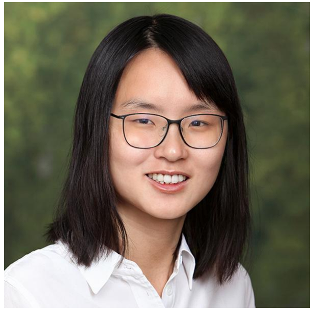
Structured Pruning Learns Compact and Accurate Models

ACL 2022



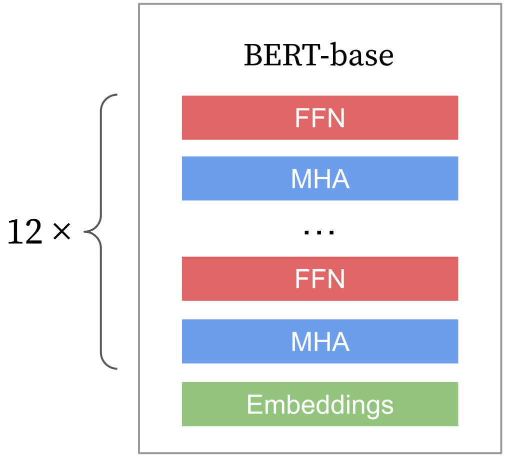
Mengzhou Xia

Zexuan Zhong

Danqi Chen

Background: Compressing Transformer Models

Language models are known to be **overparameterized**



84M parameters

1.0 × speedup

MNLI acc: 84.8

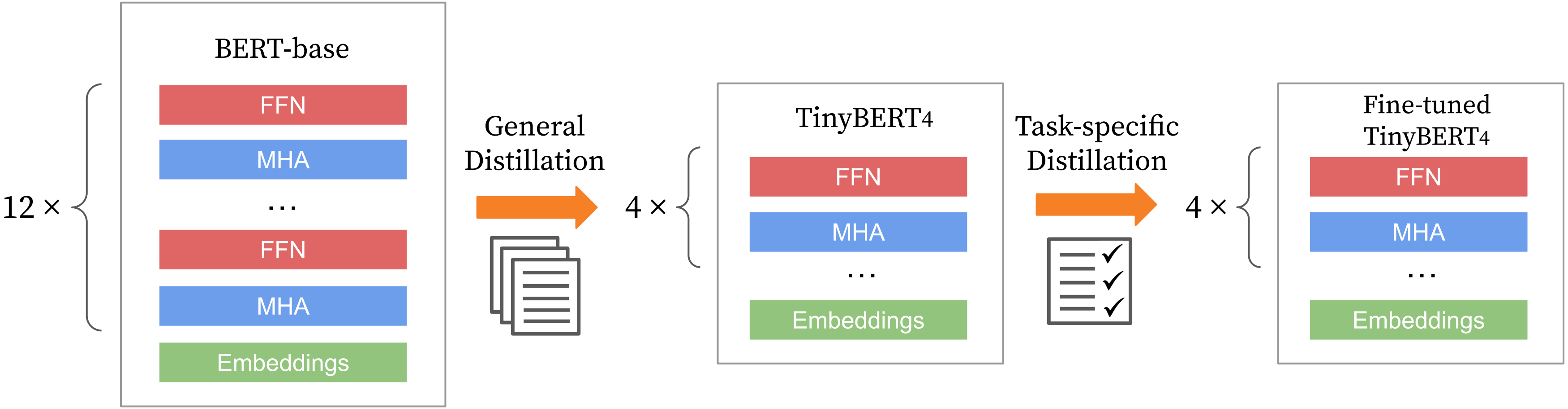
FFN: Feed-forward network

MHA: Multi-head attention

All parameter count excludes embeddings following previous works.

Background: Compressing Transformer Models

Distillation: transfers knowledge from a teacher model to a fixed student model



84M parameters
1.0 × speedup
MNLI acc: 84.8

4.7M parameters
11.4 × speedup
MNLI acc: 78.7

TinyBERT (Jiao et al. 2019), MNLI (Williams et al. 2017)

Background: Compressing Transformer Models

Knowledge distillation can:

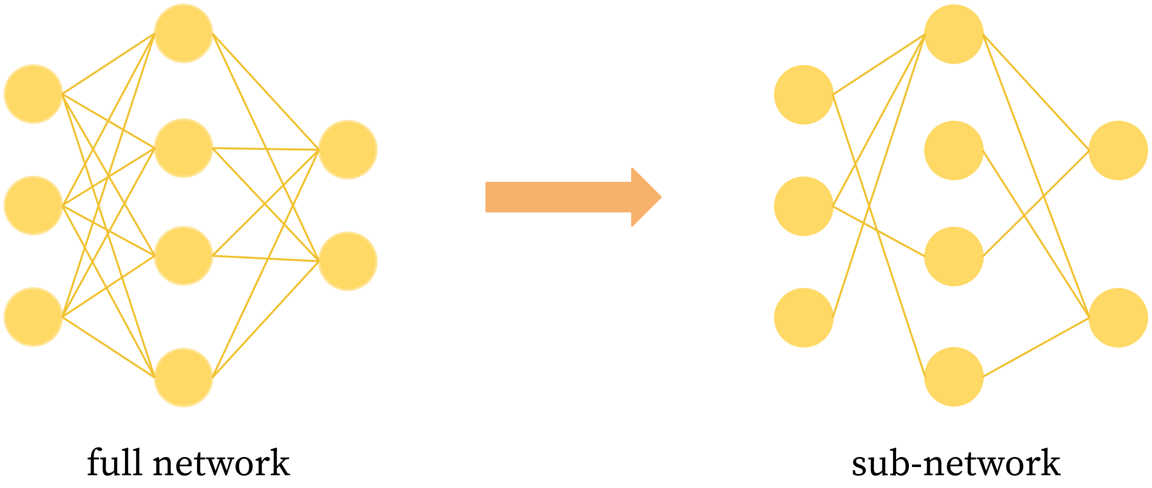
- achieve over 10× speedups

Disadvantages of knowledge distillation:

- Pre-specified student model architecture
- Trained from scratch with **unlabeled data** and computationally expensive e.g., 350 hours for TinyBERT

Background: Compressing Transformer Models

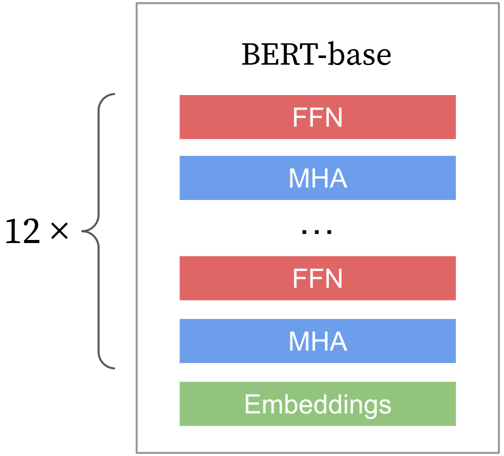
Unstructured pruning: Prunes individual parameters (Frankle and Carbin, 2019)



Hard to achieve inference speedup!

