Gradient-Free Structured Pruning with Unlabeled Data


Azade Nova


Hanjun Dai


Dale Schuurmans

Google DeepMind

THE AI CONFERENCE

Agenda

Motivation

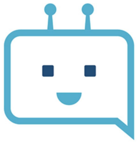
LLMs have achieved great success in solving difficult tasks across many domains.

- Cost of high parameter counts
- Significant computational overhead
- Inference latency

Conversational AI

Text generation

Text Classification



Text Summarization

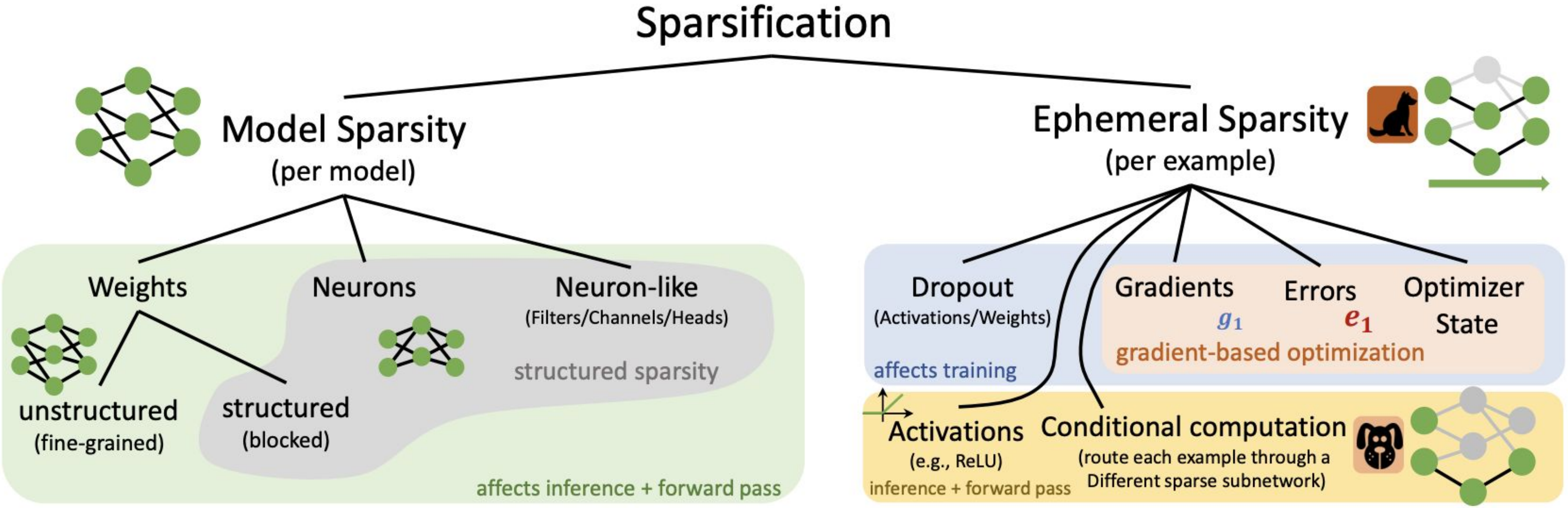
Question Answering

Dialogue Management

Overview: Compressing and Optimizing Models

- Downsizing models
  - creates smaller dense networks, e.g through distillation or neural architecture search
- Operator factorization
  - Represent/approximate matrices by factored decomposition $W_{[nxm]} = A_{[nxr]}B_{[rxm]}$
- Value quantization
  - Low precision value representation- weights, activation, etc
- Value Compression
  - Compress values, e.g., entropy-based (e.g., Huffman) or correlation based (e.g., gzip).
- Parameter sharing
  - Reuse parameters across neurons, e.g., Shapeshifter networks or CNNs
- Sparsification / pruning
  - Reduce the representational complexity using only a subset of the dimensions at a time

