n-1)\nabla t}, \hat{I}_n)$$

3. Prune the lowest magnitudes parameters in group from $\hat{\theta}_{(n-1)\nabla t}$ to sparsity s_n

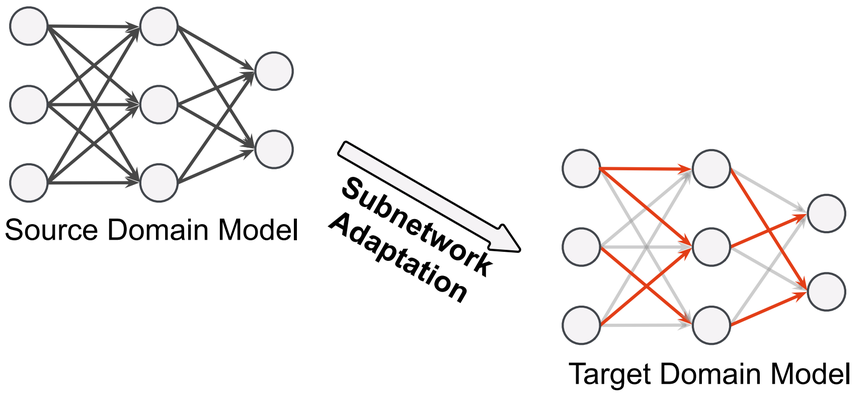
$$s_n \leftarrow s - s(1 - \frac{n}{N})^2$$

