Appendix

A. Sparse MoE Structures

Overall MoE Design Fig. A1 shows the overall MoE design adopted in this work. By default, the experts in each layer are pre-defined with little channel overlapping. The router exactly selects one expert for a given input as its pathway component in this layer.

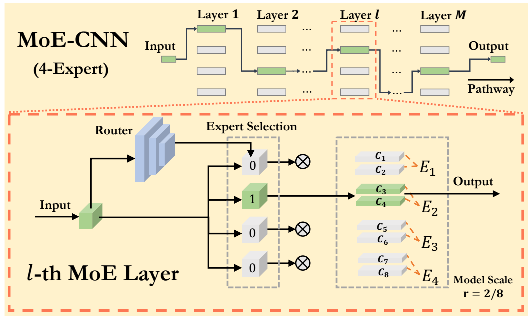

Figure A1. The sparse MoE-CNN structure and its MoE design in this paper. The router makes the input-specific expert selection and the selected experts (*e.g.*, E_2) form an end-to-end pathway (emphasized in green). This shows an example of the MoE layer with 4 experts with the model scale of 0.25.

B. Detailed Experiment Setups

Training Details We train all the methods for 100 epochs with an initial learning rate of 0.1 and a cosine decaying learning rate scheduler. In particular, following the original training pipeline of AT (Sparse) [50], we first train 100 epochs to optimize the mask for Sparse-CNN, and then finetune the model weights based on the fixed mask for another 100 epochs. We use the SGD optimizer for all the methods and a momentum value of 0.9 together with a weight decay factor of $5e^{-4}$. We use a batch size of 128 on all the datasets, except 512 for `ImageNet`.

For ADVMOE, we randomly sample *different* batches of data (of the same batch size b) for updating backbone networks (experts) and routers since the use of diverse data batches is confirmed to benefit generalization for bi-level learning like meta-learning [70] and model pruning [48].

Datasets and Model Backbones To implement MoE-CNN and other baselines, we conduct experiments on ResNet-18 [60], Wide-ResNet-28-10 (WRN-28-10) [61], VGG-16 [62], and DenseNet [63]. In particular, we adopt the ResNet-18 and WRN-28-10 with convolutional kernels of 3×3 in the first layer for `TinyImageNet`, `CIFAR-10` and `CIFAR-100`, and 7×7 for `ImageNet`, following the implementations in [71].

C. Additional Experiments

Ablation study on train-time attack generation steps In **Tab. 1.**, we adopt the 2-step PGD attacks to generate the train-time perturbation. Also, we conduct ablation studies on the train-time attack steps and raise its number from 2 to 10. We show the obtained results in **Tab. A1**. As we can see, the effectiveness of ADVMOE holds: Both RA and SA achieved by ADVMOE outperform its baselines by a substantial margin.

Table A1. Ablation study on the train-time attack step numbers. The attack step number used to generate train-time perturbation is raised to 10 from 2 compared the default setting. Other settings strictly follow **Tab 1**.

Method	ResNet-18				WRN-28-10			
	RA(%)	RA-AA(%)	SA(%)	GFLOPS	RA(%)	RA-AA(%)	SA(%)	GFLOPS
`CIFAR-10`								
● AT (Dense)	50.97±0.14	46.29±0.15	81.44±0.15	0.54	52.35±0.18	46.49±0.11	81.45±0.15	5.25
○ AT (S-Dense)	48.22±0.11	43.79±0.15	**79.93**±0.12	0.14 (74% ↓)	50.92±0.18	44.69±0.19	80.33±0.15	1.31 (75% ↓)
○ AT (Sparse)	48.29±0.14	43.18±0.19	79.35±0.17	0.14 (74% ↓)	48.69±0.18	44.50±0.16	80.32±0.11	1.31 (75% ↓)
○ AT (MoE)	46.79±0.49	41.13±0.29	78.32±0.51	0.15 (72% ↓)	47.24±0.57	42.39±0.26	76.21±0.42	1.75 (67% ↓)
○ **ADVMOE**	**52.22** ±0.14	**46.44** ±0.09	79.62±0.12	0.15 (72% ↓)	**56.13** ±0.11	**46.73** ±0.08	**82.19** ±0.14	1.75 (67% ↓)

Statistics for Fig. 7. In Fig. 7, we show the robustness comparison of different models in various model scale settings. In Tab. A2, we disclose the statistics for the plotting Fig. 7 as well as the GFLOPS for different model scales.

Table A2. Results of ADVMOE (our proposal) *vs.* baselines using different model scale settings on the datasets CIFAR-10 and CIFAR-100. The model scale $r \in \{0.2, 0.5, 0.8\}$ is considered. Other settings strictly follow Tab. 2. The statistics in this table are associated with the plots in Fig. 7.

Method	**model scale** $r = 0.2$			**model scale** $r = 0.5$			**model scale** $r = 0.8$			**AT (Dense)**, model scale $r = 1.0$		
	RA(%)	**SA(%)**	**GFLOPS**	**RA(%)**	**SA(%)**	**GFLOPS**	**RA(%)**	**SA(%)**	**GFLOPS**	**RA(%)**	**SA(%)**	**GFLOPs**
						CIFAR-10, ResNet-18						
AT (S-Dense)	43.83±0.11	78.28±0.14	0.13 (76% ↓)	48.12±0.09	80.18±0.11	0.14 (74% ↓)	49.44±0.09	81.32±0.11	0.36 (33% ↓)	50.13±0.13	82.99±0.11	0.54
AT (Sparse)	43.24±0.14	**79.14**±0.14	0.13 (76% ↓)	47.93±0.17	**80.45**±0.13	0.14 (74% ↓)	48.32±0.13	81.77±0.11	0.36 (33% ↓)			
AT (MoE)	38.75±0.41	76.54±0.29	0.14 (74% ↓)	45.57±0.51	78.84±0.75	0.15 (72% ↓)	45.99±0.42	79.46±0.31	0.37 (31% ↓)			
ADVMOE	**49.18**±0.12	**79.03**±0.19	0.14 (74% ↓)	**51.83** ±0.12	80.15±0.11	0.15 (72% ↓)	**52.38** ±0.14	81.44±0.13	0.37 (31% ↓)			
						CIFAR-10, WRN-28-10						
AT (S-Dense)	49.59±0.17	**79.93**±0.13	0.21 (96% ↓)	50.66±0.13	82.24±0.10	1.31 (75% ↓)	51.73±0.17	82.88±0.14	3.36 (38% ↓)	51.75±0.12	83.54±0.15	5.25
AT (Sparse)	48.37±0.21	79.32±0.21	0.21 (96% ↓)	48.95±0.14	82.44±0.17	1.31 (75% ↓)	50.73±0.19	82.11±0.23	3.36 (38% ↓)			
AT (MoE)	42.29±0.51	75.32±0.38	0.94 (82% ↓)	46.73±0.46	77.42±0.73	1.75 (67% ↓)	46.94±0.45	79.11±0.27	4.57 (13% ↓)			
ADVMOE	**54.02**±0.09	79.55±0.12	0.94 (82% ↓)	**55.73** ±0.13	**84.32** ±0.18	1.75 (67% ↓)	**56.07** ±0.14	**84.45** ±0.09	4.57 (13% ↓)			

Training trajectory ADVMOE. We show in Fig. A2 that the ADVMOE converges well within 100 training epochs using a cosine learning rate schedule. The SA (standard accuracy) and RA (robust accuracy) are evaluated and collected at the end of each training epoch.

(a) ResNet-18, CIFAR10 (b) WRN-28-10, CIFAR-10

Figure A2. The training trajectory of ADVMOE under different data-model settings.