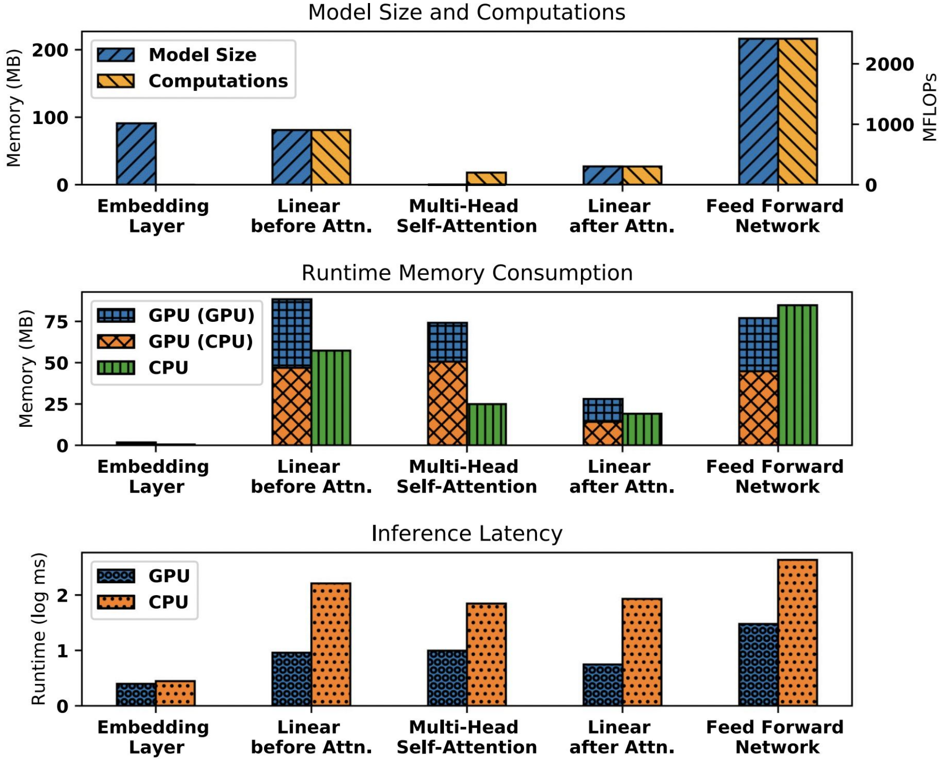
Overview: Sparsity

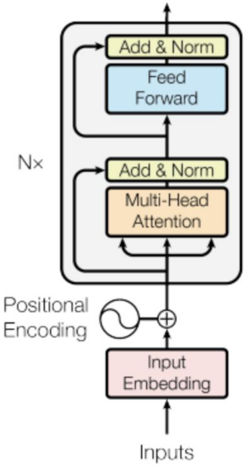


Existing approaches are complex, require a lot of labeled data, and require significant engineering effort to implement.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, Alexandra Peste, Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks

Breakdown Analysis of $BERT_{BASE}$



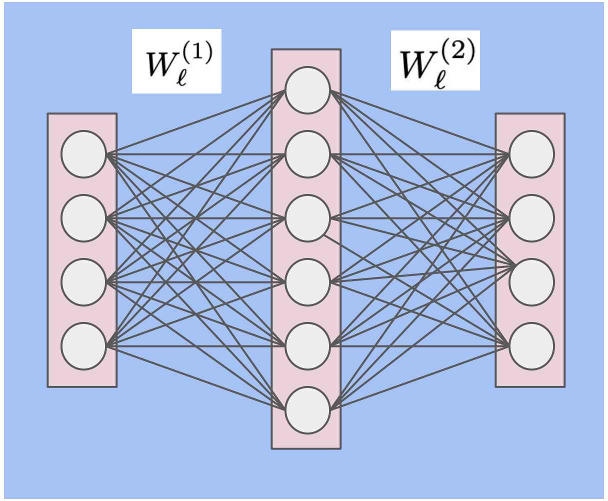


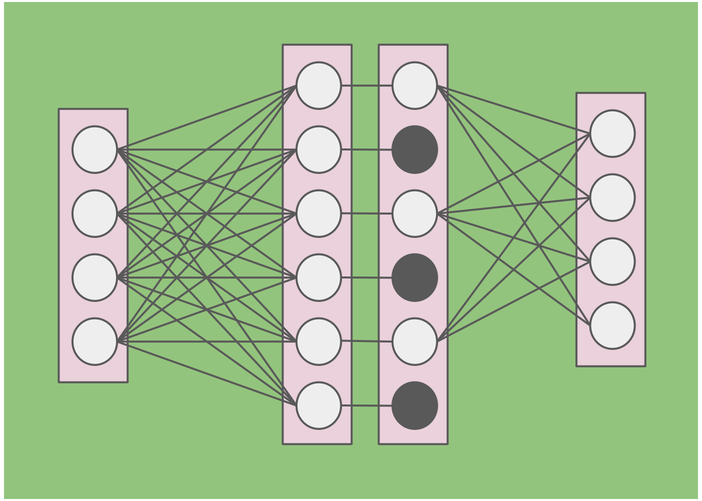
FFN sub-units are the parts consuming the most memory in terms of model size and executing the highest number of FLOPs