4. Train the model for ∇t steps

5. Repeat step 1 - 4 N times

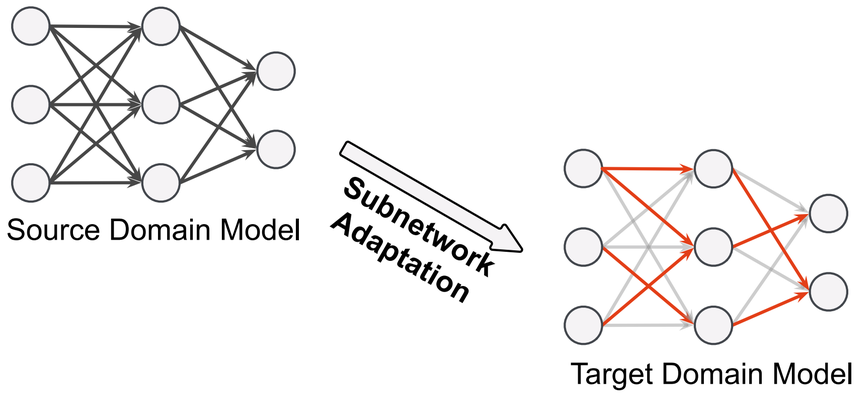


Adapting the Lottery Subnetwork



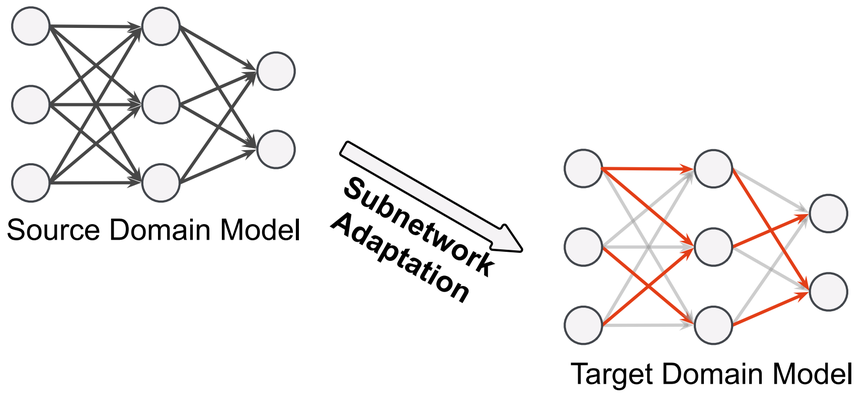
Adapting the Lottery Subnetwork

- Rewind to the original source domain model parameters



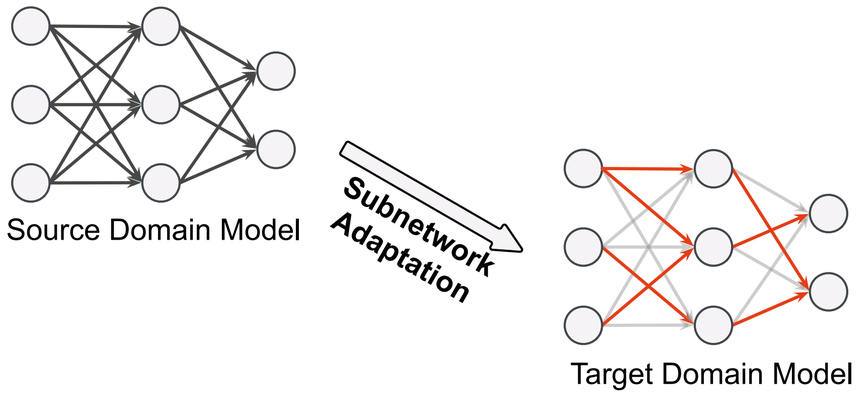
Adapting the Lottery Subnetwork

- Rewind to the original source domain model parameters
- Only update the parameters of the subnetworks



Adapting the Lottery Subnetwork

- Rewind to the original source domain model parameters
- Only update the parameters of the subnetworks
  - Pruned parameters are frozen, but participate in the forward computation



Experimental Settings

- Source domain: SQuAD
- Five target domains

Dataset	Context	Question	Q⊥C	Train	Dev
SQuAD	Wikipedia	Crowd	✘	87,599	10,507
NewsQA	News Articles	Crowd	✔	74,160	4,212
TriviaQA	Web Snippets	Trivia	✔	61,688	7,785
TweetQA	Tweets	Crowd	✘	7,108	883
NQ	Wikipedia	Queries	✔	104,071	12,836
QuAC	Wikipedia	Crowd	✔	51,695	4,368

