Robust Mixture-of-Expert Training for Convolutional Neural Networks

Yihua Zhang[1] Ruisi Cai[2] Tianlong Chen[2,3,4,5] Guanhua Zhang[6] Huan Zhang[7,8]
Pin-Yu Chen[9] Shiyu Chang[6] Zhangyang Wang[2] Sijia Liu[1,9]

[1]Michigan State University, [2]University of Texas at Austin,
[3]The University of North Carolina at Chapel Hill, [4] MIT, [5]Harvard University,
[6]UC Santa Barbara, [7]Carnegie Mellon University, [8]UIUC, [9]IBM Research

Abstract

Sparsely-gated Mixture of Expert (MoE), an emerging deep model architecture, has demonstrated a great promise to enable high-accuracy and ultra-efficient model inference. Despite the growing popularity of MoE, little work investigated its potential to advance convolutional neural networks (CNNs), especially in the plane of adversarial robustness. Since the lack of robustness has become one of the main hurdles for CNNs, in this paper we ask: How to adversarially robustify a CNN-based MoE model? Can we robustly train it like an ordinary CNN model? Our pilot study shows that the conventional adversarial training (AT) mechanism (developed for vanilla CNNs) no longer remains effective to robustify an MoE-CNN. To better understand this phenomenon, we dissect the robustness of an MoE-CNN into two dimensions: Robustness of routers (i.e., gating functions to select data-specific experts) and robustness of experts (i.e., the router-guided pathways defined by the subnetworks of the backbone CNN). Our analyses show that routers and experts are hard to adapt to each other in the vanilla AT. Thus, we propose a new router-expert alternating Adversarial training framework for MoE, termed AdvMoE. *The effectiveness of our proposal is justified across 4 commonly-used CNN model architectures over 4 benchmark datasets. We find that* AdvMoE *achieves* $1\% \sim 4\%$ *adversarial robustness improvement over the original dense CNN, and enjoys the efficiency merit of sparsity-gated MoE, leading to more than* 50% *inference cost reduction. Codes are available at* https://github.com/OPTML-Group/Robust-MoE-CNN.

1. Introduction

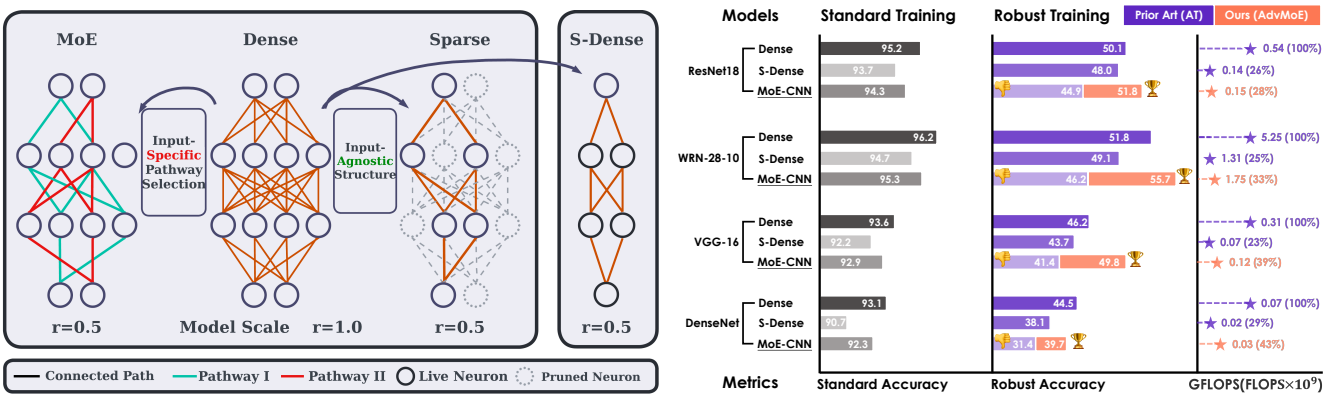
Despite the state-of-the-art performance achieved by the outrageously large networks [1–5] in various deep learning (DL) tasks, it still remains challenging to train and deploy such models cheaply. A major bottleneck is the lack of parameter efficiency [6]: A single data prediction only requires activating a small portion of the parameters of the full model. Towards efficient DL, sparse Mixture of Experts (MoE) [7–15] aims to divide and conquer the model parameters based on their optimal responses to specific inputs so that inference costs can be reduced. A typical MoE structure is comprised of a set of 'experts' (*i.e.*, sub-models extracted from the original backbone network) and 'routers' (*i.e.*, additional small-scale gating networks to determine expert selection schemes across layers). During inference, sparse MoE only activates the most relevant experts and forms the expert-guided pathway for a given input data. By doing so, sparse MoE can boost the inference efficiency (see 'GFLOPS' measurement in **Fig. 1**). Architecture-wise, sparse MoE has been used for both CNNs [8, 16] and vision transformers (ViTs) [7, 9–15, 17]. Yet, we will focus on the former since sparse MoE for CNNs is under-explored compared to non-sparse MoE for CNNs [18–20], and adversarial robustness (another key performance metric of our work) was extensively studied in the context of CNNs.