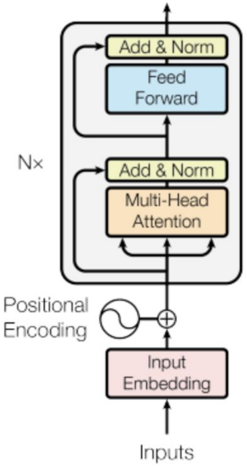
Google DeepMind

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, Marianne Winslett, Compressing Large-Scale Transformer-Based Models: A Case Study on BERT, Transactions of the Association for Computational Linguistics 2021

Structured Pruning by Masking







$$\widehat{FFN}_\ell(x) = \sum_{i=1}^{N} (\sigma(xW_\ell^{(1)}[:, i] + b_\ell^{(1)})W_\ell^{(2)}[i, :] \circ m_i) + b_\ell^{(2)}$$

$$\underset{\mathcal{M}}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{M}) \quad s.t. \quad Cost(\mathcal{M}) \leq \mathcal{C}$$

supervised setting with respect to minimizing the accuracy loss of the original model

3072 filters in $\text{BERT}_{\text{BASE}}$

$W_\ell^{(1)} \in \mathbb{R}^{768\times3072}$

$W_\ell^{(2)} \in \mathbb{R}^{3072\times768}$

Google DeepMind

Baselines from Structured Pruning Methods

Method	Gradient-free (!∇)	Retrain/Finetune-free	Supervision-free	Pruning time $\leq 7min$
FLOP (Wang et al., 2019)	✗	✗	✗	✗
SLIP (Lin et al., 2020)	✗	✗	✗	✗
Sajjad et al. (Sajjad et al., 2023)	✗	✗	✗	✗
DynaBERT (Hou et al., 2020)	✗	✗	✗	✗
EBERT (Liu et al., 2021b)	✗	✗	✗	✗
Mask-Tuning (Kwon et al., 2022)	✗	✓	✗	✓
Weight-Magnitude (Li et al., 2016)	✓	✓	N/A	✓
Weight-Magnitude-Scale	✓	✓	✓	✓
KCM (ours)	✓	✓	✓	✓

$$\widehat{FFN}_\ell(x) = \sum_{i=1}^{N}(\sigma(xW_\ell^{(1)}[:,i] + b_\ell^{(1)})W_\ell^{(2)}[i,:] \circ m_i) + b_\ell^{(2)}$$

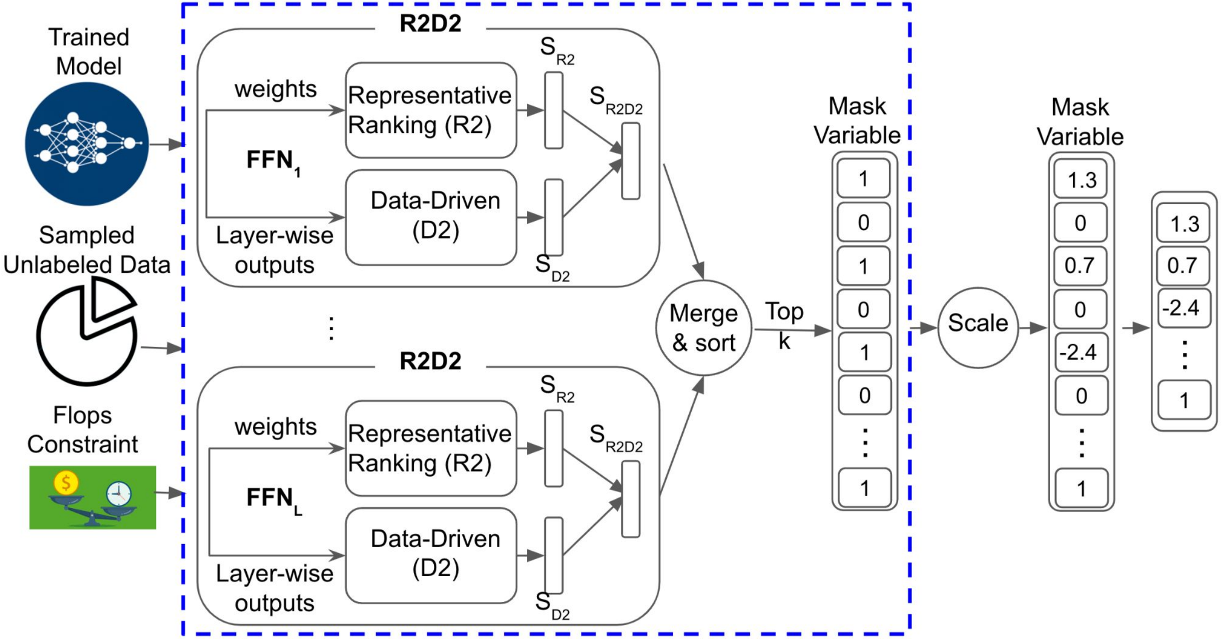
$$\underset{\mathcal{M}}{\operatorname{argmin}}\, \mathcal{L}_{FMT}(\mathcal{M}) \quad s.t. \quad Cost(\mathcal{M}) \leq \mathcal{C}$$

