IEEE Big Data, 2021

CUP: Cluster Pruning for Compressing Deep Neural Networks





Rahul Duggal
Georgia Tech

Cao Xiao
Amplitude

Richard Vuduc
Georgia Tech

Polo Chau
Georgia Tech

Jimeng Sun
UIUC

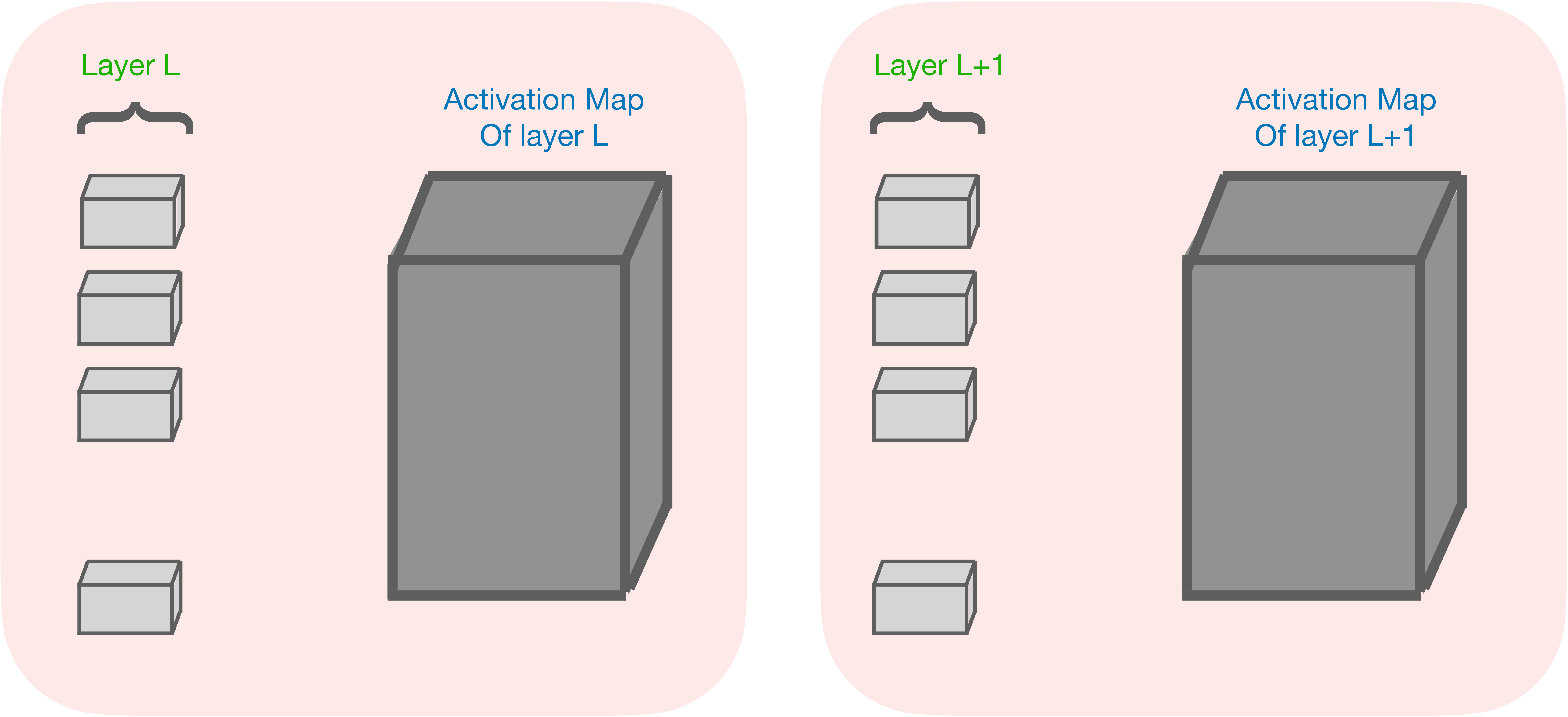
Goal

Reduce the ***storage*** and ***computation*** cost of a DNN

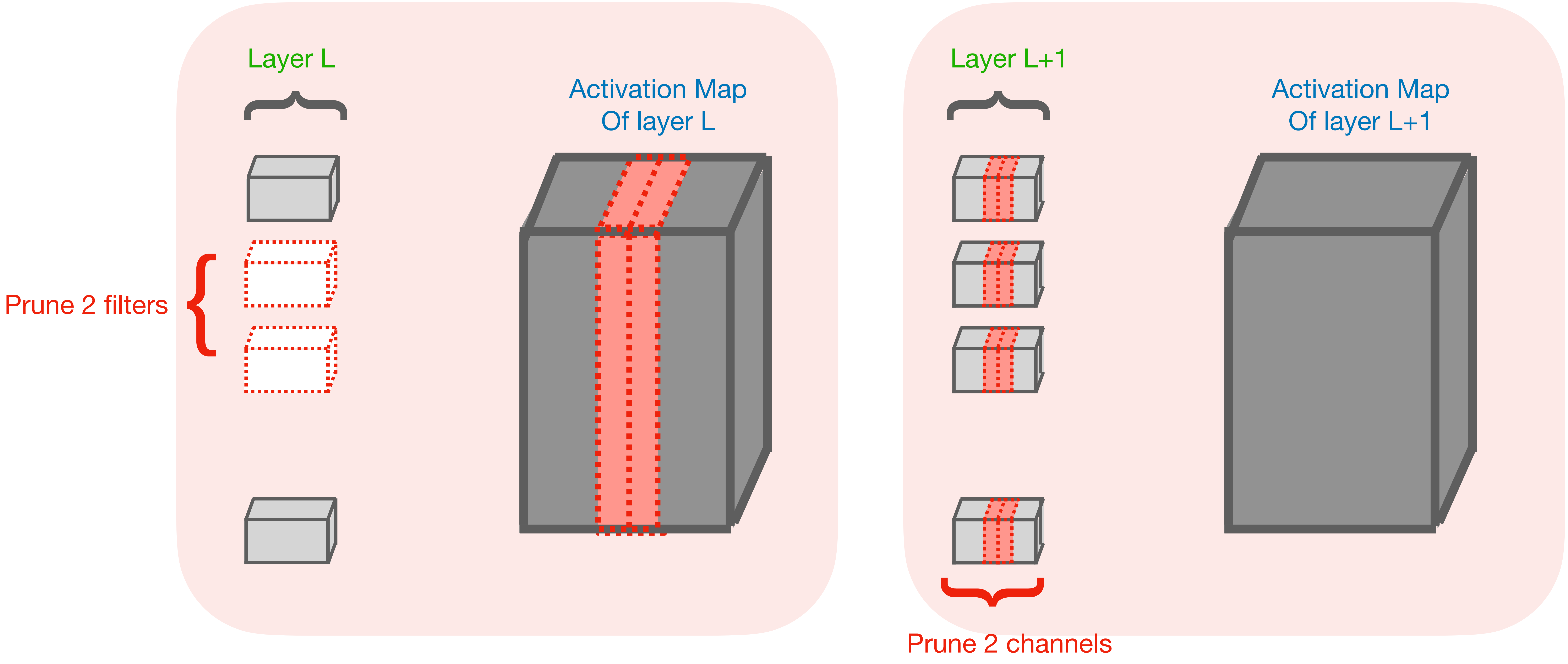
$$\mathbf{F}(x; W) \approx \mathbf{F}(x; W_{compressed})$$

Such that $|W_{compressed}| \ll |W|$

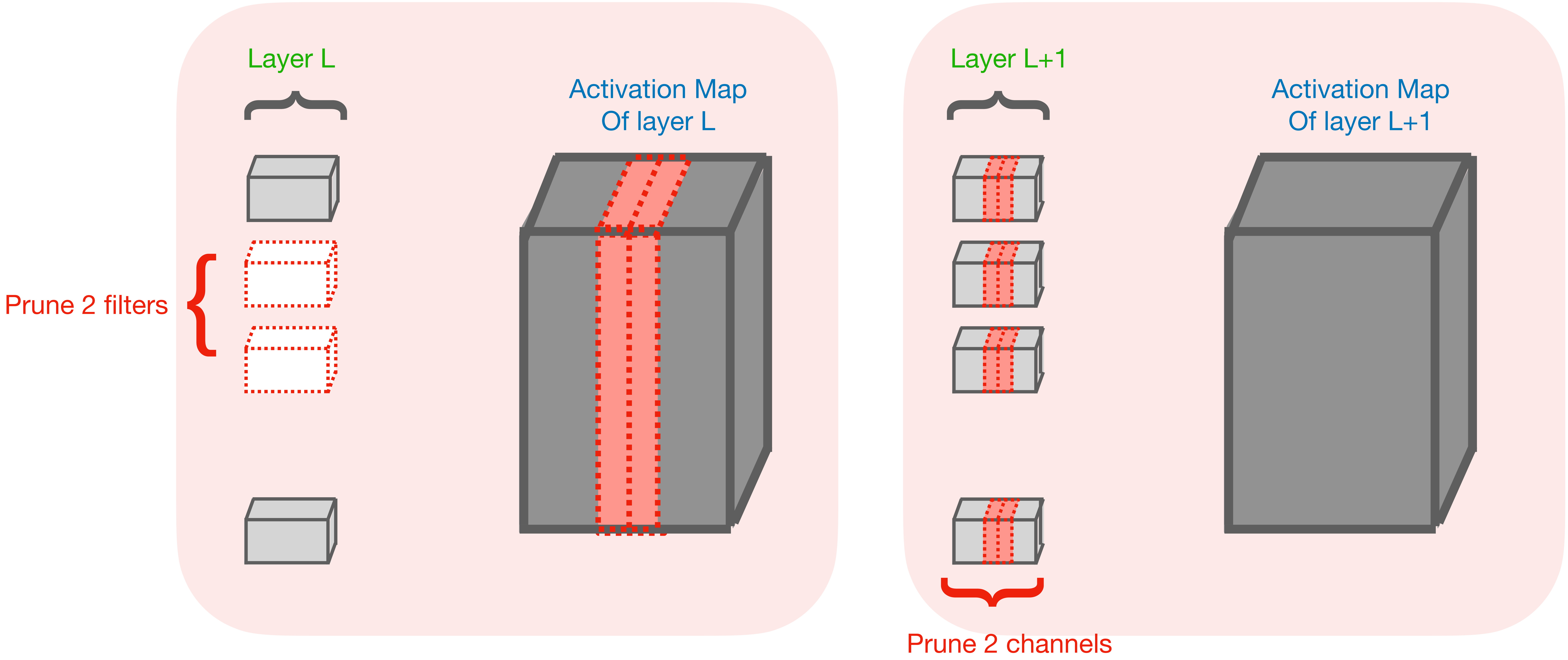
Filter Pruning



Filter Pruning



Filter Pruning



Which filters to prune?