It is known that a main weakness of DL is the lack of adversarial robustness [21–23]. For example, CNNs can be easily fooled by adversarial attacks [21–23], in terms of tiny input perturbations generated to direct to erroneous predictions. Thus, adversarial training (**AT**) of CNNs has become a main research thrust [24–29]. However, when CNN meets sparse MoE, it remains elusive if the improved inference efficiency brought by the sparse MoE comes at the cost of more complex adversarial training recipes. Thus, we ask:

> ***(Q)*** *What will be the new insights into adversarial robustness of sparse MoE-integrated CNNs? And what will be the suited AT mechanism?*

To our best knowledge, problem **(Q)** remains open in the literature. The most relevant work to ours is [30], which investigated the adversarial robustness of MoE and lever-

Correspondence to: Yihua Zhang<zhan1908@msu.edu>



(a) Illustration of CNN types considered in this work.

(b) Performance overview on CIFAR-10.

Figure 1. (a) Model types (Dense, MoE-CNN, Sparse-CNN, and S(mall)-Dense) considered in this paper; see details in 'Model setup' of Sec. 3. (b) Performance overview using the standard training and the robust training on model architectures in (a), where standard accuracy and robust accuracy are defined by testing accuracy on the benign and adversarial test datasets, respectively. Compared to standard training (results in **gray**), the conventional AT [25] is no longer effective for MoE-CNN (see results in **light purple**). This is in contrast to AT for other CNN models (Dense and S-Dense). Different from AT, our proposed AdvMoE can effectively equip MoE-CNN with improved robustness, higher than Dense (see results in **orange**), without losing its inference efficiency (see "GFLOPS"). We refer readers to Sec. 5.1 for more experiment details.

aged the ordinary AT recipe [24] to defend against adversarial attacks. However, it only focused on the ViT architecture, making a vacancy for the research on robustification for the sparse MoE-based CNN (termed **MoE-CNN** in this work). Most importantly, we find that the vanilla AT [24, 25] (widely used to robustify CNNs) is *no longer* effective for MoE-CNN. Thus, new solutions are in demand.

To address **(Q)**, we need to (1) make careful sanity checks for AT in MoE-CNN, (2) make an in-depth analysis of its failure cases, and (3) advance new AT principles that can effectively improve robustness without losing the generalization and efficiency from sparse MoE. Specifically, our **contributions** are unfolded below.

- We dissect the MoE robustness into two new dimensions (different from CNNs): routers' robustness and experts' robustness. Such a robustness dissection brings novel insights into the (in)effectiveness of AT.
- Taking inspiration from the above robustness dissection, we propose a new Adversarial training framework for MoE, termed AdvMoE, which enforces routers and experts to make a concerted effort to improve the overall robustness of MoE-CNN.
- We conduct extensive experiments to demonstrate the effectiveness of AdvMoE across 4 CNN architectures and 4 datasets. For example, AdvMoE outperforms AT on the original dense CNN model (termed Dense) by a substantial margin: $1\% \sim 4\%$ adversarial robustness improvement and over 50% reduction of inference overhead; see **Fig. 1** for illustrations on different CNN types and highlighted performance achieved.