unsupervised setting

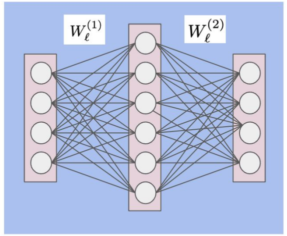
$$\mathcal{L}_{(\ell)\,FMT}(m) = \|FFN_{(\ell)}(x) - \widehat{FFN}_{(\ell)}(x)\|_2$$

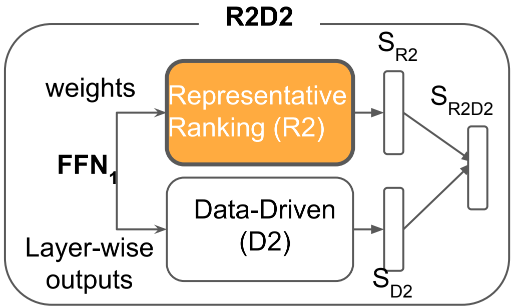
Kernelized Convex Masking (KCM)

1. R2D2
   a. Representative Ranking (R2)
   b. Data-Driven (D2)
2. Merge and Sort
3. Scale



Representative Ranking (R2)





N points in d-dim

$$FFN_\ell(H_\ell^{(1)}) = H_\ell^{(1)} W_\ell^{(2)} + b_\ell^{(2)}$$

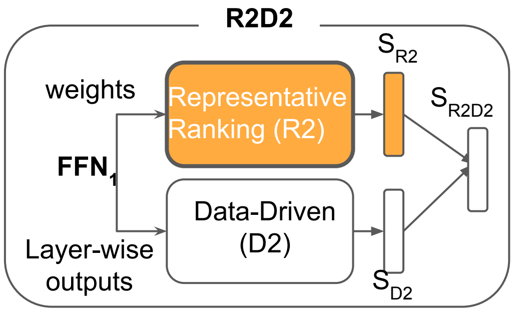
- Structured pruning goal : Select a subset of data points as representative.
- For linear functions, this problem can be reduced to finding a convex hull.
  - Complexity: $\mathcal{O}(N^{d/2})$
  - Number of convex hull points radically increases with d.

Kernelized Convex Hull Approximation[1]

Google DeepMind

[1] Huang, C., Wu, Y., Min, G., and Ying, Y. Kernelized convex hull approximation and its applications in data description tasks. In 2018 International Joint Conference on Neural Networks (IJCNN), 2018

Representative Ranking (R2)

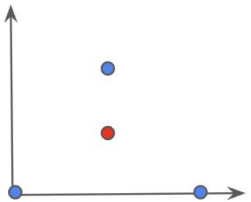


Find a positive coefficient matrix

$$C \in \mathbb{R}^{N\times N}$$

that minimizes

$$\|W_\ell^{(2)} - W_\ell^{(2)} C\|_2$$



$$\begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0.25 \\ 0 & 1 & 0 & 0.25 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{C}$$

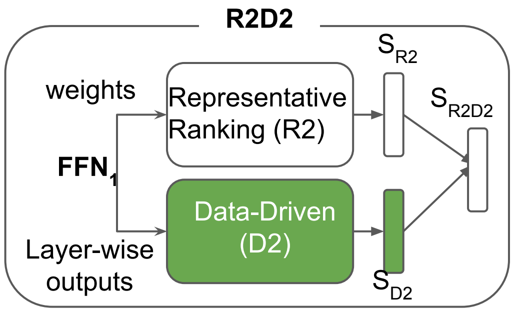
Algorithm 2 Representative Ranking (R2)

