Fire Together Wire Together: A Dynamic Pruning Approach with Self-Supervised Mask Prediction

Sara Elkerdawy[1], Mostafa Elhoushi[2], Hong Zhang[1], Nilanjan Ray[1]

1) Department of Computing Science, University of Alberta , 2) Toronto Heterogeneous Compilers Lab, Huawei

Acknowledgement

HUAWEI

compute canada | calcul canada

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

Motivation

Model	Accuracy	FLOPs
ResNet18	69.76%	1.8 billion
ResNet34	73.31%	3.6 billion
MobileNetv1	70.6%	569 million
MobileNetv1 0.75	68.4%	325 million
Inception v2	78.00%	7.0 billion
Inception v4	80.2	16.0 billion

***2x** FLOPs to gain **<4%** accuracy*

***1.75x** FLOPs to gain **<2%** accuracy*

***2.2x** FLOPs to gain **<2%** accuracy*

- Diminishing returns to adding more FLOPs. **Double** the computation for **≈2%** accuracy gain.
- Can we only enable the neurons required for each image sample?
- We propose a **dynamic inference** method to compute different sub-network based on the input samples. **Each layer is equipped by a decision gate to select few filters to apply per sample.**

Keywords: Dynamic Pruning, Conditional Inference, Efficient Neural Networks

Key Contributions

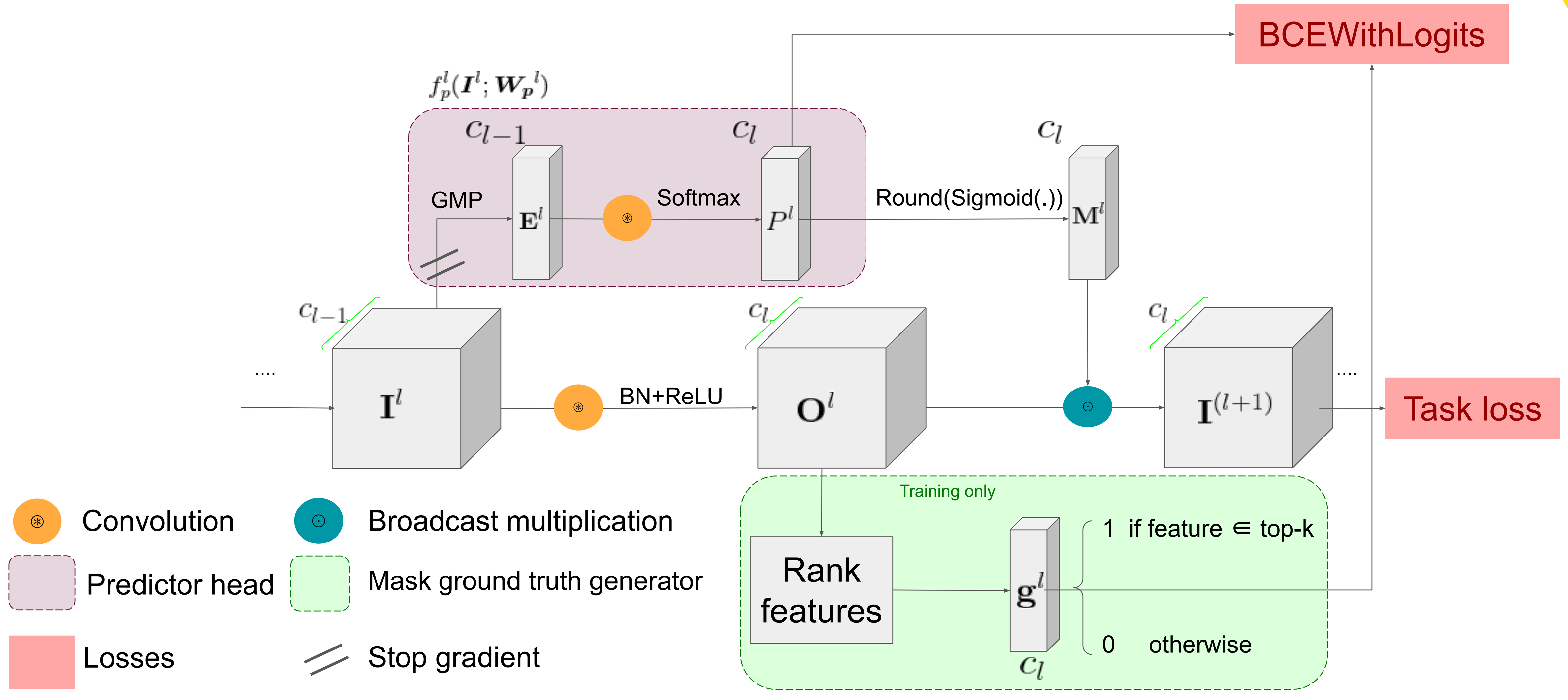
Typical dynamic inference training rely on regularization loss to learn the decision gating.