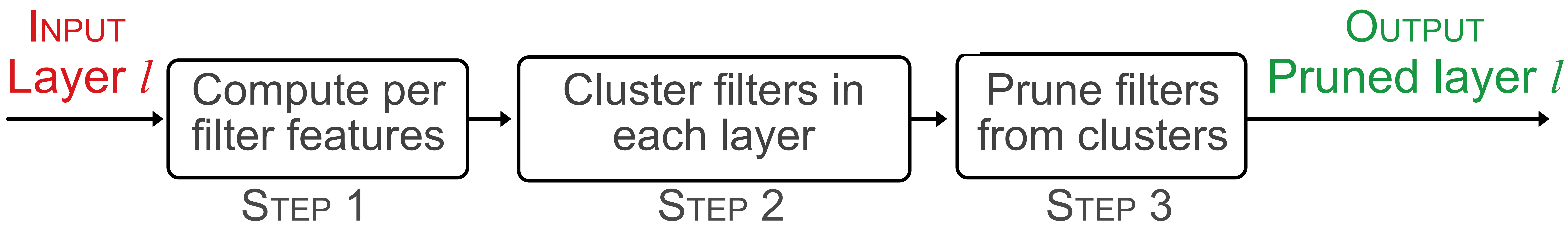
Our method

CUP: Cluster Pruning

Our Idea: Prune **similar filters**

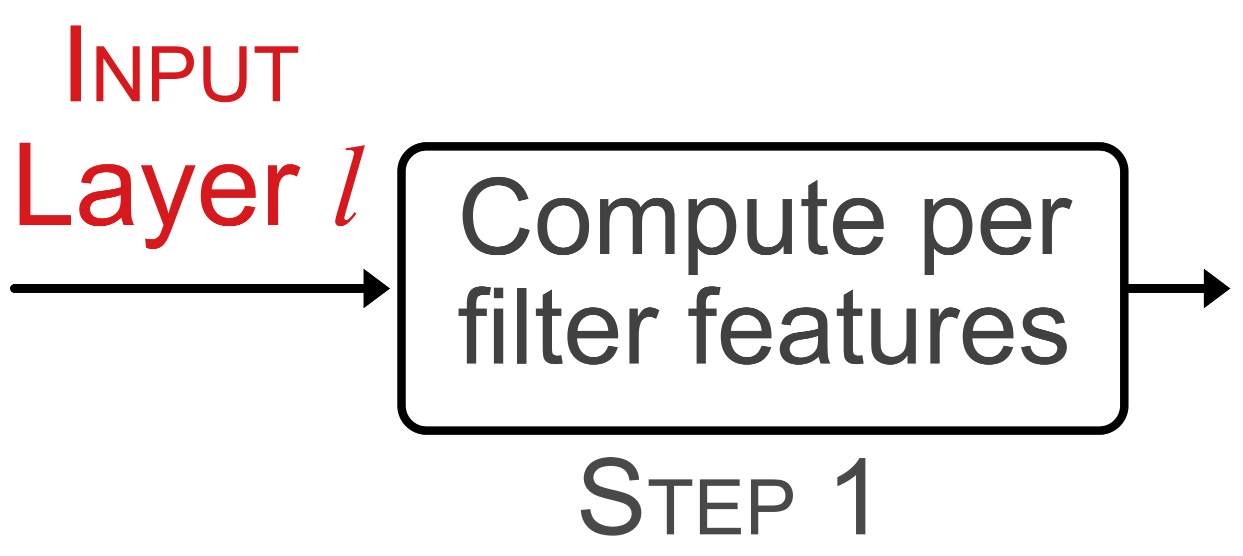
CUP: Cluster Pruning

Our Idea: Prune **similar filters**



Our method

CUP: Cluster Pruning



Fully Connected Layer

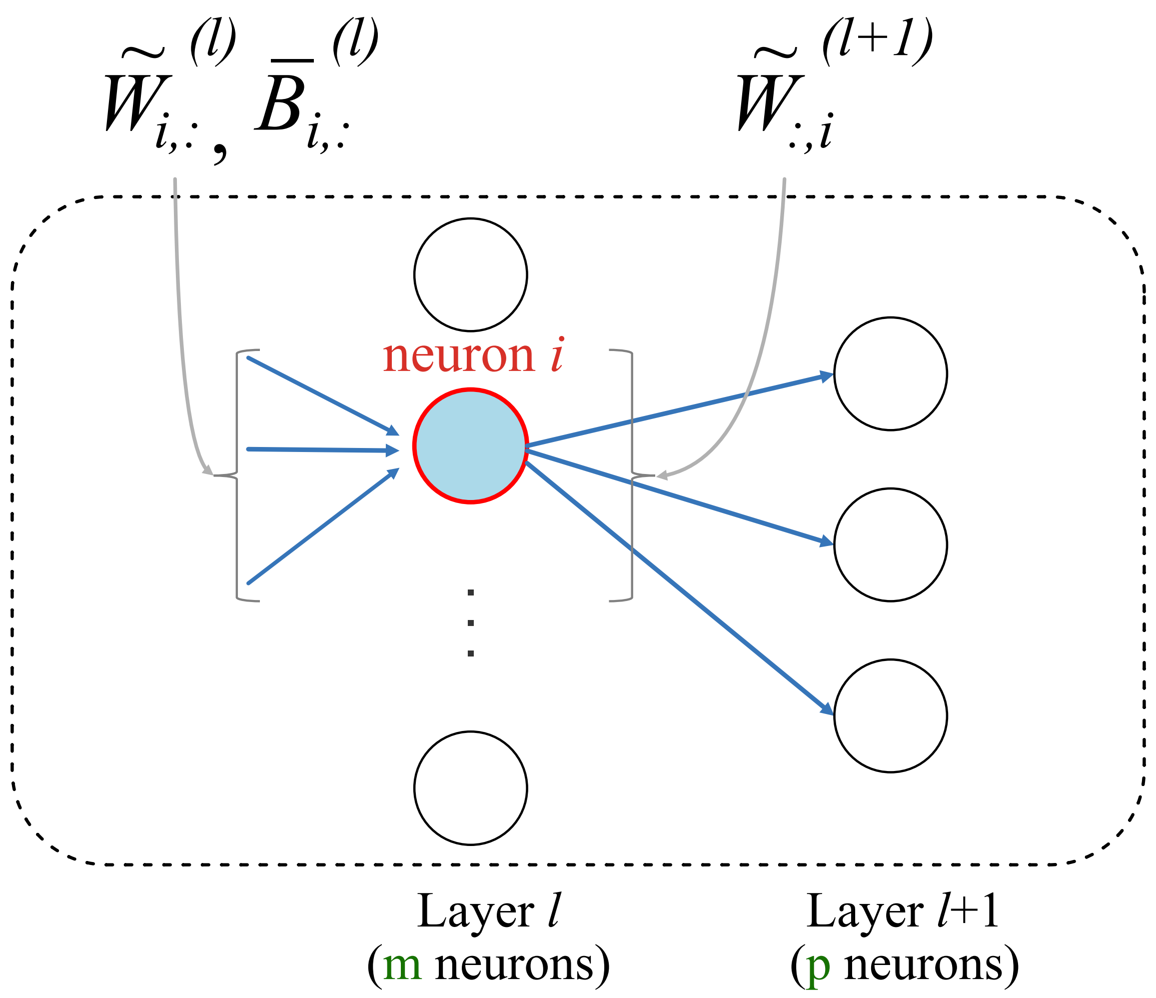
Convolutional Layer

Our method

CUP: Cluster Pruning

INPUT
Layer l → **Compute per filter features** →
STEP 1

Fully Connected Layer

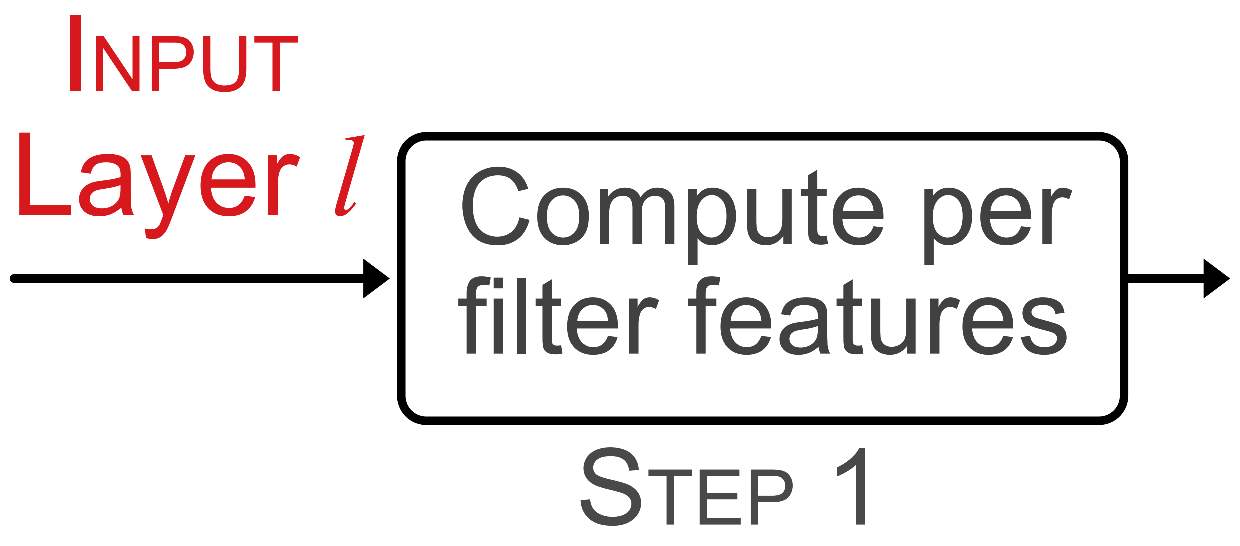


$$\widetilde{F}^{(l)}_{i,:} = [\underbrace{\widetilde{W}^{(l)}_{i,:}, \overline{B}^{(l)}_{i}}_{\text{Incoming features}} , \underbrace{\widetilde{W}^{(l+1)}_{:,i}}_{\text{Outgoing features}}]$$

