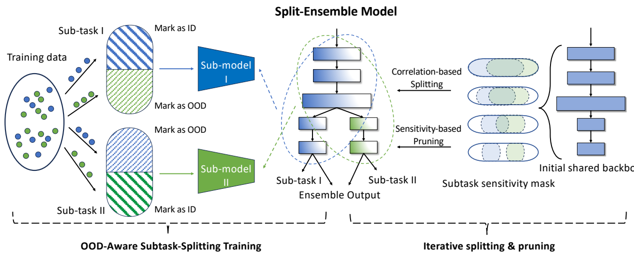
Split-Ensemble: Efficient OOD-aware Ensemble via Task and Model Splitting

Anthony Chen[1*], Huanrui Yang[2*†], Yulu Gan[1*], Denis A Gudovskiy[3], Zhen Dong[2]
Haofan Wang[4], Tomoyuki Okuno[3], Kurt Keutzer[2], Shanghang Zhang[1†]
[1]Peking University & [2]UCB & [3]Panasonic Inc. & [4]CMU * Indicates equal contribution † indicates corresponding author

Scan to view PDF

Motivation & Contribution

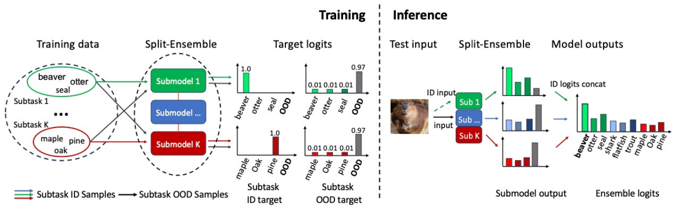


Motivation

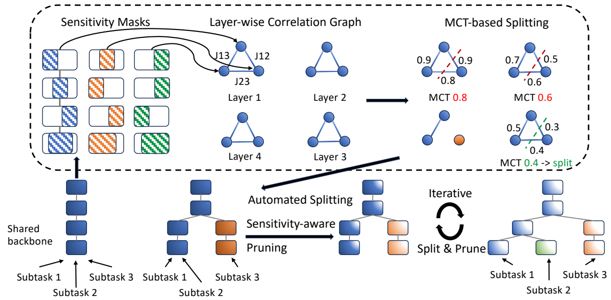
- Uncertainty estimation is crucial for deep learning models to detect out-of-distribution (OOD) inputs. Yet, improving the uncertainty estimation typically requires external data for OOD-aware training or considerable costs to build an ensemble. In this work, we improve on uncertainty estimation without extra OOD data or additional inference costs using an alternative Split-Ensemble method.

What we propose

1) a subtask-splitting training objective for OOD-aware ensemble training without external data. **2)** a dynamic splitting and pruning algorithm to build an efficient tree-like Split-Ensemble architecture corresponding to the subtask splitting. **3)** significantly improves accuracy and OOD detection over a single model baseline with a similar computational cost, and outperforms larger ensemble baselines by a factor of 4×.



Methods



Subtask Splitting

- We semantically group the N classes into K groups, each group classes are assigned to one specific submodel as In-Distribution(ID) data, while all other classes are Out-Of-Distribution(OOD) for this submodel.
- Inspired by Outlier Exposure, a normal one-hot label is used for ID data, yet a uniform label is utilized for OOD data.

Model Splitting & Pruning

- We start ensemble training from a single backbone, (i.e. all submodels share same architecture and parameters, except the final FC layer)
- We use SNIP to calculate sensitivity masks for each layer across different submodels, and build a correlation graph based on the mask IoU score, which is used to decide whether two submodels can be split or not. (Minimal Cutting Threshold, MCT): minimal correlation threshold for edge removal to cut the graph into two)
- Iteratively, to remove redundancy in submodels for simpler subtasks, we do global structural pruning, but only to the filters that are prunable for all submodel sharing it.
- The splitting is fixed when all submodels have an individual in later layers, and pruning is stopped when the Floating-point Operations (FLOPs) meet a predefined computational budget (typically the FLOPs of the original single model backbone)

Results

Method	FLOPs	CIFAR-10 Acc (↑)	CIFAR-100 Acc (↑)
Single Model	1x	94.7 / 95.2	75.9 / 77.3
Naive Ensemble	4x	**95.7** / 95.5	**80.1 / 80.4**
MC-Dropout	4x	93.3 / 90.1	73.3 / 66.3
MIMO	4x	86.8 / 87.5	54.9 / 54.6
MaskEnsemble	4x	94.3 / 90.8	76.0 / 64.8
BatchEnsemble	4x	94.0 / 91.0	75.5 / 66.1
FilmEnsemble	4x	87.8 / 94.3	77.4 / 77.2
Split-Ensemble (Ours)	1x	95.5 / **95.6**	77.7 / 77.4

Classification results on CIFAR-10 and CIFAR-100

Method	TinyImageNet	ImageNet1K
Single Model	26.1	69.0
Naive Ensemble	44.6	69.4
Split-Ensemble (ours)	**51.6**	**70.9**

Classification results on TinyIMNet and IMNet

Method	Accuracy ↑	ECE ↓	FPR95 ↓	AUROC ↑	AUPR ↑
Naive Ensemble	12.7	50.2	98.4	45.3	50.9
MC-Dropout	63.4	25.8	90.6	66.6	66.1
MIMO	35.7	28.8	96.3	55.1	56.9
MaskEnsemble	67.7	24.6	89.0	66.82	67.4
BatchEnsemble	70.1	21.1	87.45	68.0	68.7
FilmEnsemble	72.5	21.3	84.32	75.5	76.0
Split-Ensemble (ours)	**73.7**	**16.5**	**80.5**	**81.7**	**77.6**

Results on SC-OOD CIFAR10-LT benchmarks

Ablation

- Our accuracy is not sensitive to the number of splits, increasing it enables better OOD detection performance(if not over aggressively pruned)

# splits	2	4	5	8	10
Accuracy	77.7	**78.0**	77.9	77.5	77.3
AUROC	78.1	78.2	79.9	**80.4**	77.3

Cifar-100 to Cifar10

# splits	OOD class target	Accuracy	AUROC
2	One-hot	71.0	76.0
	OOD-aware	**77.7**	**78.1**
4	One-hot	77.2	77.5
	OOD-aware	**78.0**	**78.2**
5	One-hot	77.7	77.3
	OOD-aware	**77.9**	**78.1**

Cifar-100 to Cifar10

- We use an outlier exposure-inspired target for the inputs belonging to the OOD class, to better calibrate the confidence during submodel training.