Background: Compressing Transformer Models

Structured pruning: Prunes groups of parameters, e.g., heads, FFNs, leads to actual speedups unlike unstructured pruning

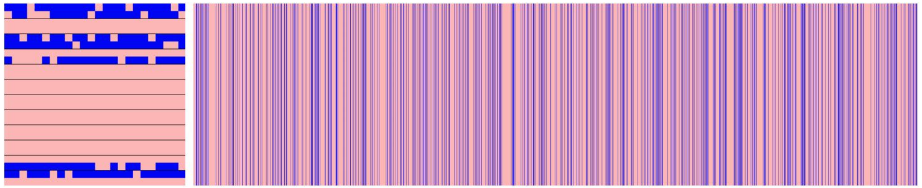


84M parameters

1.0 × speedup

MNLI acc: 84.8

Block Pruning (Lagunas et al. 2021)



25M parameters

2.7 × speedup

MNLI acc: 83.7

Background: Compressing Transformer Models

Why is structured pruning appealing:

- Flexible model structure with different sparsities
- Can achieve competitive results without unlabeled data
- Can be combined with task-specific distillation objectives

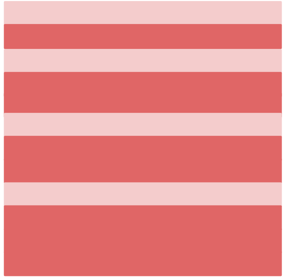
Hard to achieves a large speedup e.g., 10 ×, without a significant performance drop.

Motivation

Why can't structured pruning approaches achieve a large speedup?

$$\mathrm{MHA}(X) = \sum_{i=1}^{N_h} \mathbf{z}_{\mathrm{head}}^{(i)} \mathrm{Att}(W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, W_O^{(i)}, X)$$

Solution 1



4 heads are pruned in a layer

Solution 2

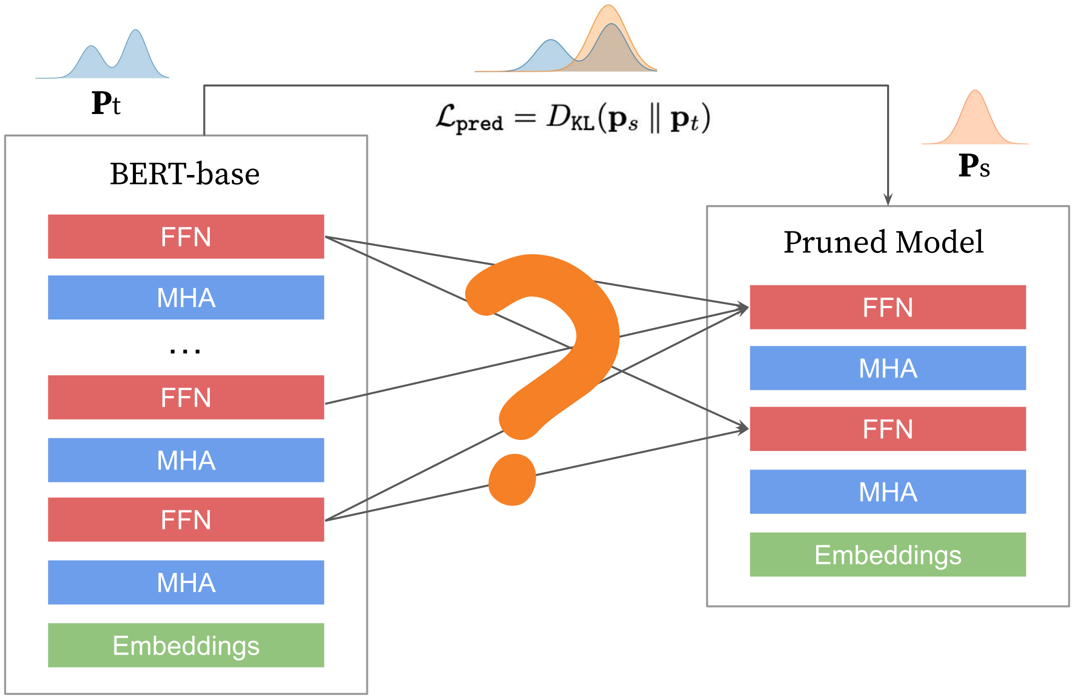
Difficult to optimize in practice

all heads are pruned in a layer

Pruning layers leads to significant speedup gains!

Motivation

Layer-wise distillation could possibly improve pruning performance



The prediction-layer loss is used in Sanh et al. 2020 and Lagunas et al. 2021

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

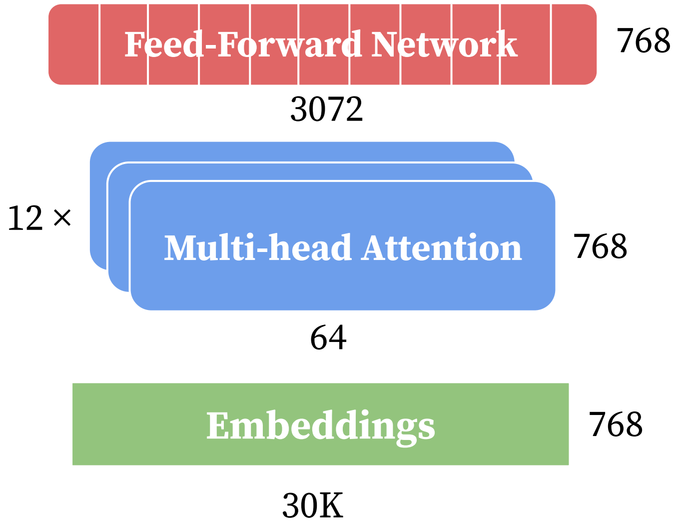
- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

Achieves **10×** speedup

Preserves **90%** accuracy

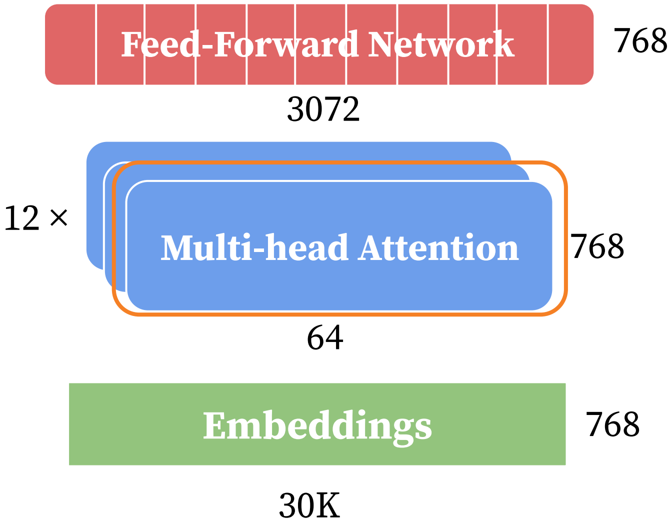
☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective



☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

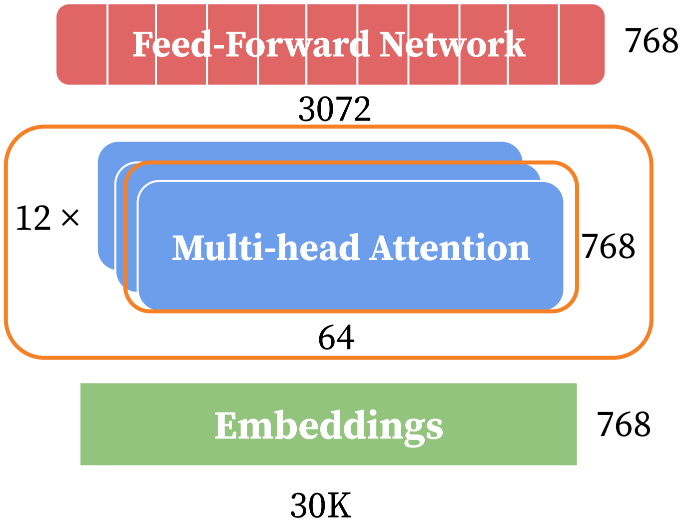


Fine-grained units:

- Heads

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective



Fine-grained units:

- Heads

Coarse units:

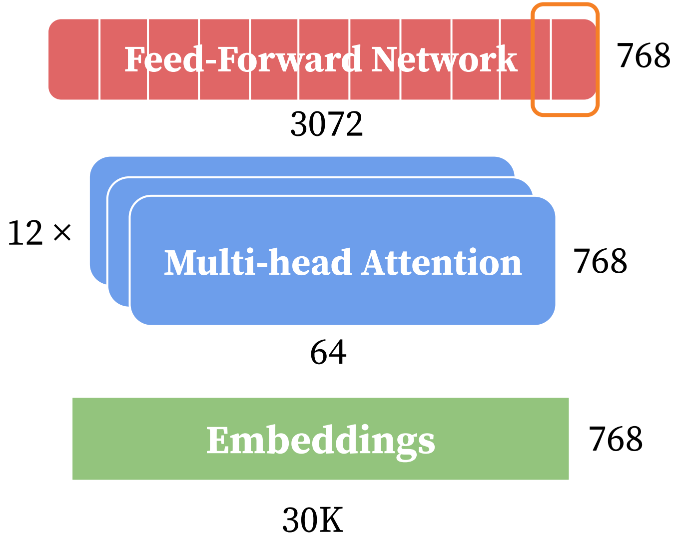
- MHA layers

$$\mathrm{MHA}(X) = z_{\mathrm{MHA}} \cdot \sum_{i=1}^{N_h} (\mathbf{z}_{\mathrm{head}}^{(i)} \cdot \mathrm{Att}(W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, W_O^{(i)}, X))$$

$$z \in \{0, 1\}$$

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective



Fine-grained units:

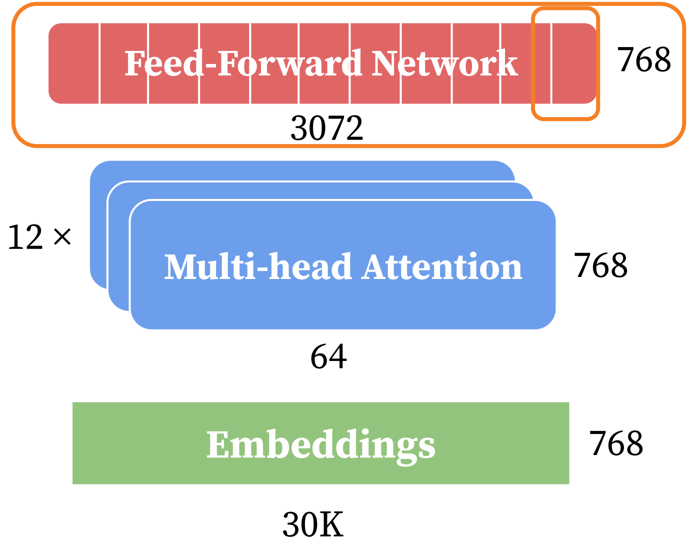
- Heads
- Intermediate dimensions

Coarse units:

- MHA layers

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective



Fine-grained units:

- Heads
- Intermediate dimensions

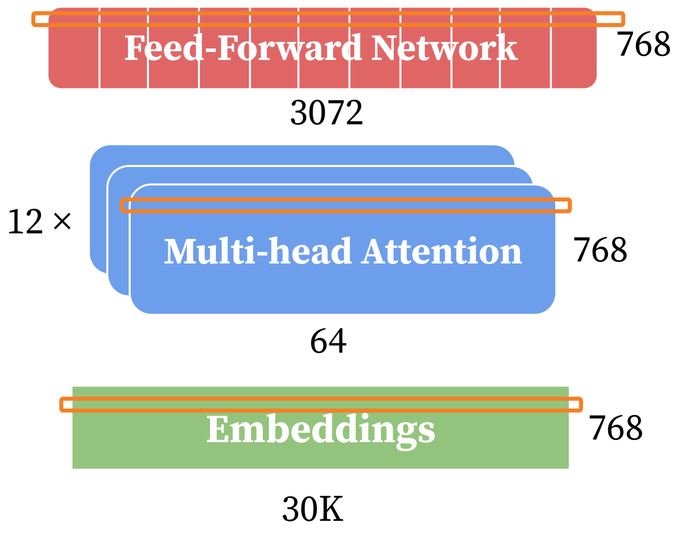
Coarse units:

- MHA layers
- FFN layers

$$\mathrm{FFN}(X) = z_{\mathrm{FFN}} \cdot \mathrm{gelu}(XW_U) \cdot \mathrm{diag}(\mathbf{z}_{\mathrm{int}}) \cdot W_D$$

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective



Fine-grained units:

- Heads
- Intermediate dimensions
- Hidden dimension

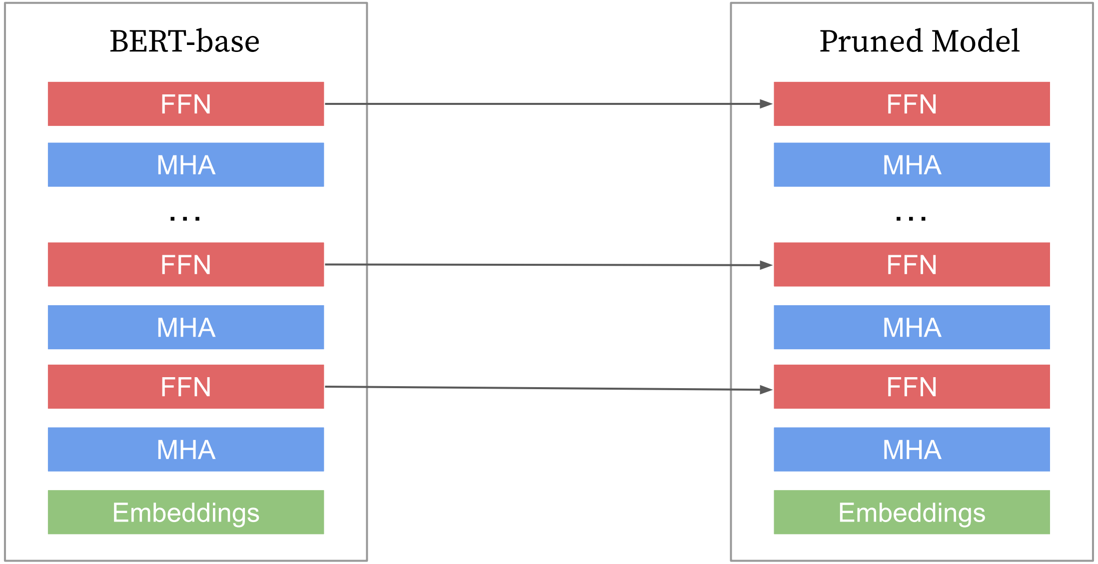
Coarse units:

- MHA layers
- FFN layers

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

Naive Approach

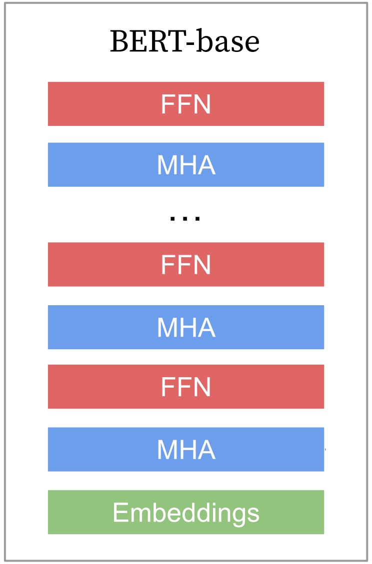


Suboptimal when upper layers are pruned.

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

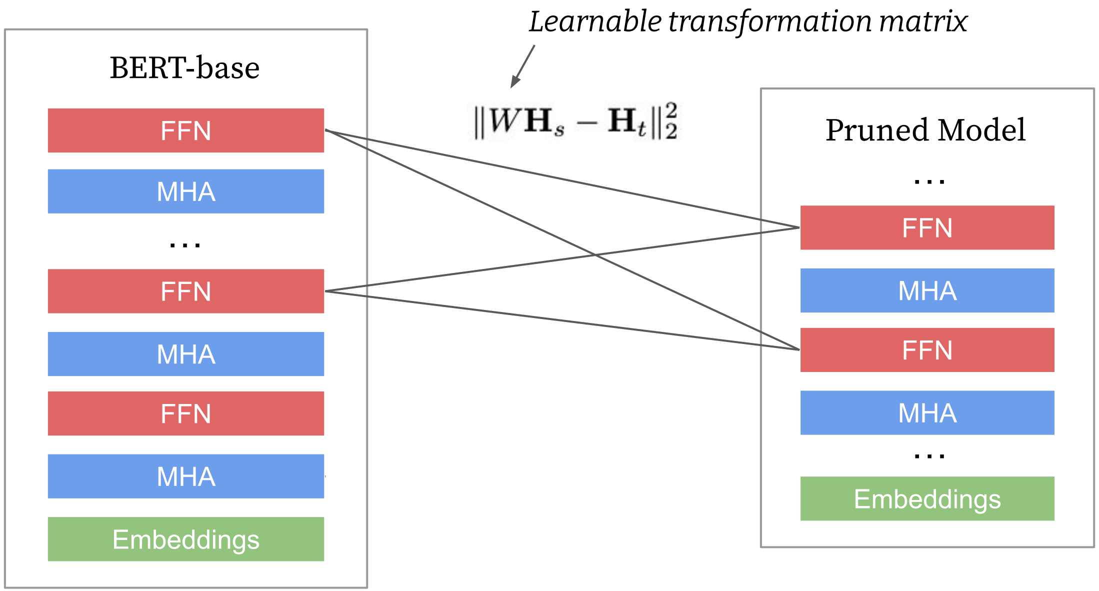
Select **4** teacher layers $\mathcal{T}$, following Jiao et al. 2020



☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

Calculate the **L2 distance** using the **training batch**

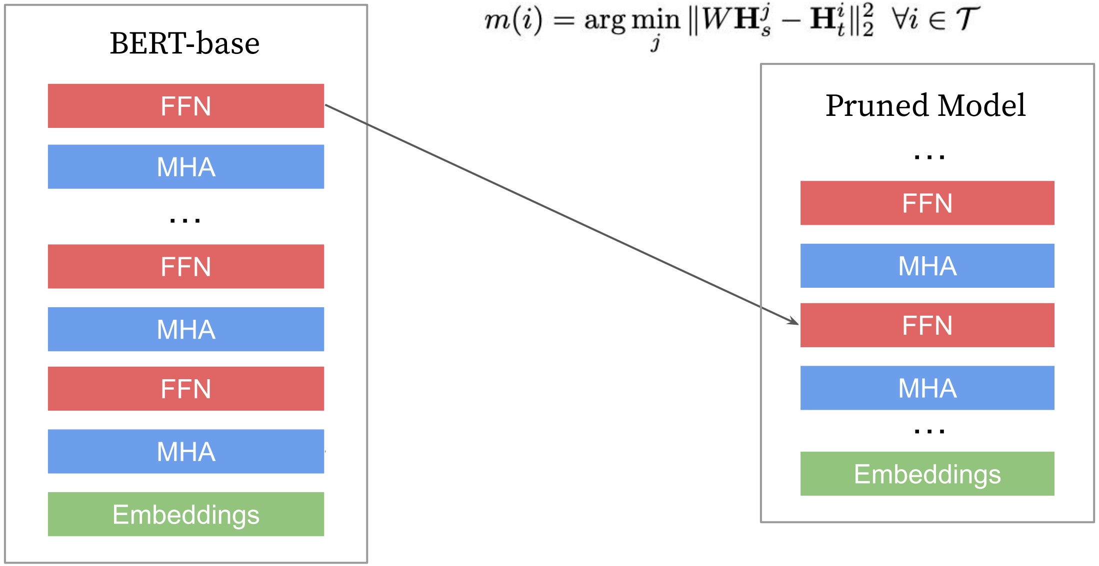


☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

A **low** L2 distance ⟶ A **good** approximation

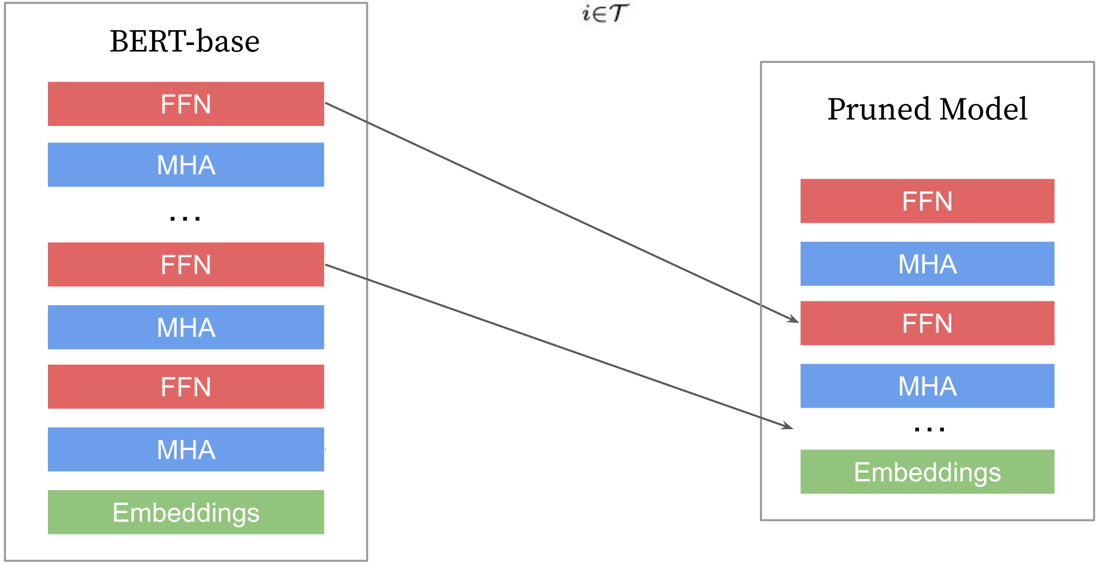
$$m(i) = \arg\min_{j} \|W\mathbf{H}_s^j - \mathbf{H}_t^i\|_2^2 \quad \forall i \in \mathcal{T}$$



☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

- Jointly prune coarse- and fine-grained units
- A layerwise distillation objective

$$\mathcal{L}_{\texttt{layer}} = \sum_{i \in \mathcal{T}} \| W \mathbf{H}_s^{m(i)} - \mathbf{H}_t^i \|_2^2$$



☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

How to control the sparsity of the final model?

Adapted the Lagrangian loss from Wang et al. 2020.

- How to model z:
  - hard-concrete distribution (Loizos et al. 2018)

- Expected sparsity:

f: function to calculate the sparsity

M: full model Size

$$\hat{s} = f(\mathbf{z}_{\text{int}}, \mathbf{z}_{\text{head}}, \mathbf{z}_{\text{hidden}}, z_{\text{FFN}}, z_{\text{MHA}}, M)$$

- Lagrangian loss:

s*: target sparsity

$$\mathcal{L}_{\text{lag}}(\lambda_1, \lambda_2, \hat{s}) = \lambda_1(s^* - \hat{s}) + \lambda_2(s^* - \hat{s})^2$$

☕ CoFi: **Co**arse- and **Fi**ne-grained Pruning

Final Objective

$$\max_{\lambda_1,\lambda_2} \min_{\theta,\hat{s}} \underbrace{\lambda \mathcal{L}_{\texttt{pred}}(\theta)}_{\text{Prediction layer distillation loss}} + \underbrace{(1-\lambda)\mathcal{L}_{\texttt{layer}}(\theta)}_{\text{Layerwise distillation loss}} + \lambda_1(s^* - \hat{s}) + \lambda_2(s^* - \hat{s})^2$$

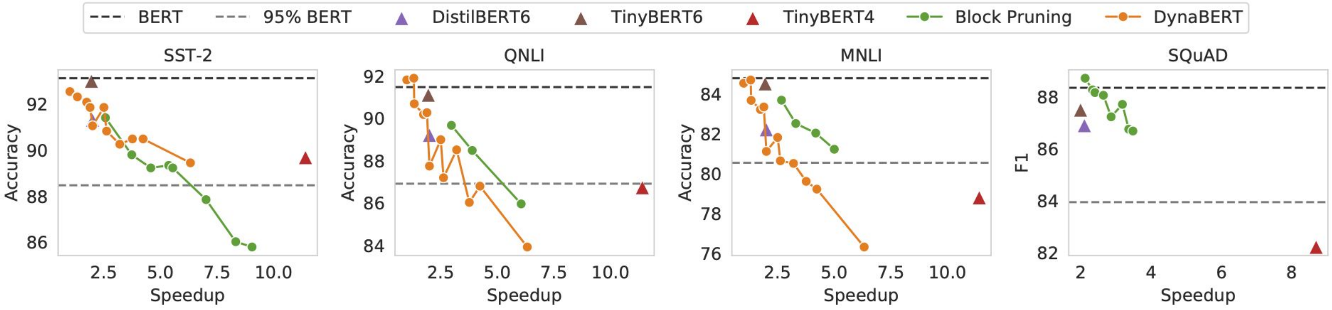
Target sparsity Expected sparsity

Experiment Results - GLUE and SQuAD 1.1

- GLUE: sentence classification tasks
- SQuAD 1.1: extractive question answering task