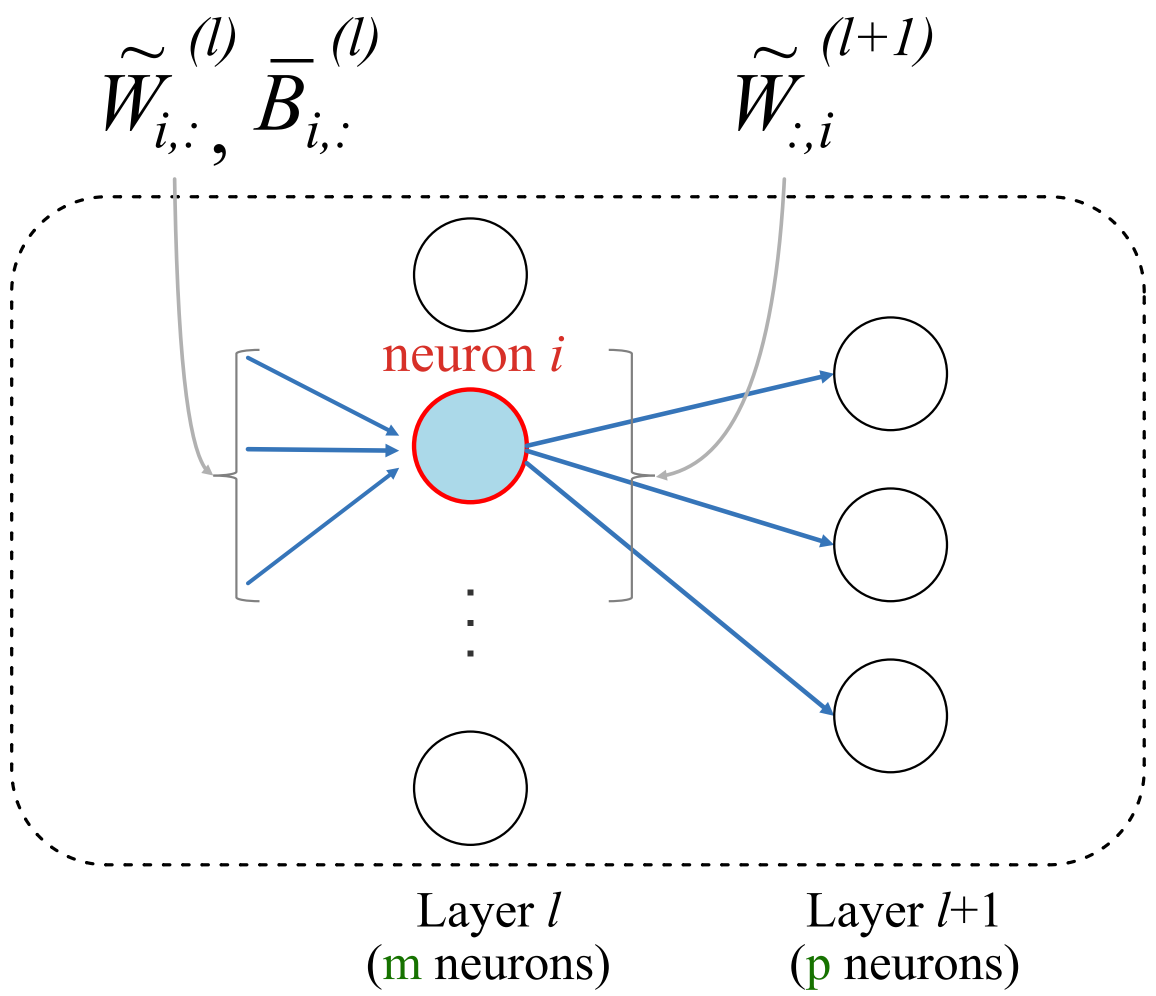
Convolutional Layer

Our method

CUP: Cluster Pruning

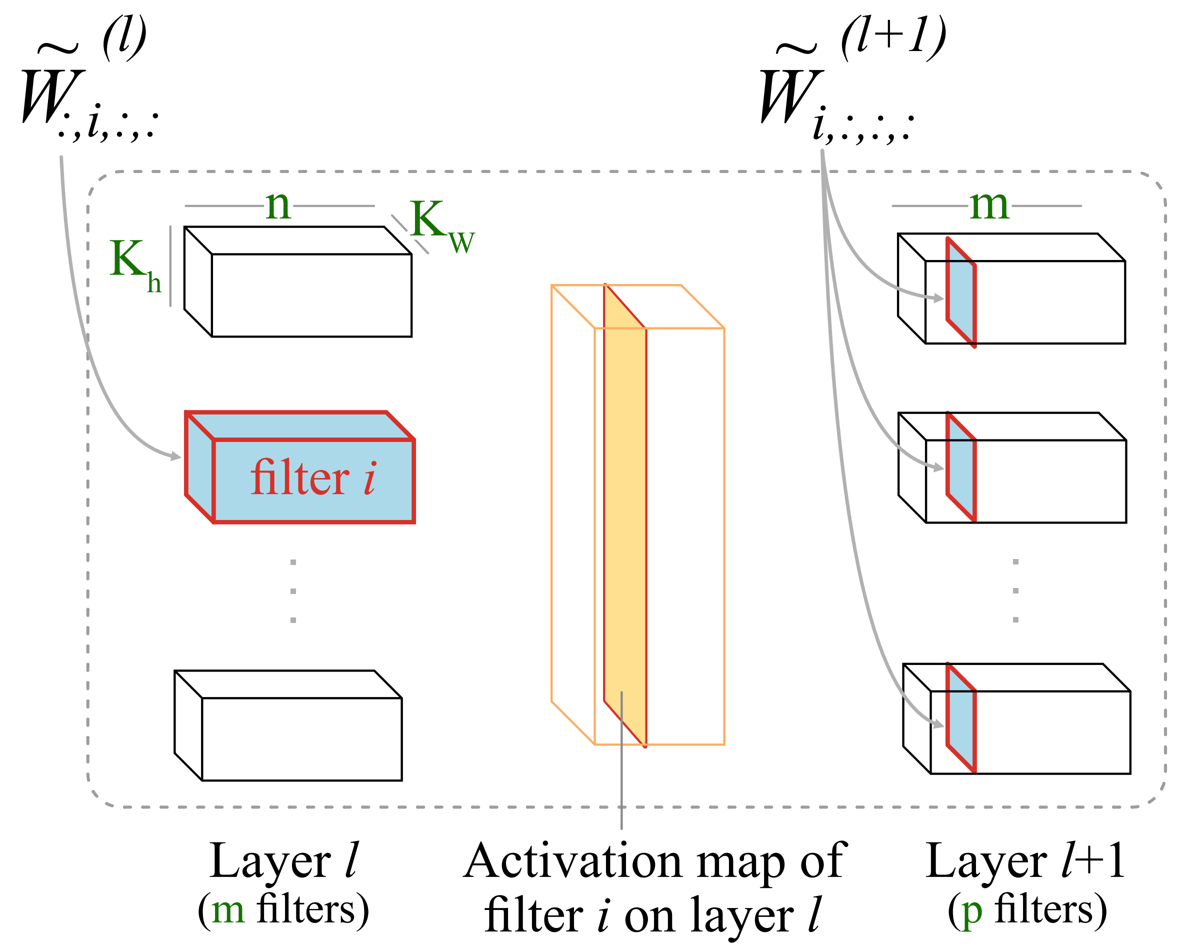


Fully Connected Layer



$$\widetilde{F}^{(l)}_{i,:} = [\underbrace{\widetilde{W}^{(l)}_{i,:}, \overline{B}^{(l)}_{i}}_{\text{Incoming features}}, \underbrace{\widetilde{W}^{(l+1)}_{:,i}}_{\text{Outgoing features}}]$$