Regularization loss can be hard to tune as pruning ratio increases due to multi-loss (i.e task and regularization loss) gradient interference.

In this paper, we propose:

- A novel decision gating loss formulation with **self-supervised ground truth mask generation** that is stochastic gradient descent (SGD) friendly and **decoupled from task loss.**
- A novel **dynamic signature based on the heatmap mass** without a pre-defined pruning ratio per layer.

Proposed Method (FTWT)



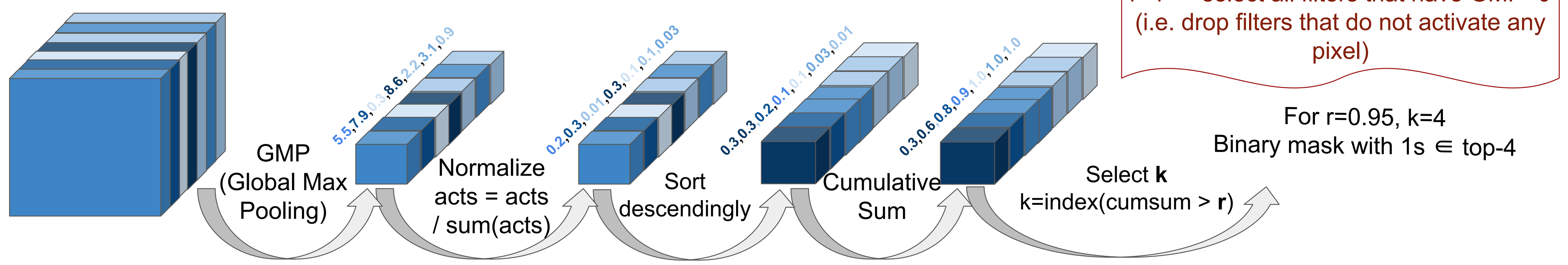
- **Self-Supervised Binary Gating**
  - Our proposed method learns the dynamic decision gating in a self-supervised way.
  - During training, we rank layer's output features and push the decision gating to predict the top-k highly activated features. Top-k is selected based on a hyperparameter **r**.
  - During inference, we use the binary prediction output from the learned decision gate to perform handful of filters from the layer based on the input.
- **Loss Function**

$$\min_{\{\boldsymbol{W}, \boldsymbol{W_p}\}} L_{total} = L_{ent}(f_n(\boldsymbol{x}; \boldsymbol{W}), \boldsymbol{y}_k) + L_{pred}(\{f_p^l(\boldsymbol{I}^l; \boldsymbol{W_p}^l), \boldsymbol{g}^l\}_L)$$

$$L_{pred}(\{\boldsymbol{P}^l, \boldsymbol{g}^l\}_L) = \sum_{l}^{L} \sum_{f}^{F_l} BCEWithLogits(\boldsymbol{P}^l_f, \boldsymbol{g}^l_f)$$