2. Related Work

Sparsely-activated Mixture of Experts (Sparse MoE). As a special instance of compostional neural architectures [31–33], MoE [4, 7–11, 16, 18–20, 34–41] aims at solving ML tasks in a divide-and-conquer fashion, which creates a series of sub-models (known as the *experts*) and conducts input-dependent predictions by combing the output of sub-models. As an important branch of MoE, sparsely gated MoE [4, 7–16, 39–43] only activates a subset of experts based on a routing system. The major advantage brought by sparse MoEs lies in its sub-linear increasing inference costs (FLOPs) with respect to (*w.r.t.*) model scales (parameter counts) [7]. In the vision domain, a vast majority of the existing works focus on the design of MoE for ViTs [9–15, 40, 42, 43], leaving MoE for CNNs under-explored [8, 16]. To our best knowledge, DeepMoE [8] is the most recent work that systematically studies the integration of MoE with CNNs, but restricts to the standard (non-robust) training paradigm. Meanwhile, there also exist other works related to MoE-CNN, but they either fall out of the "sparse" MoE scope [18, 20] or bring no efficiency gains [19]. By contrast, we focus on the *efficiency-promoting* MoE-CNN setup throughout this work.

Adversarial robustness. CNNs are notoriously vulnerable to imperceptible adversarial samples [22, 23, 44] and thus training adversarially robust models [21, 24, 45] has become a main research focus in many areas. Most of the robust training methods [24–29] are extended from min-max optimization-based adversarial training [46]. For instance, the work [25] seeks an optimal balance between

robustness and standard generalization ability. Other work [26, 28, 47–54] aims at trimming down the computational costs of robust training while maintaining robustness. The work [30] studies the robustness of MoE-based architectures for the first time. Yet, its focus stays on MoE for ViTs and the relationship between model capacity and robustness.

3. Problem Statement

In this section, we start by presenting the setup of MoE-CNN in this work and then introduce the robust learning paradigm. The lack of adversarial robustness of deep models inspires us to investigate whether the adversarial training (AT) approach designed for vanilla CNNs keeps effective for MoE-CNN. Through a motivating example, we show that the conventional AT recipe is *incapable* of equipping MoE-CNN with desired robustness. The resulting performance is even worse than that of AT-produced S-Dense, which has a much smaller model capacity than MoE-CNN. Thus, the question of how to robustify MoE-CNN arises.

Model setup. We consider a CNN-backboned MoE that consists of multiple MoE layers. Each MoE layer involves a router and a vanilla convolutional layer from the backbone CNN model. Within one MoE layer, we define N experts, each of which picks a subset of the channels from the convolutional layer. Specifically, suppose the l-th layer contains C_l channels, one expert will contain $r \times C_l$ channels, where we call the ratio $r \in [0, 1]$ *model scale* and keep it the same across different layers (see **Fig. 1a**). It is worth noting that as r increases, the per-expert model capacity increases (*i.e.*, with more parameters) at the cost of the efficiency reduction. In a forward path, the router first makes an *input-specific* expert selection. These selected layer-wise experts then form an end-to-end pathway to process this input. We use "*pathway*" to describe one experts-guided forward path (see **Fig. 1a**). We summarize the model setup in **Fig. A1**.

Further, we introduce different model types considered in this work and shown in **Fig. 1a**. First, we term the original dense CNN model '**Dense**', which serves as the *model basis* for other model types that derive from. Second, we directly shrink the channel number of each layer in Dense (based on the model scale parameter r) to obtain the 'small dense' model (termed '**S-Dense**'). Notably, S-Dense has the size *equivalent to a single pathway* in MoE-CNN. Third, we use the structured pruning method [50] to create a sparse subnetwork from Dense, with the weight remaining ratio same as the model scale parameter r in MoE-CNN, which we call '**Sparse-CNN**'. In summary, S-Dense has the smallest model capacity (comparable to a single pathway of MoE-CNN), and should provide the *performance lower-bound* for MoE-CNN. By contrast, Sparse-CNN has a larger model capacity but is smaller than MoE-CNN as it encodes a data-agnostic pathway of Dense, while MoE-CNN yields data-specific pathways at the same scale. Dense has the largest model capacity but the least inference efficiency.