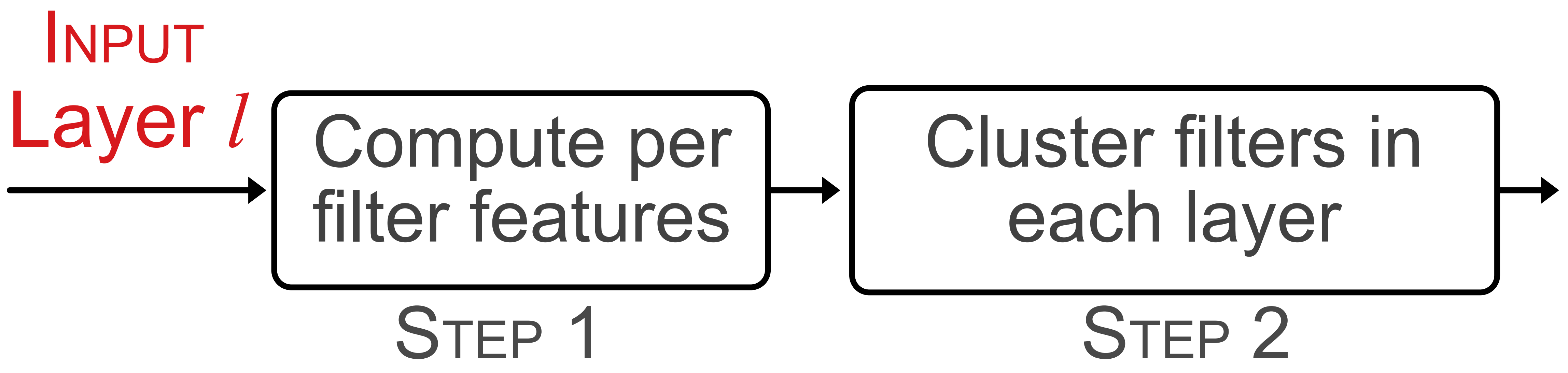
Convolutional Layer

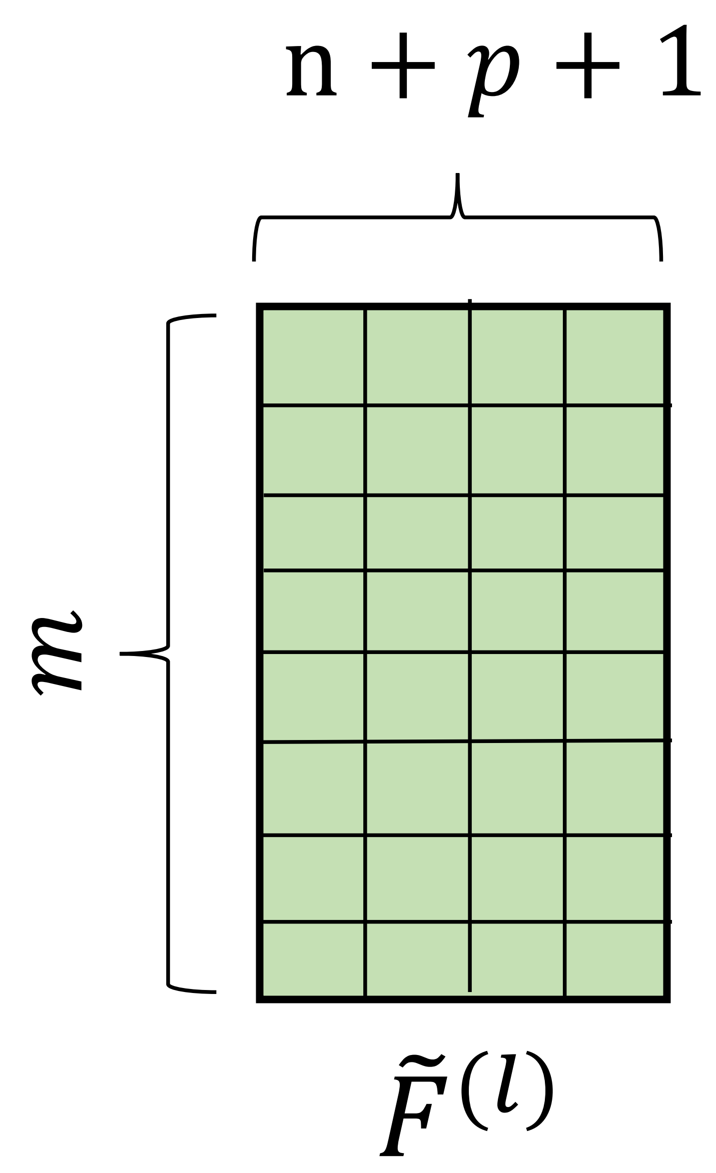


$$\widetilde{F}^{(l)}_{i,:} = [\underbrace{g(\widetilde{W}^{(l)}_{:,i,:,:}), \overline{B}^{(l)}_{i}}_{\text{Input features}}, \underbrace{g(\widetilde{W}^{(l+1)}_{i,:,:,:})}_{\text{Output features}}]$$

Our method

CUP: Cluster Pruning



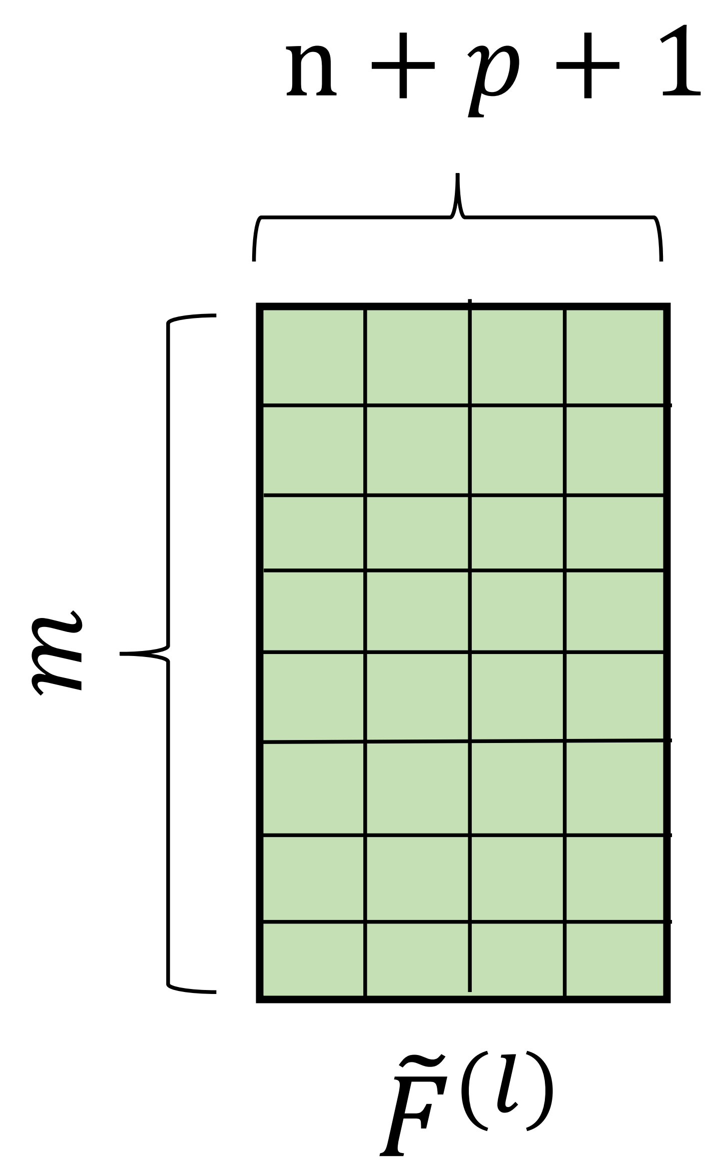


Our method

CUP: Cluster Pruning

INPUT

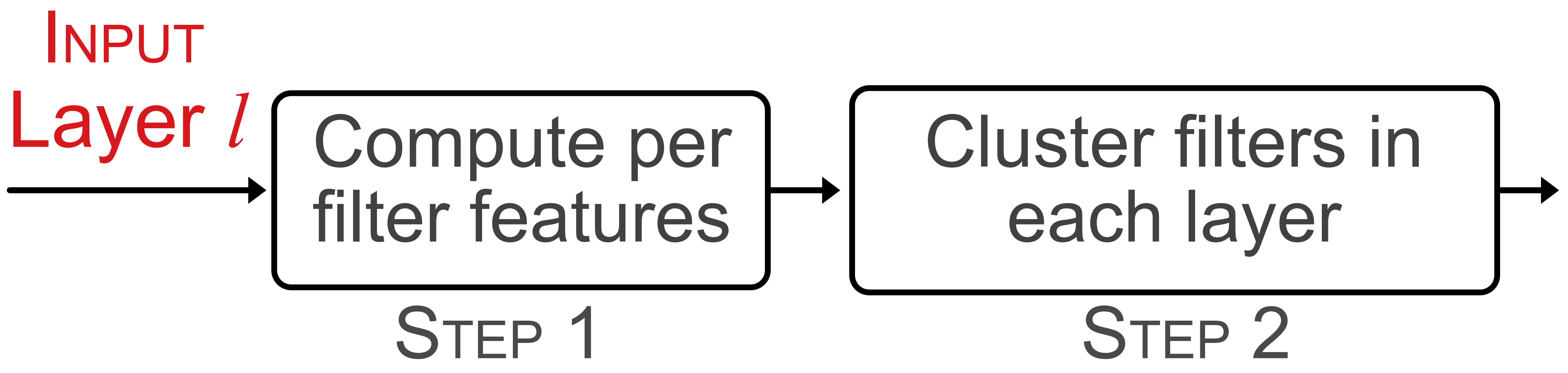
Layer l → Compute per filter features (STEP 1) → Cluster filters in each layer (STEP 2) →