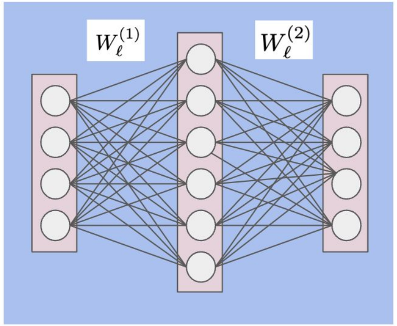
Input: Trained model: Model, Gaussian Kernel K with width σ, convergence rate α
Output: S_{R2} that represents importance of Filters in all layers.
for layer ℓ in layers of the Model **do**
    $W_\ell^{(2)} \in \mathbb{R}^{N\times d}$ of FFN_ℓ
    Initialize coefficient matrix: $C_0 \in \mathbb{R}^{N\times N} = \frac{1}{N}$
    repeat
        $C_{i+1} = C_i \circ \sqrt{\frac{K(W_\ell^{(2)}, W_\ell^{(2)})}{K(W_\ell^{(2)}, W_\ell^{(2)})C_i}}$
        $\delta = \frac{(C_{i+1} - C_i).sum()}{C_i.sum}$
        $C_i = C_{i+1}$
    until convergence i.e. $\delta \leq \alpha$
    $S_{R2}[\ell]$= diagonal(C_i)
end for
return S_{R2}

Data-Driven (D2)

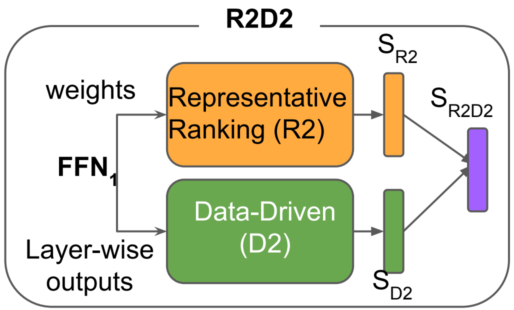
Algorithm 1 Kernelized Convex Masking (KCM)

```
Input: Trained model: Model, FLOPs constraint 𝒞, Gaussian Kernel K, convergence rate α
Output: Mask ℳ
Initialize mask ℳ as 0
//Call Representative Ranking (R2) Algorithm 2
S_R2 = R2(Model, K, α)
// Data-Driven (D2) Ranking
for batch in sample-data do
    for each layer ℓ in Model collect H_ℓ^(1)
    S_D2[ℓ] = average over H_ℓ^(1) for each filter
end for
S_R2D2[ℓ] = S_R2[ℓ] * normalized(S_D2[ℓ])
k = Number of neurons to satisfy FLOPs constraint 𝒞
Candidates = top-k filters of the sorted S_R2D2
ℳ[Candidates] = 1.0
return ℳ
```