Adversarial robustness: From CNN to MoE-CNN. It has been known that current machine learning models (*e.g.*, CNNs) are vulnerable to adversarial attacks [21–23]. Towards the robust design, a variety of AT (adversarial training) methods have been developed. The predominant ones include the min-max optimization-based vanilla AT [24] and its TRADES variant [25] that strikes a balance between generalization and adversarial robustness. Throughout the paper, we adopt TRADES as the default conventional AT recipe, which solves the following problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}} \left[\ell(\boldsymbol{\theta}; \mathbf{x}, y) + \frac{1}{\lambda} \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \ell_{\mathrm{KL}}(f_{\boldsymbol{\theta}}(\mathbf{x}), f_{\boldsymbol{\theta}}(\mathbf{x}+\boldsymbol{\delta})) \right] \quad \text{(AT)}$$

where $\boldsymbol{\theta}$ denotes model parameters to be robustified, $(\mathbf{x}, y) \in \mathcal{D}$ is a training sample, drawn from the training set $\mathcal{D}$, with input feature $\mathbf{x}$ and label y, $\ell(\boldsymbol{\theta}, \mathbf{x}; y)$ denotes the cross-entropy loss using model $\boldsymbol{\theta}$ at data point $(\mathbf{x}, y)$, $\boldsymbol{\delta}$ signifies the input perturbation variable subject to the ℓ_∞-norm ball of radius ϵ, $f_{\boldsymbol{\theta}}(\cdot)$ denotes the model's predictions, ℓ_{KL} is the KL divergence loss that characterizes the worst-case prediction stability at the presence of $\boldsymbol{\delta}$, and $\lambda > 0$ is a regularization parameter to strike the tradeoff between empirical risk minimization and the robustness of model predictions.

Although AT has been well studied for adversarial robustness of CNNs, there exists few attempts to robustify MoE-CNN. This raises the problem of our interest:

> **(Problem statement)** Can MoE-CNN be robustified as effectively as an ordinary CNN using AT? If not, how to robustly train MoE-CNN to achieve robustness not worse than AT-oriented S-Dense, Sparse-CNN, and Dense while preserving MoE's efficiency?

Warm-up study: AT for MoE-CNN is *not* trivial. Our goal to robustify MoE-CNN includes (1) achieving high robustness, (2) maintaining high prediction accuracy, and (3) making full use of MoE routing to keep the model's high efficiency and expressiveness. Nonetheless, the routing system in MoE brings extra robustification challenges, which never exist in ordinary CNNs. Specifically, the input-specific expert selection in MoE could make the attacker easier to succeed, since input perturbations can *either* mislead routers to select incorrect experts *or* fool the pathway-designated predictor. Such a '*two-way attack mode*' makes AT for MoE-CNN highly non-trivial.

Fig. 2 empirically justifies that the direct application of (AT) to MoE-CNN is problematic. In Fig. 2, we consider ResNet-18 as the model backbone (Dense) and `CIFAR-10` for image classification. We apply (AT) to train MoE-CNN

and S-Dense, and report the robust accuracy (RA), *i.e.*, test-time accuracy over adversarial examples generated by 50-step PGD attacks [24], against different attack strengths ϵ. As we can see, although MoE-CNN has a much larger model capacity than S-Dense, it leads to a significant RA drop when the conventional AT approach is applied. This implies that the design of AT for MoE-CNN is far from trivial. A new robust learning protocol is thus needed to improve the robustness of MoE-CNN without losing its merits in efficiency and generalization.

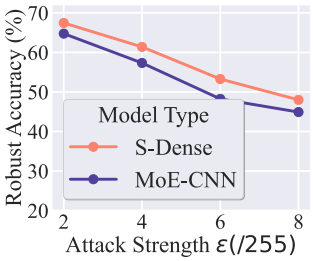


Figure 2. Performance of MoE-CNN and S-Dense robustly trained using (AT) on CIFAR-10 with ResNet-18 as the backbone.

4. Methods

In this section, we start by peering into the failure case of (AT) in MoE-CNN by understanding the roles of the routers and pathways in (AT). We empirically show that these individual components are hard to adapt to each other and cannot make a concerted effort in AT. Based on that, we develop a new AT framework for MoE-CNN, ADVMOE, which also takes inspiration from bi-level optimization.

Dissecting robustness of MoE-CNN: Routers' robustness vs. pathways' robustness. The main puzzle in robustifying MoE-CNN comes from the coupling between the robustness of routers (which are responsible for expert selection across layers) and the robustness of the input-specific MoE pathways (which are in charge of the final prediction of an input). Given the failure case of AT for MoE-CNN in Fig. 2, we need to understand the roles of routers and pathways in AT, *i.e.*, how the adversarial robustness of MoE-CNN is gained in the presence of the 'two-way attack mode'. To this end, we begin by assessing the influence of the routers' robustness on the overall robustness. This is also inspired by the recent pruning literature [50] showing that model robustness can be gained solely from network's sparse topology (regardless of model weights). We thus ask:

(Q1) Is improving routers' robustness sufficient to achieve a robust MoE-CNN?