GLUE: Wang et al. 2019, SQuAD: Rajpurkar et al. 2016

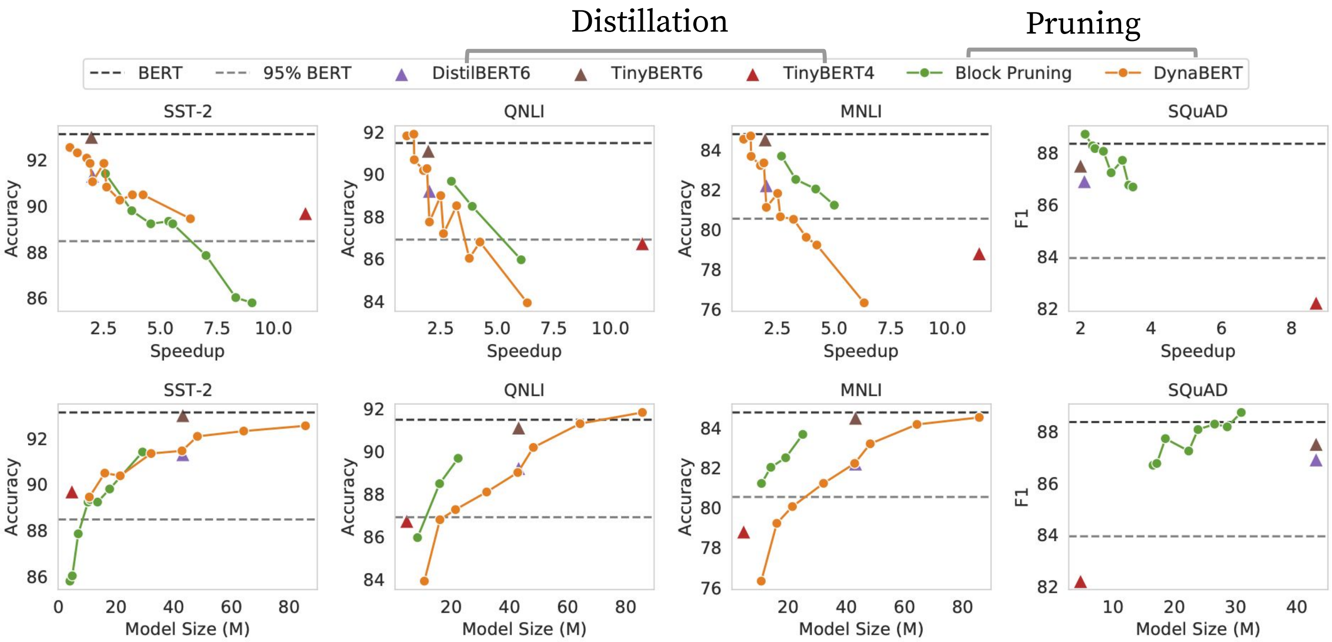
Baseline Results - GLUE and SQuAD 1.1



Speedup v.s. Performance

Please find more results on RoBERTa models and other compression approaches in the paper

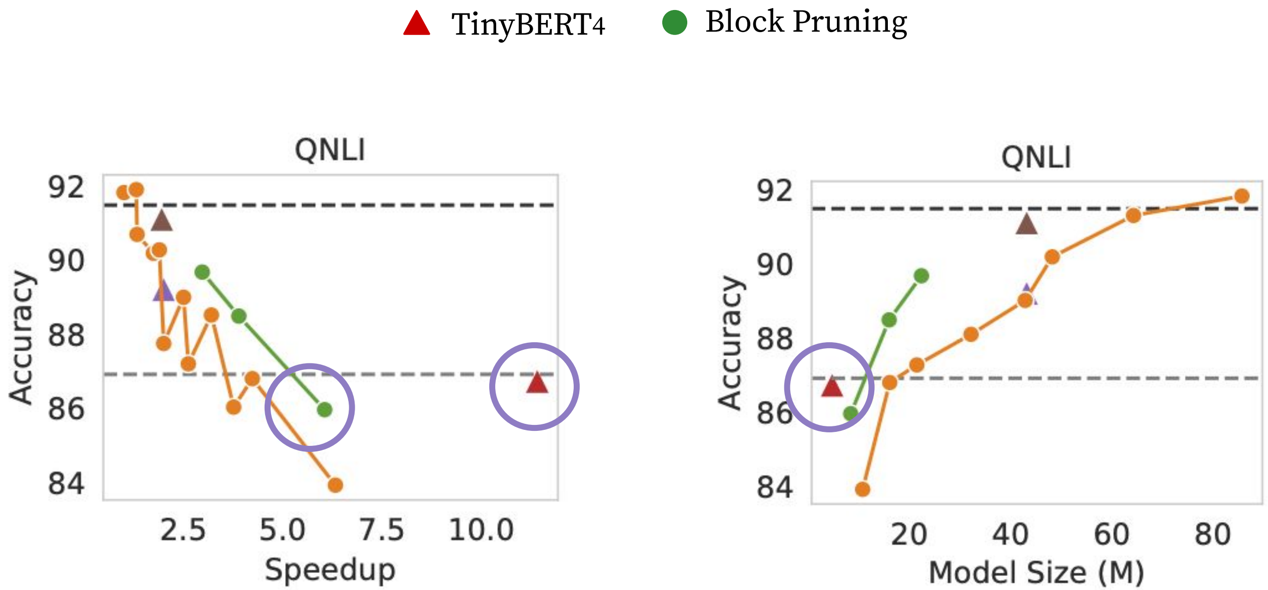
Baseline Results - GLUE and SQuAD 1.1



Speedup v.s. Performance and Model-size v.s. Performance

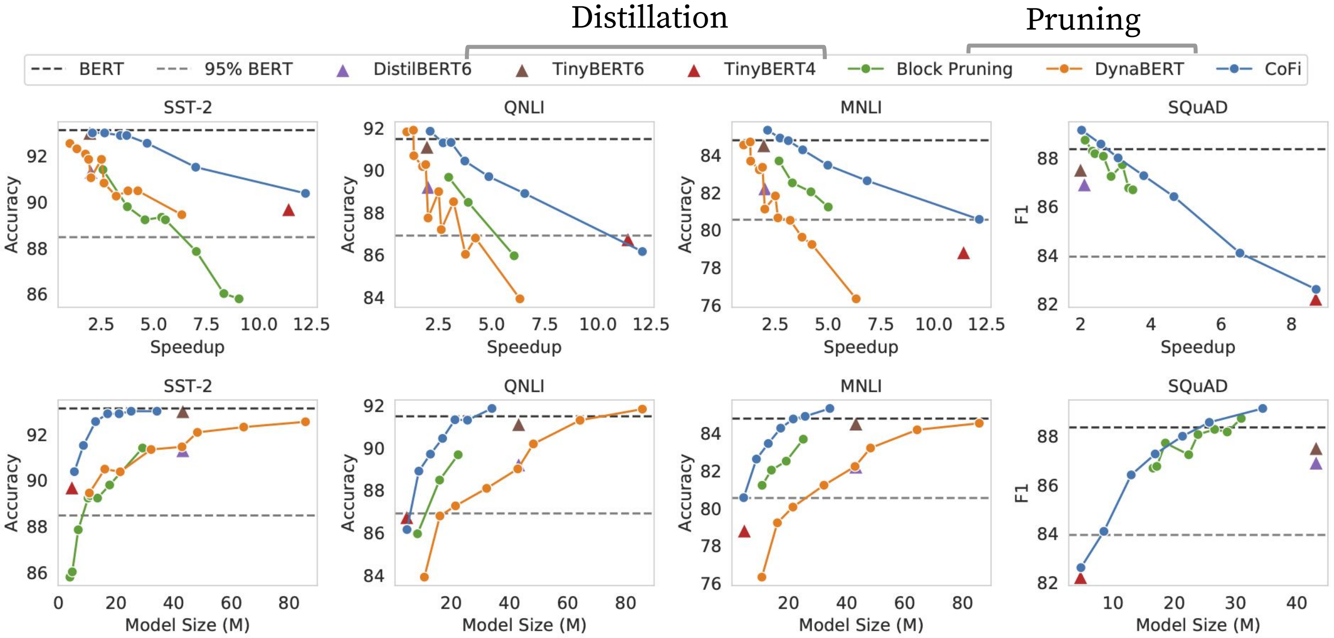
Please find more results on RoBERTa models and other compression approaches in the paper

Baseline Results - GLUE and SQuAD 1.1



Pruning falls behind distillation approaches on high sparsity levels.