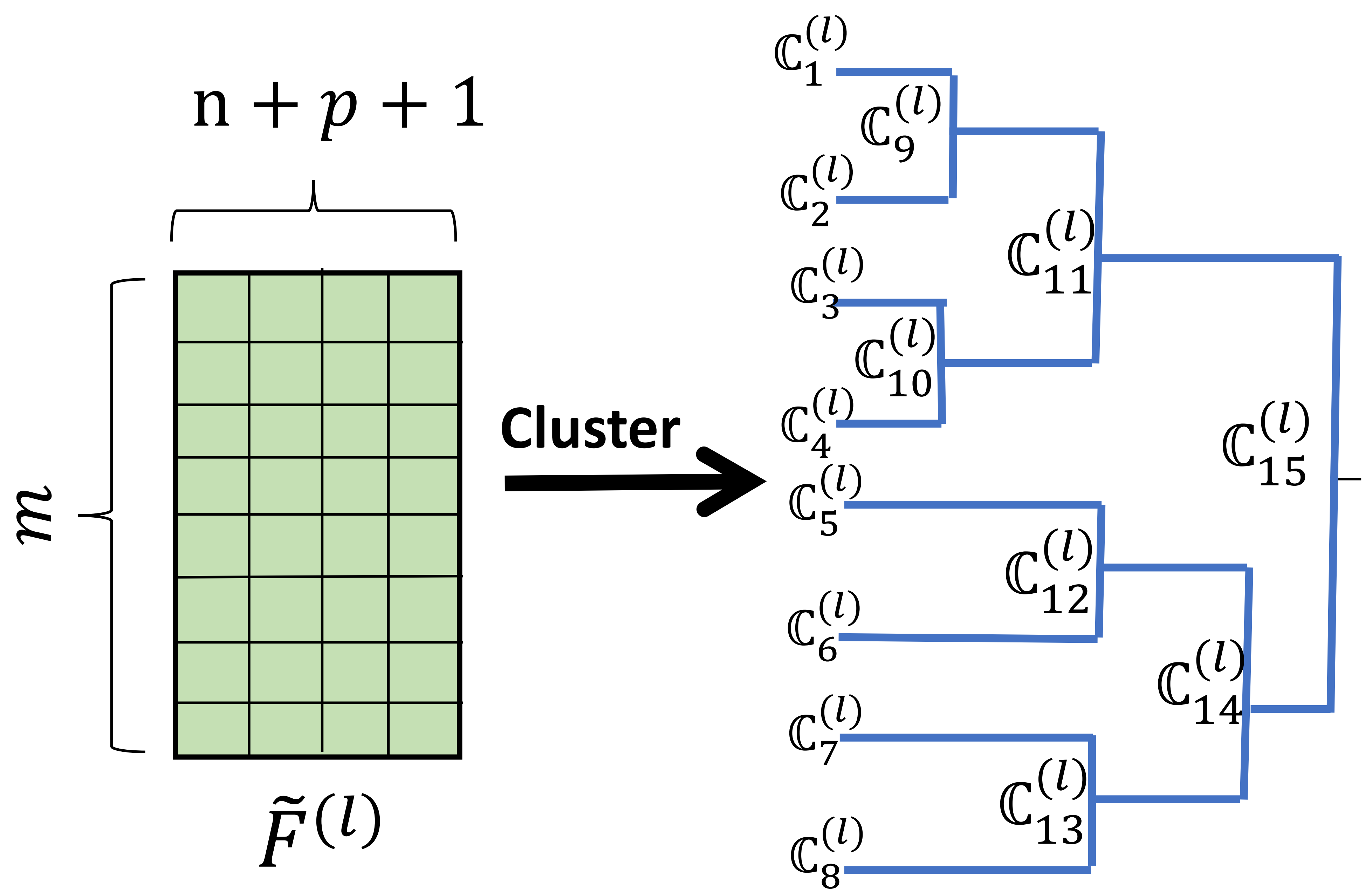


How many clusters?

Our method

CUP: Cluster Pruning

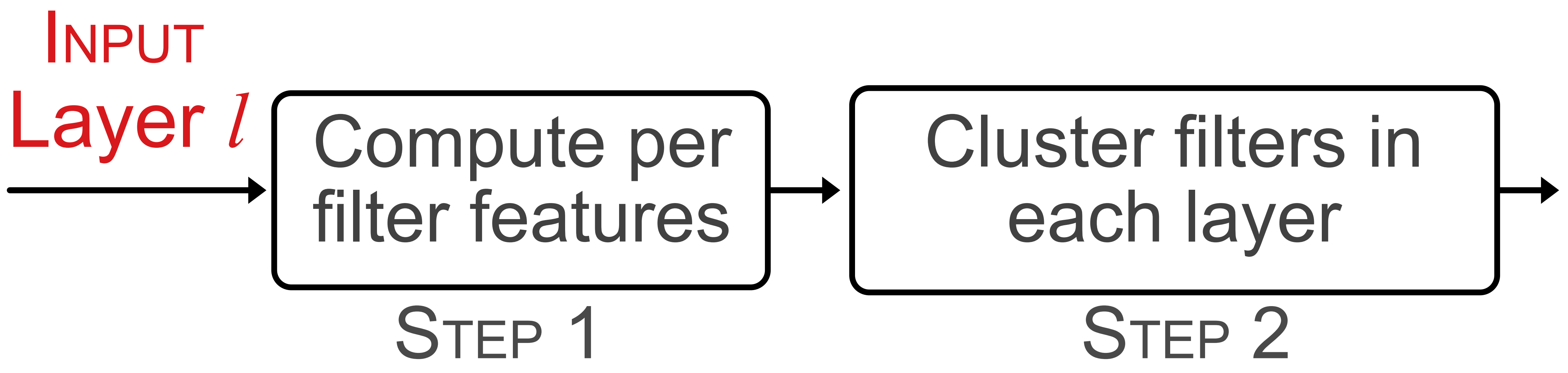


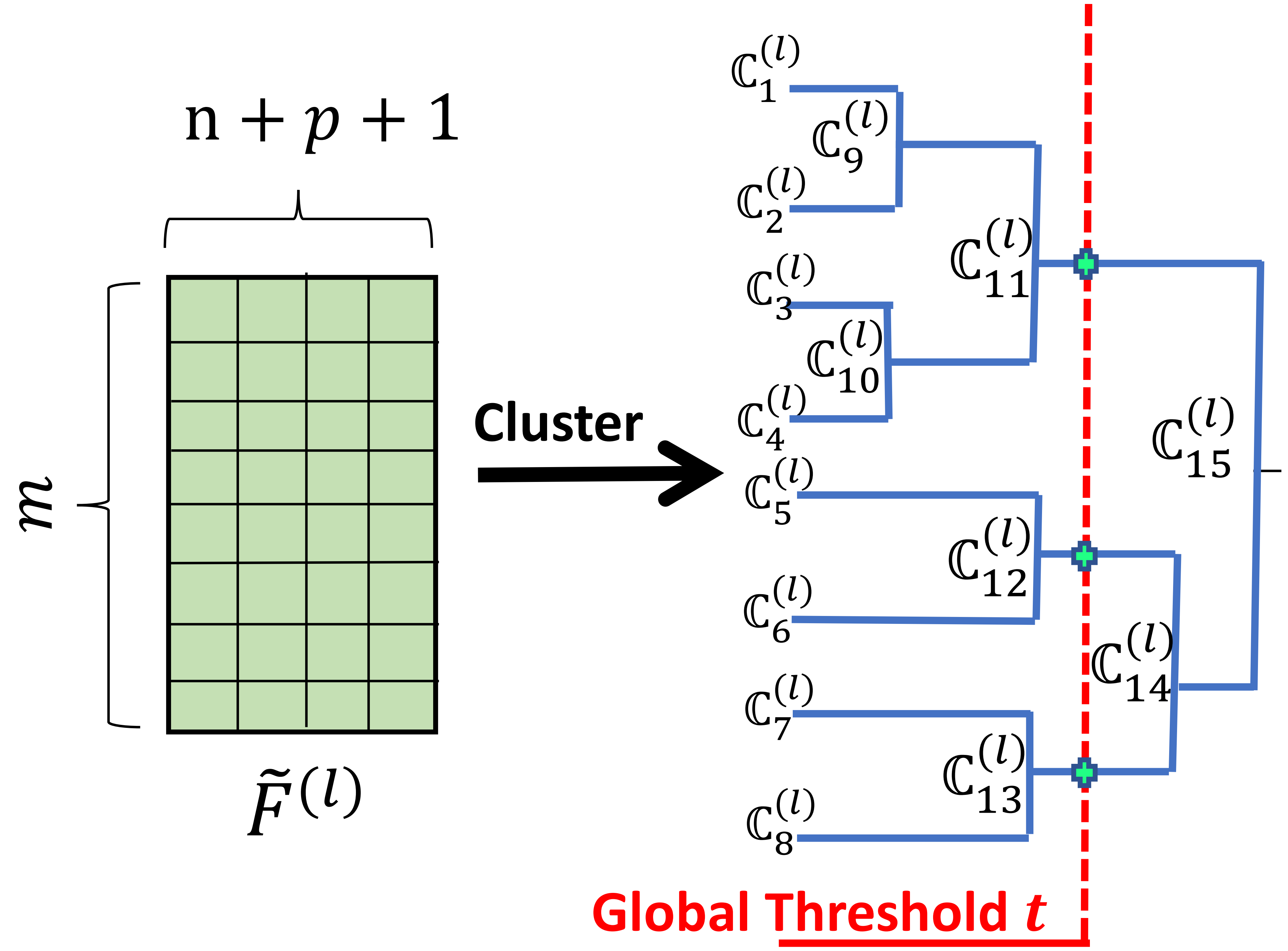


How many clusters?