To tackle **(Q1)**, we first split the parameters of MoE-CNN (*i.e.*, $\boldsymbol{\theta}$) into two parts, the parameters of routers $\boldsymbol{\phi}$ and the parameters of the backbone network $\boldsymbol{\psi}$. This yields $\boldsymbol{\theta} = [\boldsymbol{\phi}^\top, \boldsymbol{\psi}^\top]^\top$, where $\top$ is the transpose operation. We then call (AT) to robustly train routers ($\boldsymbol{\phi}$) but *fix* the backbone network ($\boldsymbol{\psi}$) at its standard pre-trained weights. We denote this partially-robustified model by $\bar{\boldsymbol{\theta}} = [\bar{\boldsymbol{\phi}}^\top, \boldsymbol{\psi}^\top]^\top$, where $\bar{}$ indicates the updated parameters. To answer **(Q1)**, we assess the robustness gain of $\bar{\boldsymbol{\theta}}$ vs. 3 baselines (M1-M3): (**M1**) the standard MoE-CNN $\boldsymbol{\theta}$, (**M2**) AT-robustified S-Dense, and (**M3**) Sparse-CNN achieved by the robust sparse mask learning method [50] over the original Dense model. **Fig. 3** shows the robust accuracy of the router-robustified MoE-CNN $\bar{\boldsymbol{\theta}}$ and its performance comparison with other baseline models. As we can see, the robustified router improves the overall robustness (*e.g.*, 37.64% for $\bar{\boldsymbol{\theta}}$ with model scale 0.5) compared to the undefended MoE-CNN (**M1**: 0%) and the robustified mask (**M3**: 20.04%). However, there is still a huge robustness gap compared to the (AT)-robustified S-Dense (**M2**: 47.68%). Based on the results above, we acquire the first insight into **(Q1)**:

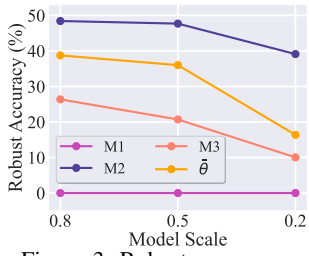


Figure 3. Robustness comparison of router-robusified MoE-CNN (*i.e.* $\bar{\boldsymbol{\theta}}$) and baseline models (M1 – M3) for different model scales under CIFAR-10 given the backbone network ResNet-18.

Insight 1: Robustifying routers improves the overall robustness of MoE-CNN but is *not* as effective as AT-resulted S-Dense.

Based on Insight 1, we further peer into the resilience of expert selection decisions to adversarial examples. If expert selections in *all* MoE layers keep intact in the presence of an adversarial perturbation, we say that the routing system of MoE-CNN is robust against this adversarial example. We then divide adversarial examples into **four categories** according to whether they successfully attacked routers and the router-oriented pathways: ❶ *unsuccessful* attack on *both* routers and MoE pathways, ❷ *successful* attack on routers but *not MoE* pathways, ❸ *successful* attack on MoE pathways but *not routers*, and ❹ *successful* attack on *both* routers and MoE pathways. Here ❶ + ❸ characterizes the robustness of routers, while ❶ + ❷ represents that of MoE. Thus, if ❷ or ❸ takes a large portion of generated adversarial examples, it implies that the routers' robustness does *not* directly impact the MoE pathway-based predictor's robustness. **Fig. 4** shows the above categories ❶-❹ when attacking the router-robustified MoE-CNN (*i.e.*, $\bar{\boldsymbol{\theta}}$). As we can see, routers' robustness indeed improves prediction robustness (as shown by 31.74% unsuccessful attacks against the MoE predictor in ❶). However, in the total number of unsuccessful attacks against routers (*i.e.*, ❶+❸= 76.27%), more than half of them successfully fool