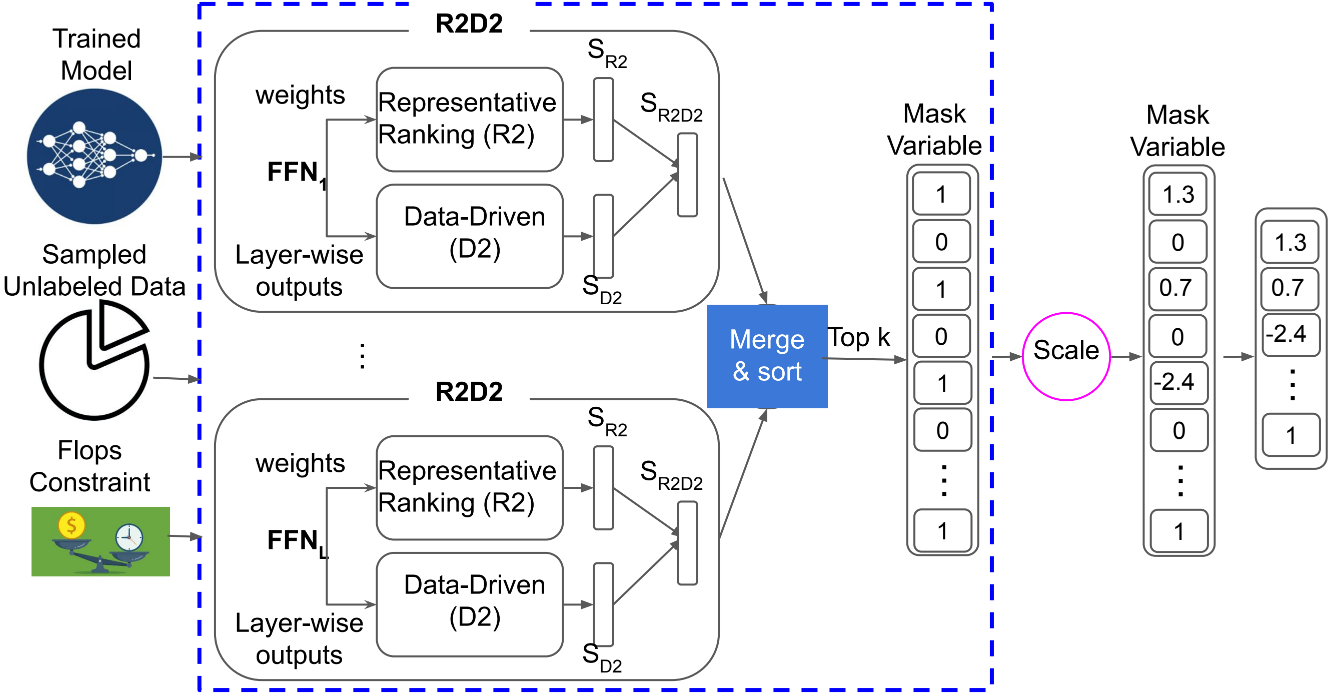
R2D2



Algorithm 1 Kernelized Convex Masking (KCM)

Input: Trained model: Model, FLOPs constraint $\mathcal{C}$, Gaussian Kernel K, convergence rate α
Output: Mask $\mathcal{M}$
Initialize mask $\mathcal{M}$ as **0**
//Call Representative Ranking (R2) *Algorithm 2*
$S_{R2} = R2(\text{Model}, K, \alpha)$
// Data-Driven (*D2*) Ranking
for batch in sample-data **do**
    for each layer ℓ in Model collect $H_\ell^{(1)}$
    $S_{D2}[\ell]$ = average over $H_\ell^{(1)}$ for each filter
end for
$\boxed{S_{R2D2}[\ell] = S_{R2}[\ell] * normalized(S_{D2}[\ell])}$
k = Number of neurons to satisfy FLOPs constraint $\mathcal{C}$
Candidates = top-k filters of the sorted S_{R2D2}
$\mathcal{M}[Candidates] = 1.0$
return $\mathcal{M}$