Our method

CUP: Cluster Pruning

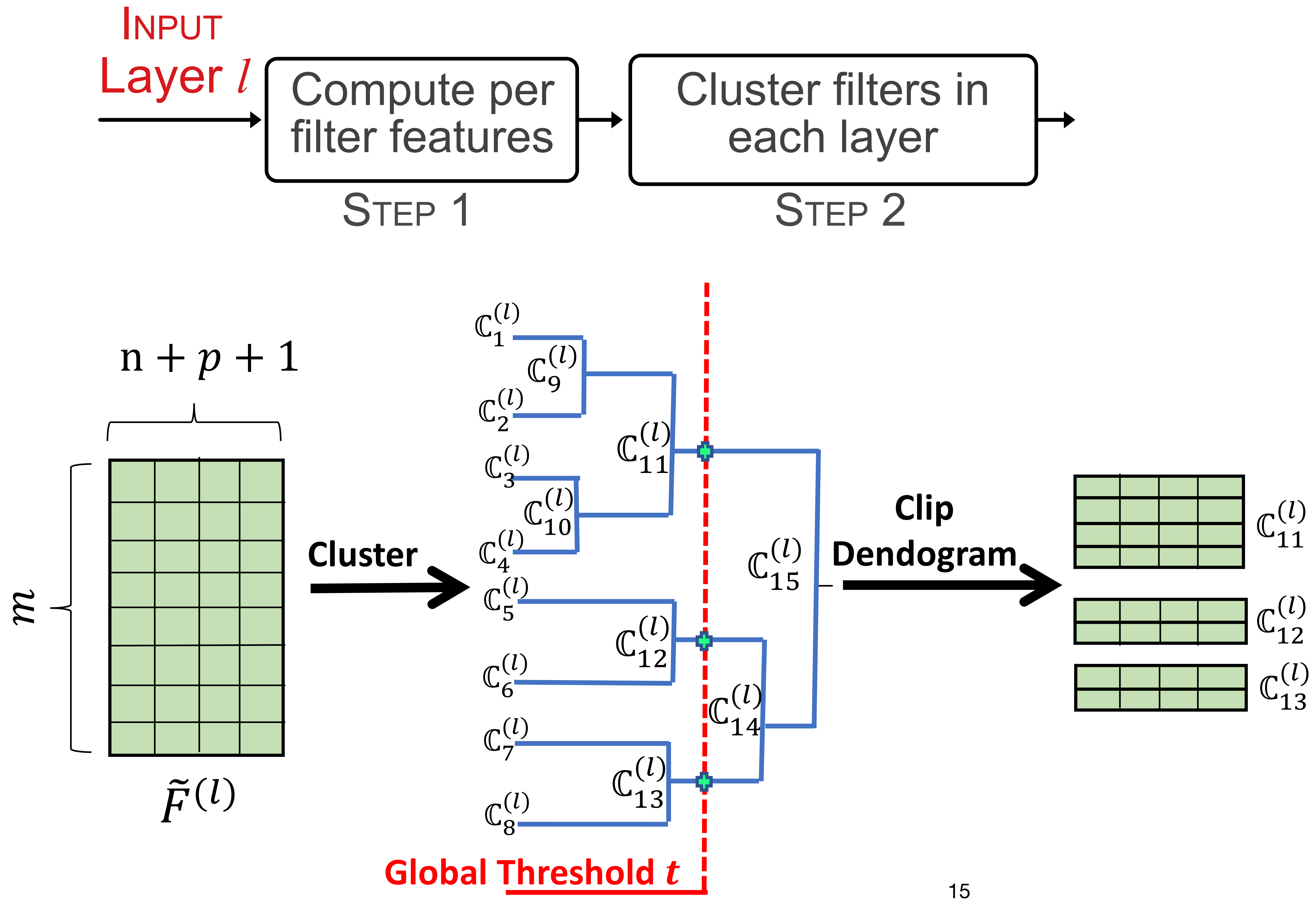




How many clusters?

Our method

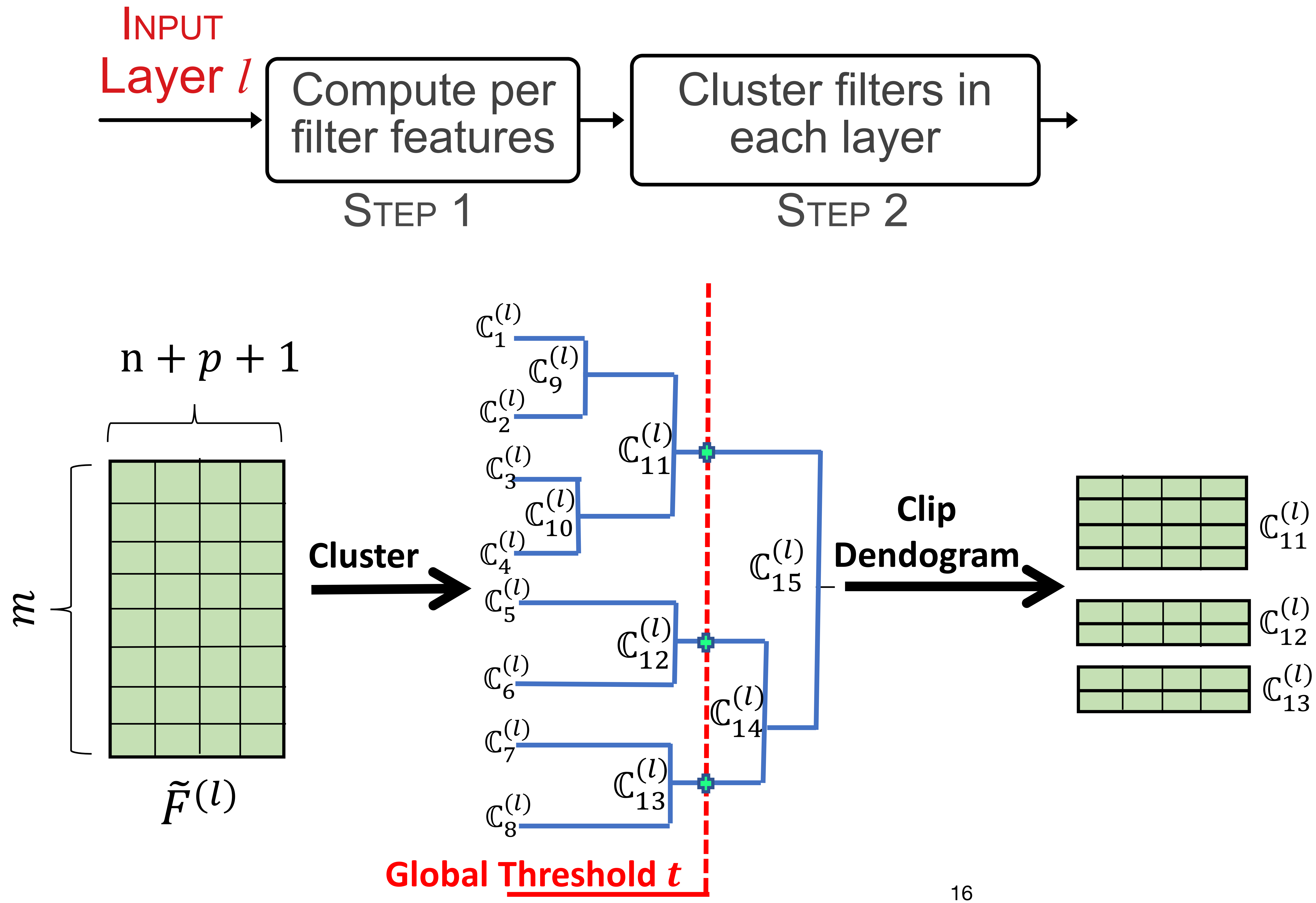
CUP: Cluster Pruning



How many clusters?

Our method

CUP: Cluster Pruning



How many clusters?