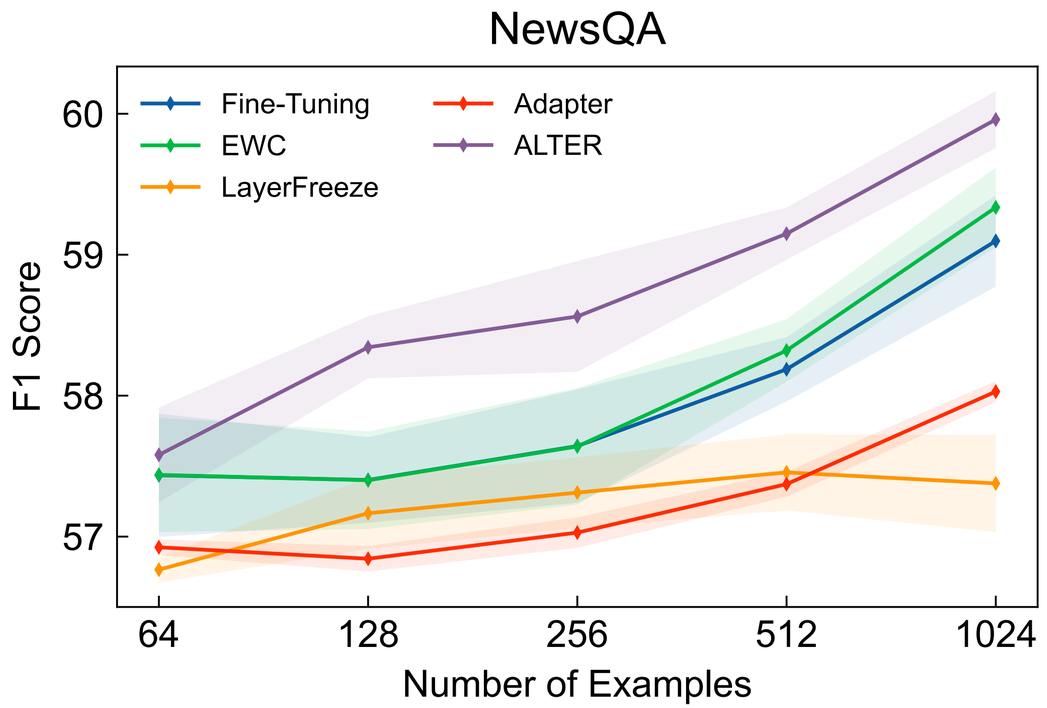
Experimental Results

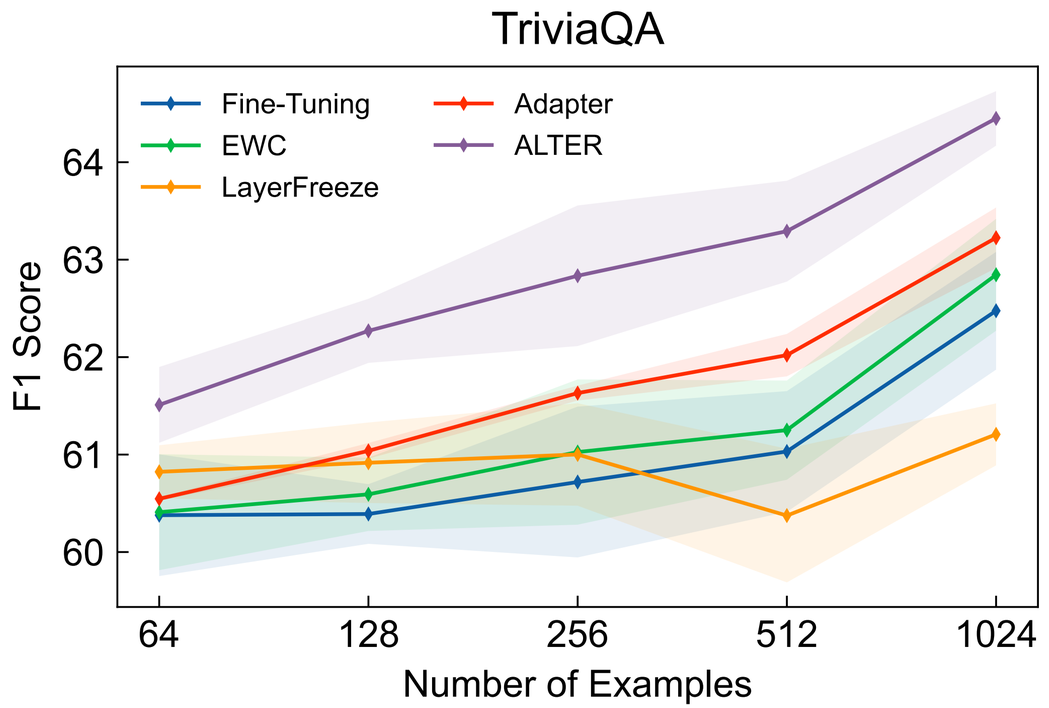
- 1024 target domain examples
- ~20M parameters for adaptation
- Our method **Alter** (**A**daptable **L**ott**e**ry)

Model	Training Parameters	NewsQA	TriviaQA	TweetQA	NQ	QuAC
Zero-Shot	None	40.05/56.76	50.52/60.11	67.46/79.48	43,49/57.45	15.82 /37.31
Fine-Tuning	84M	43.24/59.10	55.60/62.48	70.59/81.81	**55.23/68.68**	26.73/49.25
EWC	84M	43.44/59.34	55.95/62.85	70.48/81.82	55.09/68.54	26.82/49.37
LayerFreeze	21M	40.68/57.38	53.83/61.21	70.32/81.54	50.41/64.11	25.39/47.56
Adapter	20M	41.14/58.03	55.71/63.22	69.50/80.81	49.45/63.44	24.06/46.22
Alter	21M	**43.73/59.78**	**57.47/64.45**	**71.18/82.31**	54.62/68.17	**27.50/49.50**
Full Data	**84M**	**52.18/66.95**	**64.44/70.26**	**68.59/80.58**	**67.03/78.89**	**38.37/60.38**

Experimental Results

- Varying the number of target domain examples
- 20% ~ 30% parameters are satisfactory