Experiment Results - GLUE and SQuAD 1.1



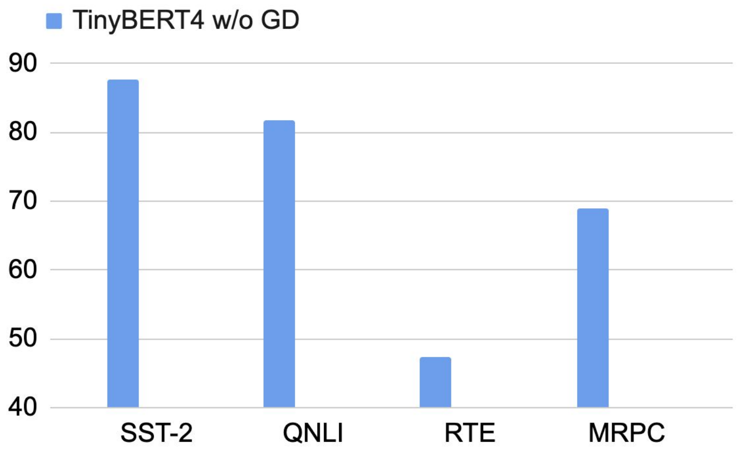
CoFi Pruning outperforms all distillation and pruning baselines **comparing under the same speedup and model size**