t parameterizes the number of clusters

Our method

CUP: Cluster Pruning

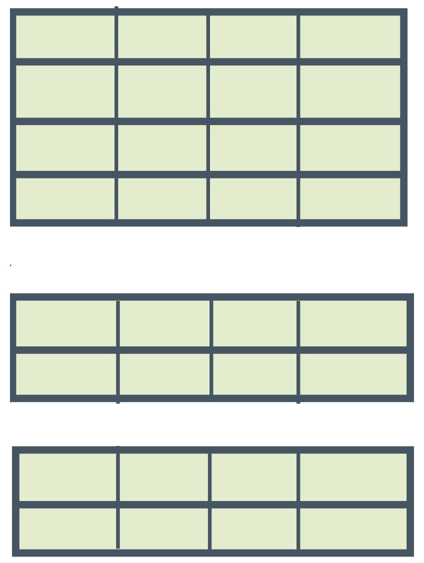
INPUT
Layer l

Compute per filter features
STEP 1

Cluster filters in each layer
STEP 2

Prune filters from clusters
STEP 3

OUTPUT
Pruned layer l

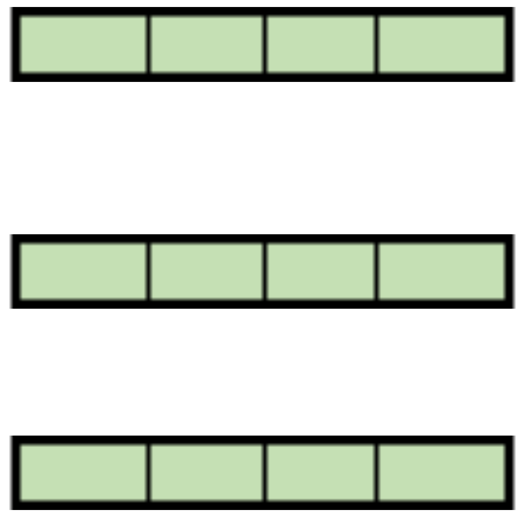


$\mathbb{C}_{11}^{(l)}$

$\mathbb{C}_{12}^{(l)}$

$\mathbb{C}_{13}^{(l)}$

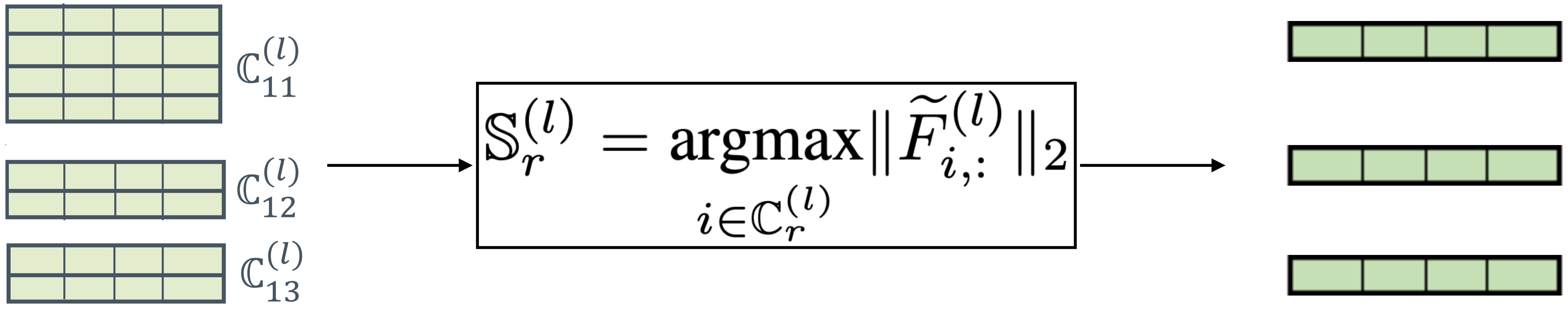
$$\boxed{\mathbb{S}_r^{(l)} = \underset{i \in \mathbb{C}_r^{(l)}}{\operatorname{argmax}} \| \widetilde{F}_{i,:}^{(l)} \|_2}$$



Our method

CUP: Cluster Pruning

INPUT
Layer l
→ Compute per filter features (STEP 1) → Cluster filters in each layer (STEP 2) → Prune filters from clusters (STEP 3) →
OUTPUT
Pruned layer l