Merge and Scale

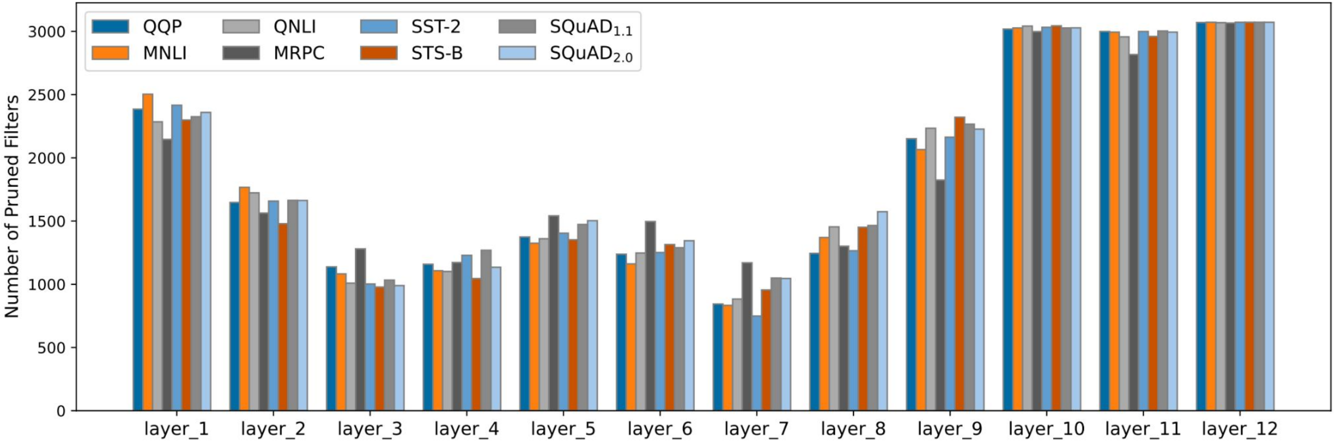


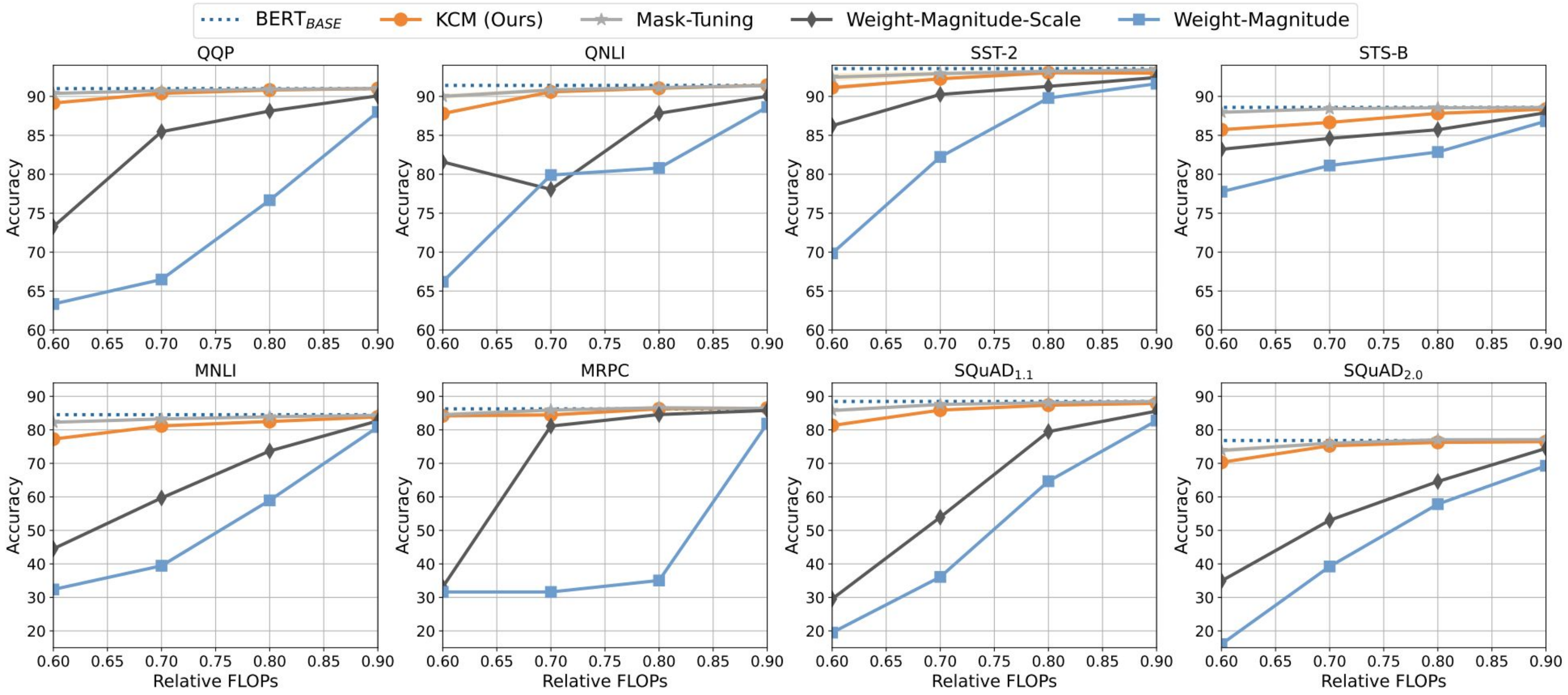
Experiments on $BERT_{BASE}$

!∇	Method	QQP 60%	QQP 70%	MNLI 60%	MNLI 70%	MRPC 60%	MRPC 70%	QNLI 60%	QNLI 70%
	baseline	91.00		84.53		86.27		91.41	
✗	Mask-Tuning	90.38 ± 0.07	90.74 ± 0.07	82.26 ± 0.21	83.24 ± 0.16	84.51 ± 0.63	85.91 ± 0.40	90.00 ± 0.26	90.83 ± 0.16
✓	**KCM (Ours)**	89.15 ± 0.04	90.39 ± 0.04	77.24 ± 0.10	81.18 ± 0.10	84.19 ± 0.44	84.46 ± 0.29	87.79 ± 0.15	90.58 ± 0.08