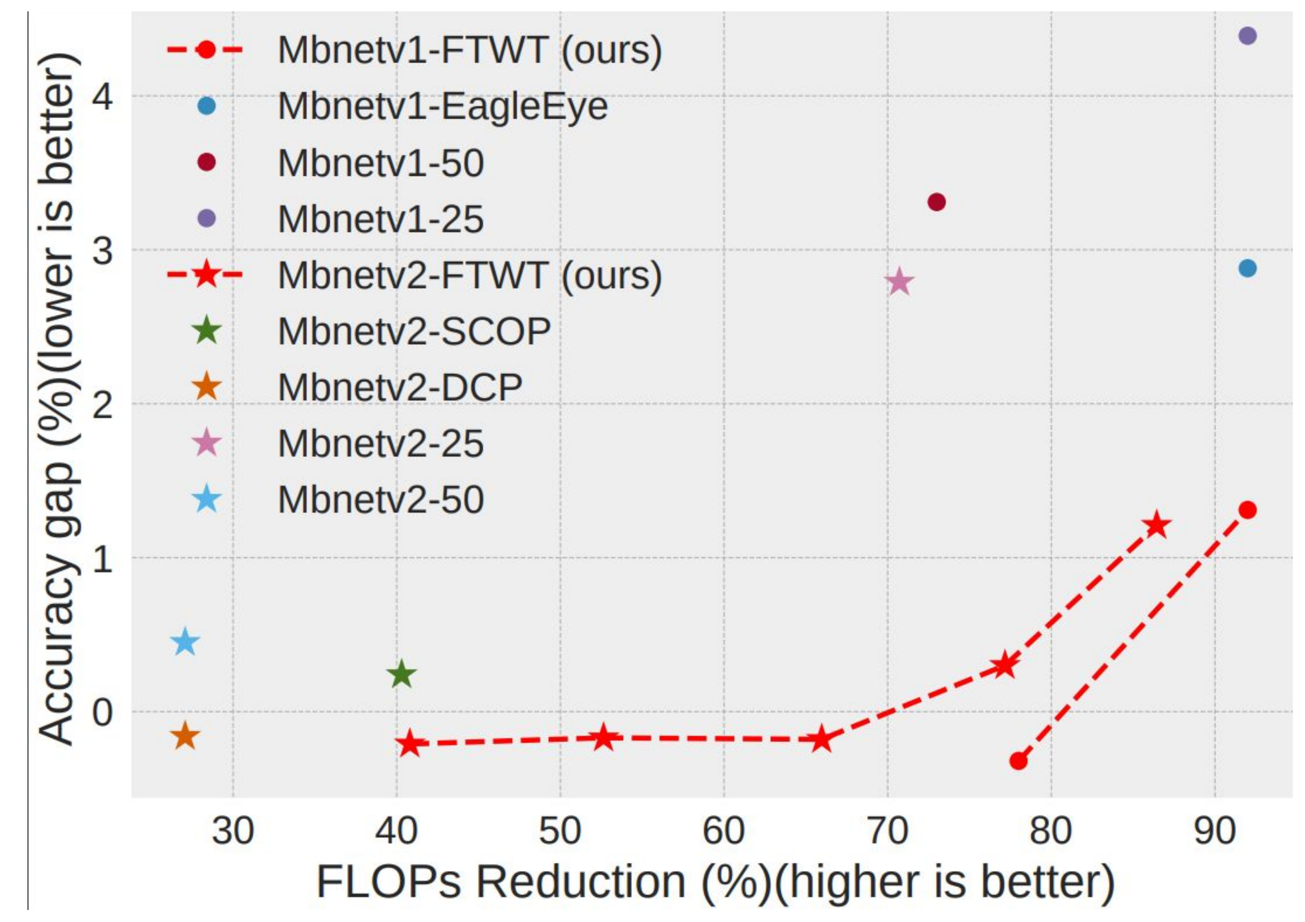
- **Top-k Filters (Rank features block)**

r is a hyperparameter ∈ [0,1]
r=1 → select all filters that have GMP>0 (i.e. drop filters that do not activate any pixel)



Results

MobileNet - CIFAR



ImageNet

	Method	Dynamic?	Top-1 Acc. (%) Baseline	Pruned	Delta	FLOPs red. (%)
ResNet34	Taylor [34]	N	73.31	72.83	0.48	22.25
	LCCL [6]	Y	73.42	72.99	0.43	24.80
	FTWT ($r = 0.97$)	Y	73.30	73.25	**0.05**	25.86
	FTWT ($r = 0.95$)	Y	73.30	72.79	0.51	37.77
	SFP [13]	N	73.92	71.83	2.09	41.10
	FPGM [14]	N	73.92	72.54	1.38	41.10
	FTWT ($r = 0.93$)	Y	73.30	72.17	**1.13**	47.42
	ResNet18 [12]	N	73.30	69.76	3.54	50.04
	FTWT ($r = 0.92$)	Y	73.30	71.71	**1.59**	52.24
ResNet18	PFP-B [24]	N	69.74	65.65	4.09	43.12
	SFP [13]	N	70.28	67.10	3.18	41.80
	LCCL [6]	Y	69.98	66.33	3.65	34.60
	FBS [10]	Y	70.70	68.20	2.50	49.49
	FTWT ($r = 0.91$)	Y	69.76	67.49	**2.27**	51.56
MobileNetV1	MobileNetV1-75 [16]	N	69.76	67.00	2.76	42.85
	FTWT ($r = 1$)	Y	69.57	69.66	**-0.09**	41.07

***2x** FLOPs reduction in ResNet34*

***1.7x** FLOPs reduction in MobileNetv1*

* Compare with the motivation

We discuss more in our paper on:

- Decoupled vs joint training.
- Selection of hyperparameter r.
- Out-of-distribution tests.
- Challenges and limitations with latency reduction.

Code is available at: https://github.com/selkerdawy/FTWT