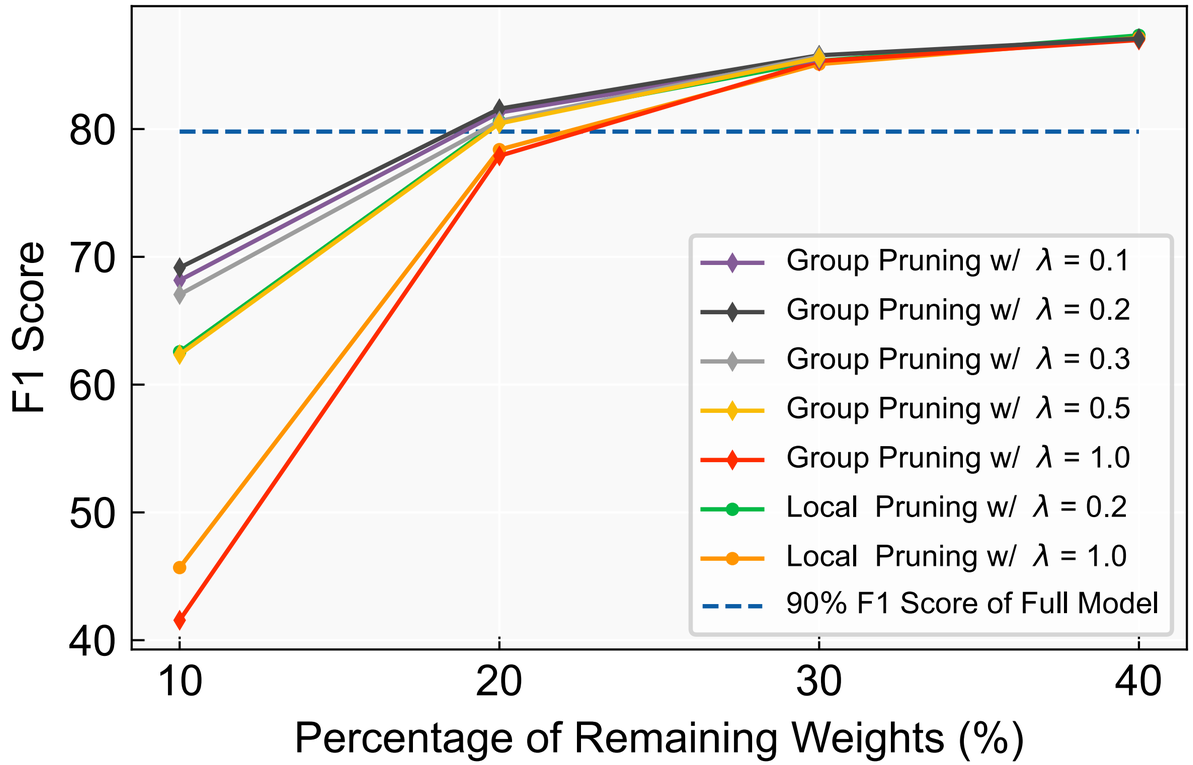
Analysis

Analysis

- Does structure-aware pruning deliver better lottery subnetworks?

Analysis

- Does structure-aware pruning deliver better lottery subnetworks?
  - Performance on source domain by the end of pruning