!∇	Method	SST-2 60%	SST-2 70%	STS-B 60%	STS-B 70%	$SQuAD_{1.1}$ 60%	$SQuAD_{1.1}$ 70%	$SQuAD_{2.0}$ 60%	$SQuAD_{2.0}$ 70%
	baseline	93.57		88.59		88.48		76.82	
✗	Mask-Tuning	92.47 ± 0.41	92.92 ± 0.26	87.95 ± 0.12	88.40 ± 0.05	85.77 ± 0.41	87.57 ± 0.11	73.86 ± 0.55	76.00 ± 0.29
✓	**KCM (Ours)**	91.11 ± 0.23	92.26 ± 0.09	85.72 ± 0.12	86.66 ± 0.05	81.29 ± 0.06	85.89 ± 0.04	70.30 ± 0.13	75.24 ± 0.10

Dynamic Pruning Size





BERT$_{BASE}$
KCM (Ours)
Mask-Tuning
Weight-Magnitude-Scale
Weight-Magnitude
QQP
QNLI
SST-2
STS-B
MNLI
MRPC
SQuAD$_{1.1}$
SQuAD$_{2.0}$
Accuracy
Relative FLOPs

+ Limited labeled data

- One forward-backward pass and gather the gradient over the mask variables
- Refine top-k results

$!\nabla$	Method	QQP		MNLI		MRPC		QNLI	
		60%	70%	60%	70%	60%	70%	60%	70%
	baseline	89.99		82.11		84.80		88.56	
✗	Mask-Tuning	88.71 ± 0.22	89.66 ± 0.06	80.51 ± 0.19	81.65 ± 0.09	84.73 ± 0.71	84.83 ± 0.35	87.72 ± 0.38	88.43 ± 0.07
✓	**KCM**	88.16 ± 0.03	89.28 ± 0.03	78.05 ± 0.08	80.60 ± 0.05	79.66 ± 0.27	83.01 ± 0.16	85.93 ± 0.09	86.93 ± 0.13
✗	**Extension(512 labeled data)**	88.76 ± 0.25	89.45 ± 0.07	80.02 ± 0.25	81.37 ± 0.11	83.70 ± 1.40	84.49 ± 0.49	87.21 ± 0.54	88.21 ± 0.15
✗	**Extension(1k labeled data)**	88.92 ± 0.20	89.53 ± 0.08	80.41 ± 0.11	81.50 ± 0.12	84.17 ± 0.45	84.68 ± 0.49	87.60 ± 0.31	88.29 ± 0.16

$!\nabla$	Method	SST-2		STS-B		$SQuAD_{1.1}$		$SQuad_{2.1}$	
		60%	70%	60%	70%	60%	70%	60%	70%
	baseline	91.40		86.12		85.73		68.84	
✗	Mask-Tuning	90.44 ± 0.41	90.93 ± 0.24	85.73 ± 0.07	85.96 ± 0.10	83.20 ± 0.16	84.64 ± 0.09	62.36 ± 1.40	65.32 ± 0.48
✓	**KCM**	88.38 ± 0.25	90.61 ± 0.25	85.26 ± 0.02	85.55 ± 0.03	76.92 ± 0.11	82.65 ± 0.06	64.56 ± 0.11	68.19 ± 0.06
✗	**Extension(512 labeled data)**	89.32 ± 0.54	90.38 ± 0.35	85.83 ± 0.11	86.02 ± 0.07	81.41 ± 0.26	83.30 ± 0.09	66.51 ± 0.24	67.72 ± 0.18
✗	**Extension(1k labeled data)**	89.86 ± 0.56	90.62 ± 0.40	85.90 ± 0.11	86.04 ± 0.06	81.16 ± 0.22	83.34 ± 0.11	66.35 ± 0.32	67.74 ± 0.18

Conclusion

- We studied the problem of structured pruning with unlabeled data and no backward pass.
- We proposed a gradient-free structured pruning framework that prunes the filters with the help of our proposed R2D2 that combines two ranking techniques called Representative Ranking (R2) and Data-Driven (D2).
- We empirically evaluated our framework on GLUE and SQuAD benchmarks using $\text{BERT}_{\text{BASE}}$ and DistilBERT. Compared to when the labeled data is available, our approach achieved up to 40% FLOPs reduction with less than 4% accuracy loss over all tasks considered.