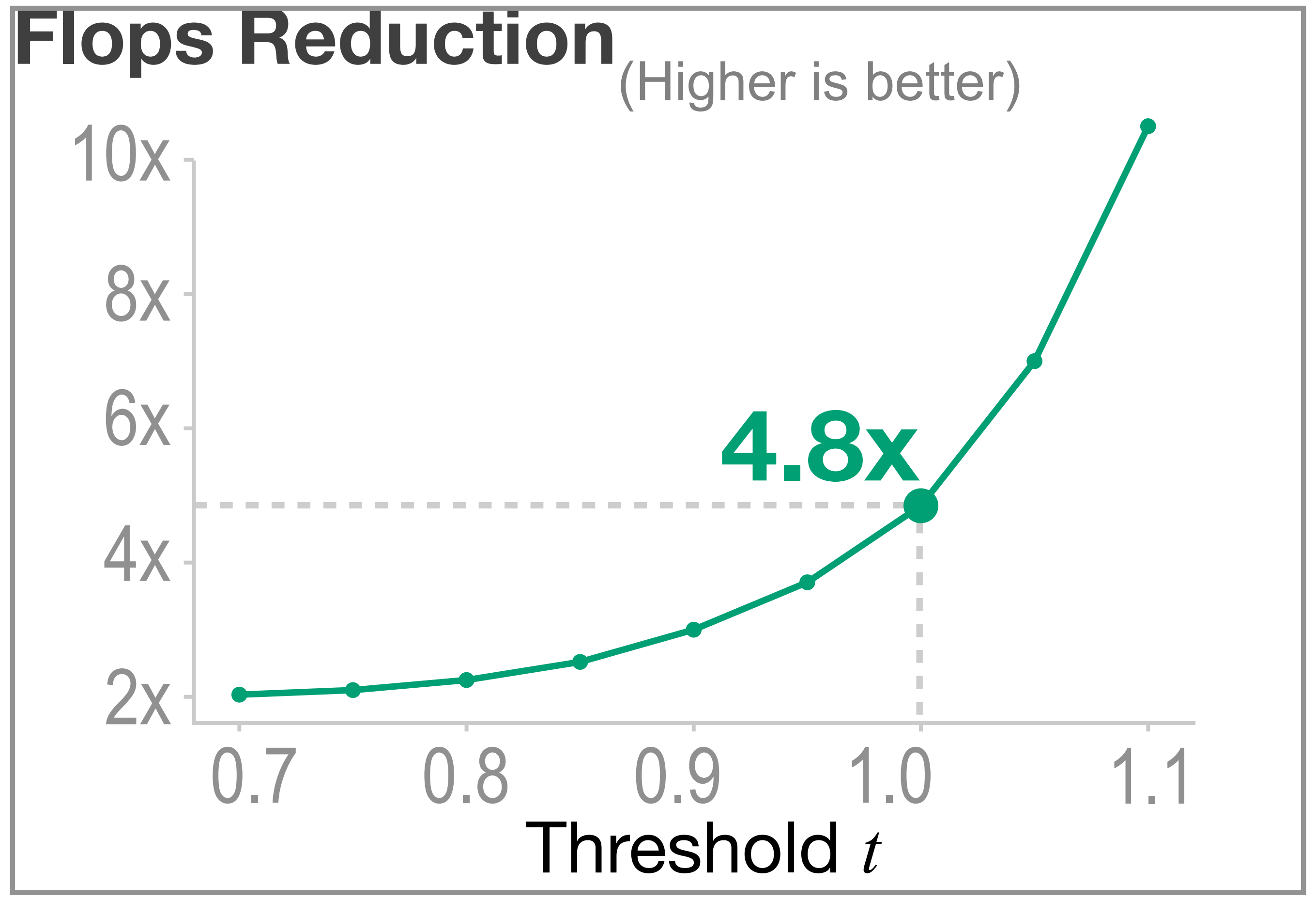


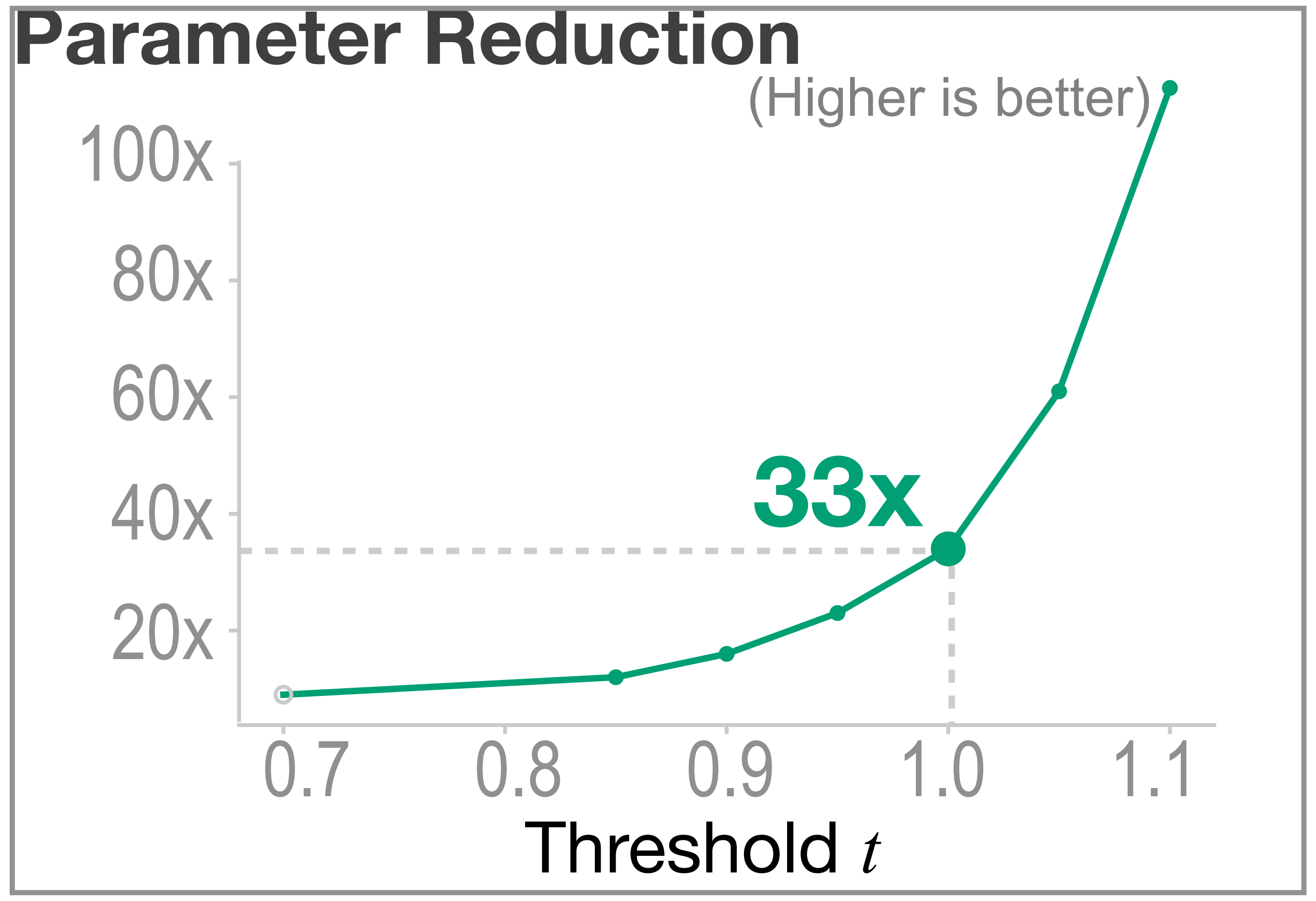
#clusters = # remaining filters

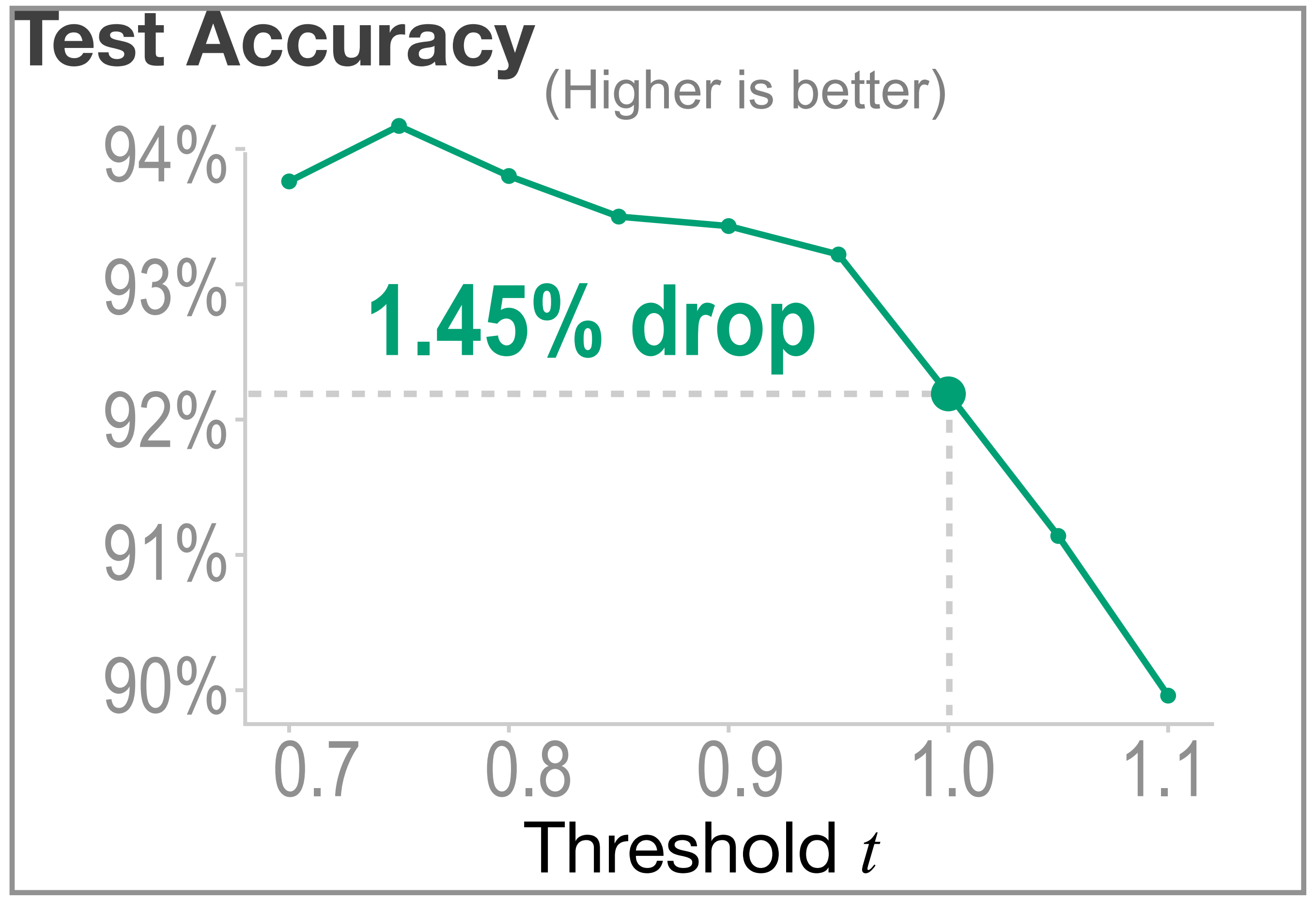
t parameterizes pruning amount

Results

Benefit 1: Single hyper parameter control over pruning amount

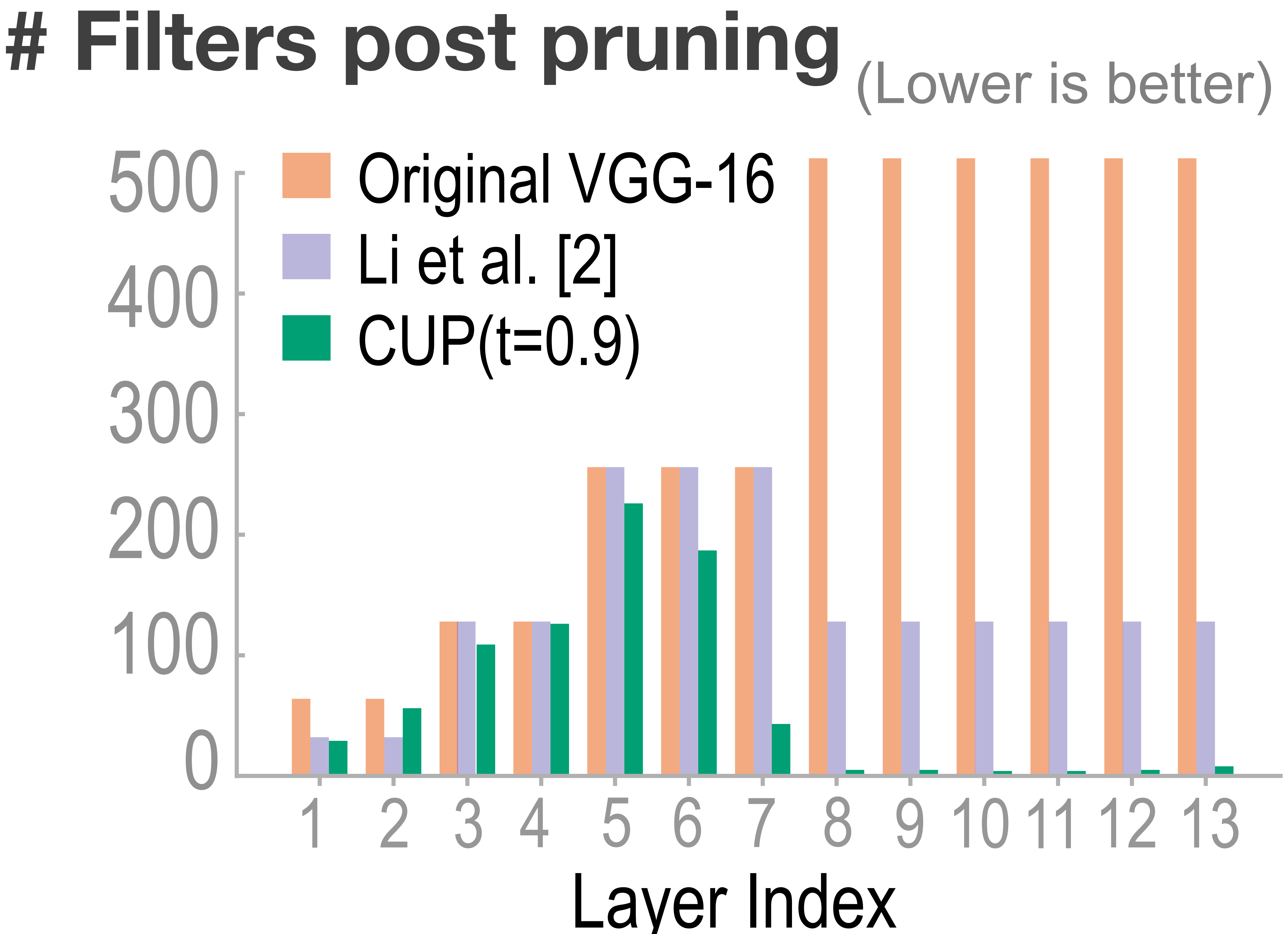






Results

Benefit 2: Non uniform pruning with a single hyper-parameter t



Results

Benefit 3: Training time reduction through train time pruning.

Method	Retrain?	Top-1 (%)	FR (×)	Training Time (GPU Hours)
Resnet-50	-	75.86	1.00	66.0
SFP [14]	✗	74.01	1.73	61.8
GM [15]	✗	74.13	2.15	62.2
CUP-RF (ours)	✗	**74.34**	**2.21**	**51.6**

~15 hours saving with 2x compression

Results

Benefit 4: State-of-the-art compression

Model	Method	Retrain?	FR (×)	Acc. (Δ%) Top-1	Acc. (Δ%) Top-5
ResNet-18	GM [15]	✓	1.71	-1.87	-1.15
	COP [29]	✓	**1.75**	-2.48	-
	CUP (Our)	✓	**1.75**	**-1.00**	**-0.79**
	SFP [14]	✗	1.71	-3.18	-1.85
	GM [15]	✗	1.71	-2.47	-1.52
	CUP-RF (ours)	✗	**1.75**	**-2.37**	**-1.40**
ResNet-34	L1 [2]	✓	1.31	-1.06	-
	GM [15]	✓	1.69	-1.29	-0.54
	CUP (ours)	✓	**1.78**	**-0.86**	**-0.53**
	SFP [14]	✗	1.69	-2.09	-1.29
	GM [15]	✗	1.69	-2.13	-0.92
	CUP-RF (ours)	✗	**1.71**	**-1.61**	**-0.89**
ResNet-50	SFP [14]	✓	2.15	-14.0	-8.20
	MP [30]	✓	2.05	-1.20	-
	CUP (ours)	✓	**2.47**	**-1.17**	**-0.81**
	SFP [14]	✗	1.71	-1.54	**-0.81**
	GM [15]	✗	2.15	-2.02	-0.93
	CUP-RF (ours)	✗	**2.20**	**-1.47**	-0.88

Conclusion

Thank you!

CUP: Cluster pruning framework

- Prunes a DNN by clustering similar filters.

Benefits of CUP

- Single hyper-parameter control over pruning amount.
- Enables non uniform pruning across layers.
- Train time savings.

Extensive evaluation on large DNNs & datasets