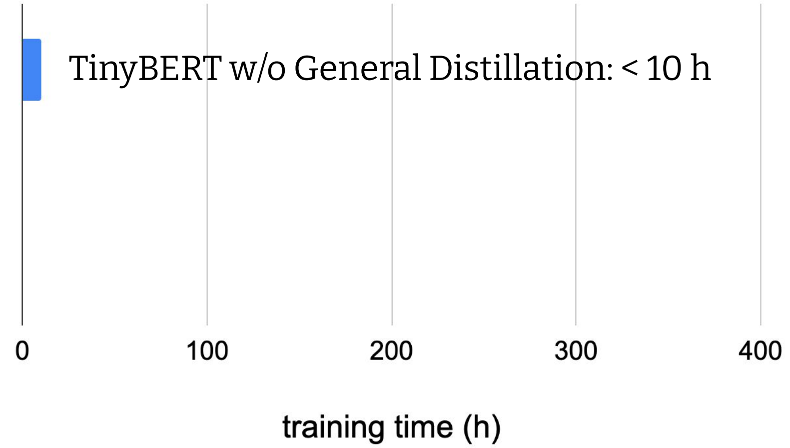
Experiment Results

Models with $10\times$ speedups

Experiment Results

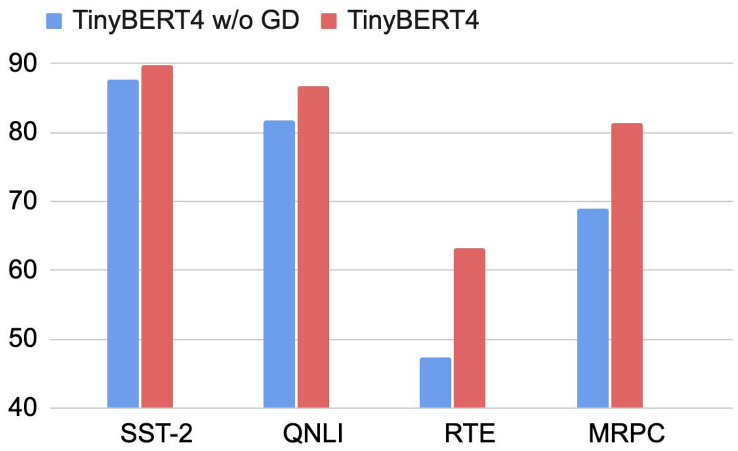
Models with 10 × speedups

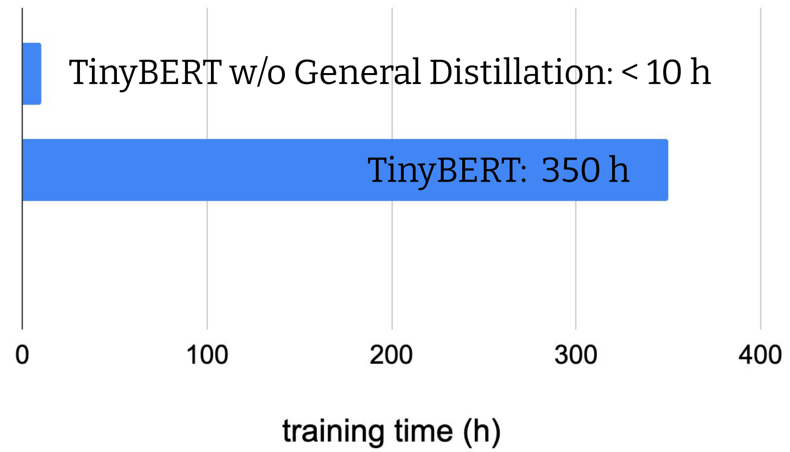




Experiment Results

Models with 10 × speedups

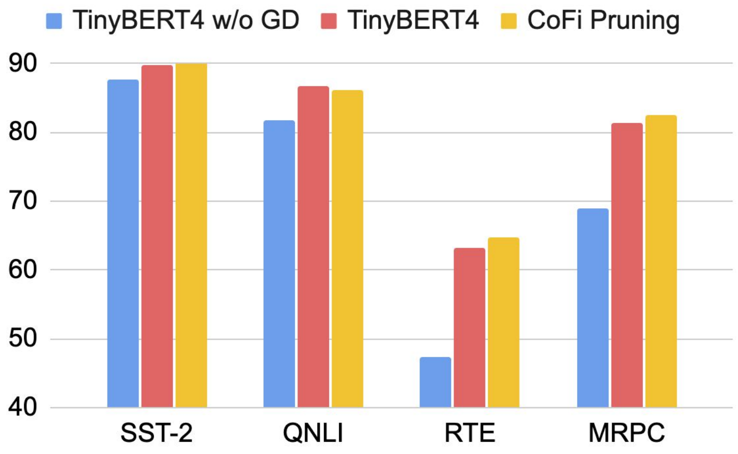


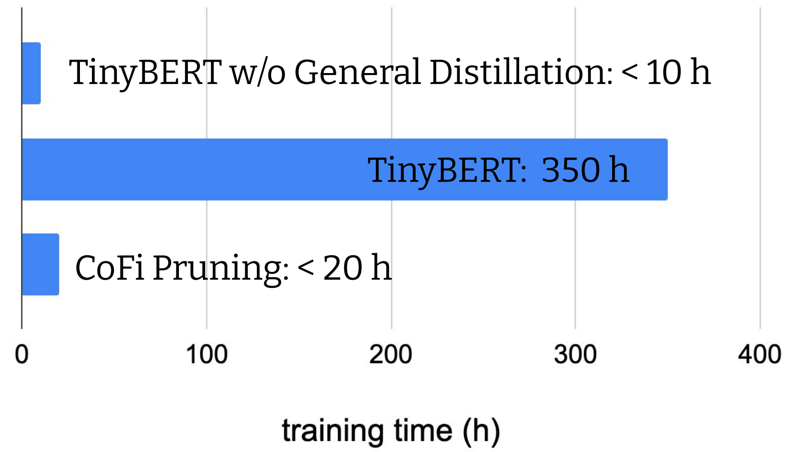


General distillation is essential but time consuming!

Experiment Results

Models with 10 × speedups





CoFi achieves **comparable or better performance** and speedup with **much less computation time**

*We also consider **task-specific data augmentation used in (Jiao et al., 2020)** and please find results in our paper.*

Ablation - Distillation Loss on 95% Models

	SST-2	QNLI	MNLI	SQuAD	Avg.
No distillation	86.6	84.2	78.2	75.8	81.2

Ablation - Distillation Loss on 95% Models

	SST-2	QNLI	MNLI	SQuAD	Avg.
No distillation	86.6	84.2	78.2	75.8	81.2
$+\mathcal{L}_{\texttt{pred}}$	**91.1**	85.1	79.7	82.5	84.6

Prediction layer distillation **brings a large gain**

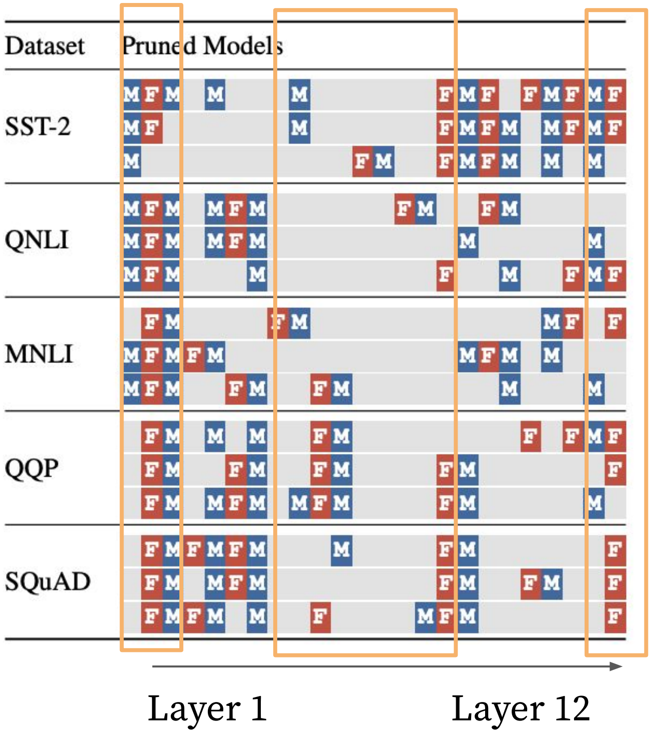
Prediction layer distillation are used in all our baselines.

Ablation - Distillation Loss on 95% Models

	SST-2	QNLI	MNLI	SQuAD	Avg.
No distillation	86.6	84.2	78.2	75.8	81.2
$+\mathcal{L}_{\texttt{pred}}$	**91.1**	85.1	79.7	82.5	84.6
$+\mathcal{L}_{\texttt{layer}}, \mathcal{L}_{\texttt{pred}}$ ☕	90.6	**86.1**	**80.6**	**82.6**	**85.0**

- Our proposed layer distillation loss brings **additional gains**
- The improvements on smaller sparsities are much larger

Model Structures with 95% Sparsity

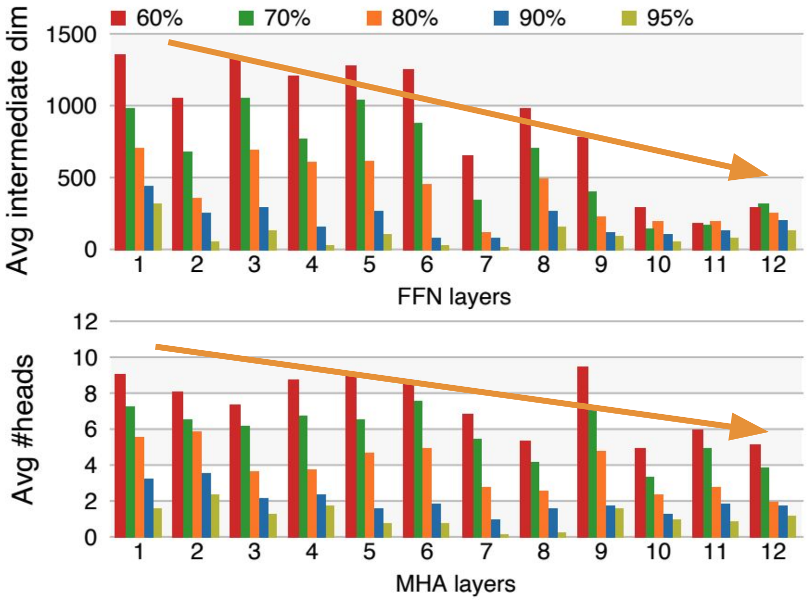


M: MHA layer, F: FFN layer

Coarse-grained units:

- First and last FFN layers are largely retained
- Middle layers are more likely to be pruned

Model Structures with 95% Sparsity



Fine-grained Units:

Heads and intermediate dimensions from the top-layers are more likely to be pruned

Summary

CoFi Pruning

- Jointly prune coarse- and fine-grained units
- An additional layerwise distillation loss to guide pruning

Compressed models

- Over 10 × speedups while maintaining 90% accuracy
- Closes the gap between structured pruning and knowledge distillation with much less computation

Q & A

Codebase: https://github.com/princeton-nlp/CoFiPruning

Contact: mengzhou@princeton.edu