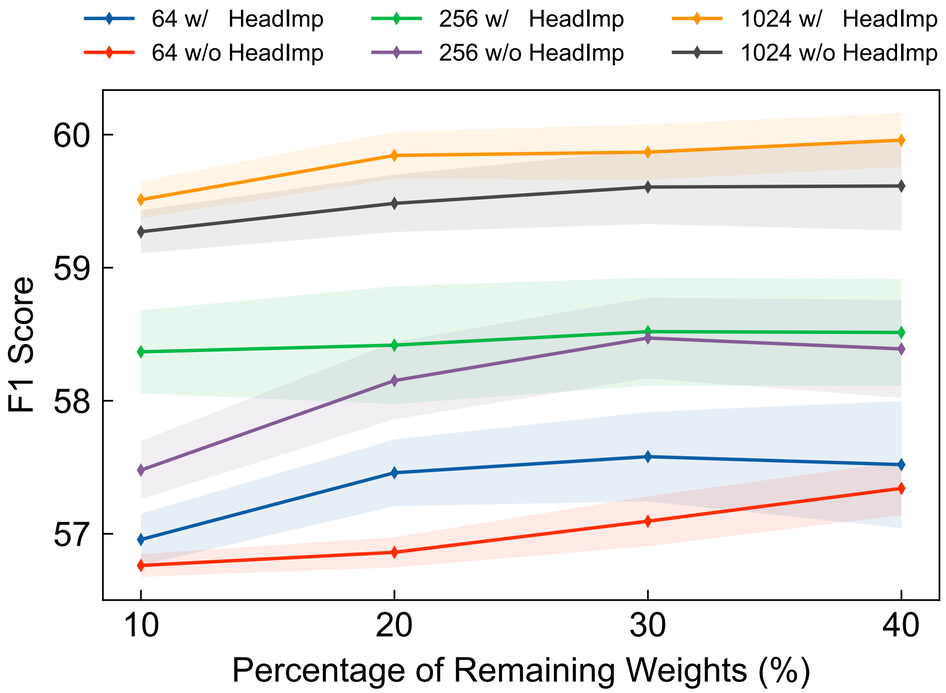
Analysis

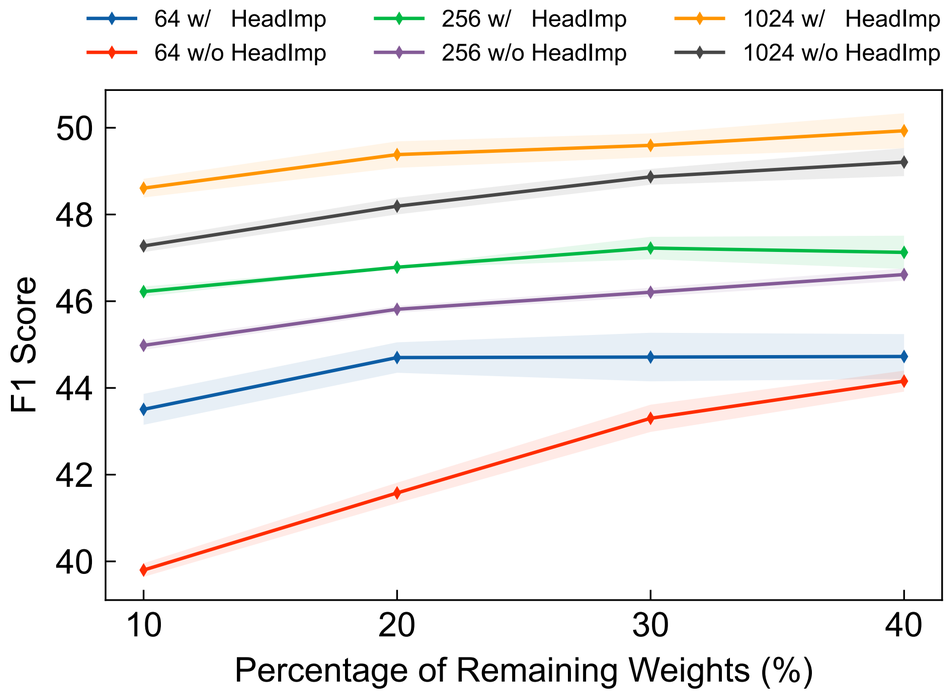
Analysis

- Do better lottery subnetworks improve domain adaptation performance?

Analysis

- Do better lottery subnetworks improve domain adaptation performance?
  - Adaptation w/ and w/o attention head importance on NewsQA and QuAC





Analysis

- What about other alternatives to lottery networks identification?

Analysis

- What about other alternatives to lottery networks identification?

Random

chooses parameters to constitute subnetworks randomly

Analysis

- What about other alternatives to lottery networks identification?

Random

chooses parameters to constitute subnetworks randomly

Magnitude

selects the highest magnitudes parameters in one-shot

Analysis

- What about other alternatives to lottery networks identification?

Random
chooses parameters to constitute subnetworks randomly

Magnitude
selects the highest magnitudes parameters in one-shot

Salvage
reuses the pruned redundant parameters, which operates conversely with our method

Analysis

- What about other alternatives to lottery networks identification?

Random
chooses parameters to constitute subnetworks randomly

Magnitude
selects the highest magnitudes parameters in one-shot

Salvage
reuses the pruned redundant parameters, which operates conversely with our method

AttrHead
prunes the whole attention head with structured pruning, and applies magnitude pruning in feed-forward layers

Analysis

- What about other alternatives to lottery networks identification?

Random
chooses parameters to constitute subnetworks randomly

Magnitude
selects the highest magnitudes parameters in one-shot

Salvage
reuses the pruned redundant parameters, which operates conversely with our method

AttrHead
prunes the whole attention head with structured pruning, and applies magnitude pruning in feed-forward layers

Method	NewsQA	TriviaQA	TweetQA
Fine-Tuning	40.59/57.40	53.13/60.39	68.23/79.93
Random	40.98/57.72	54.45/61.98	68.57/80.24
Magnitude	40.76/57.56	54.10/62.11	68.76/80.21
Salvage	40.86/57.67	54.39/61.75	68.82/80.24
AttrHead	41.31/58.08	54.35/61.80	68.88/**80.39**
Alter	**41.38/58.11**	**54.60/62.21**	68.89/80.35

Contributions

Contributions

- Propose Alter, a simple and effective few-shot domain adaptation paradigm for reading comprehension

Contributions

- Propose Alter, a simple and effective few-shot domain adaptation paradigm for reading comprehension
- Exploit a small fraction of parameters of the over-parameterized source domain model to adapt to the target domain

Contributions

- Propose Alter, a simple and effective few-shot domain adaptation paradigm for reading comprehension
- Exploit a small fraction of parameters of the over-parameterized source domain model to adapt to the target domain
- Introduce self-attention attribution to identify better subnetworks and improve the target domain performance

Contributions

- Propose Alter, a simple and effective few-shot domain adaptation paradigm for reading comprehension
- Exploit a small fraction of parameters of the over-parameterized source domain model to adapt to the target domain
- Introduce self-attention attribution to identify better subnetworks and improve the target domain performance
- Find subnetwork structures are critical to the effectiveness besides using fewer parameters

Thanks!
Q&A

The code is publicly available at *https://github. com/haichao592/ALTER.*