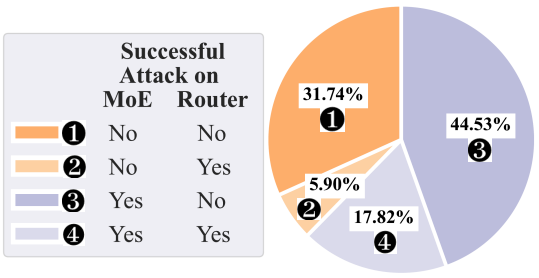


Figure 4. Adversarial attack success analysis on dissected MoE-CNN models $\bar{\boldsymbol{\theta}} = [\bar{\boldsymbol{\phi}}^\top, \boldsymbol{\psi}^\top]$ (model scale $r = 0.5$), where only $\bar{\boldsymbol{\phi}}$ is (AT)-robustified. The adversarial evaluation is based on 50-step PGD attack [24] to fool $\bar{\boldsymbol{\theta}}$, and other experiment setups align with Fig. 3. The evaluation is carried out on the test set with a total number of 10000 samples.

the MoE predictor (*i.e.*, ❸>❶). The above results provide us an additional insight:

Insight 2: Improving routers' robustness is *not* sufficient for the MoE predictor to gain satisfactory robustness although the former makes a positive impact.

Both **Insight 1** and **Insight 2** point out that only improving routers' robustness is *not* adequate to obtain the desired robustness for the overall MoE-CNN. Thus, we next ask:

(Q2) Given the router-robustified model $\bar{\boldsymbol{\theta}}$, can we equip $\bar{\boldsymbol{\theta}}$ with additional robustness by robustly training expert weights ($\boldsymbol{\psi}$)? And how does it further impact routers?

To answer **(Q2)**, we call (AT) to further robustly train the backbone network $\boldsymbol{\psi}$ on top of the router-robustified model $\bar{\boldsymbol{\theta}}$. We denote the resulting model by $\bar{\bar{\boldsymbol{\theta}}} = [\bar{\boldsymbol{\phi}}^\top, \bar{\boldsymbol{\psi}}^\top]$. **Fig. 5** shows the dissection of the robustness of $\bar{\bar{\boldsymbol{\theta}}}$ in the same setup of Fig. 4. Obviously, the overall prediction robustness (❶+❷) is further enhanced after updating $\bar{\boldsymbol{\theta}}$ to $\bar{\bar{\boldsymbol{\theta}}}$. Thus, the gains in the robustness of experts' weights indeed further help improve the overall robustness. However, this leads to a surprising drop in the router's robustness (❶+❸) when comparing $\bar{\bar{\boldsymbol{\theta}}}$ with $\bar{\boldsymbol{\theta}}$. This shows that routers' robustness is *not* automatically preserved if experts are updated. We obtain the following insight into **(Q2)**:

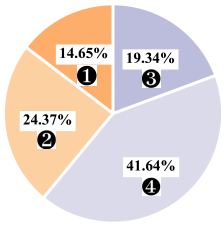


Figure 5. Adversarial attack success analysis on routers $\bar{\boldsymbol{\phi}}$ and MoE-CNN models $\bar{\bar{\boldsymbol{\theta}}} = [\bar{\boldsymbol{\phi}}^\top, \bar{\boldsymbol{\psi}}^\top]$. Other setups remain the same as Fig. 4.

Insight 3: Robustifying routers and MoE weights can yield complementary benefits but the inadaptability of routers' robustness to MoE's robustness prevents AT from achieving significant robustness improvement.

ADVMOE: Router-expert alternating AT through a bi-level optimization viewpoint. As illuminated by insights above, we provide a reason for the ineffectiveness of AT in robustifying MoE-CNN. **Insights 1-2** show that the robustness of routers ($\boldsymbol{\phi}$) and the robustness of MoE-based predictor ($\boldsymbol{\psi}$) are intertwined and their interrelation is non-trivial. As a result, the single-level (non-convex) robust optimization over the entire model parameters ($\boldsymbol{\phi}, \boldsymbol{\psi}$) experiences difficulty in co-optimizing routers and MoE prediction pathways to achieve the best complementary robustness gains, as supported by **Insight 3**. A key missing optimization factor in AT for MoE-CNN is its incapability of modeling and optimizing the coupling between the robustness of the routers and that of the MoE pathways. Without such an optimization design, it is difficult for AT to robustify routers and MoE pathways in a cooperative and adaptive mode.

Spurred by the above, we develop a new AT framework through bi-level optimization (**BLO**). In general, BLO provides a hierarchical learning framework with two levels of optimization tasks, where the objective and variables of an upper-level problem depend on the optimizer of the lower level. BLO then enables us to explicitly model the coupling between AT for routers and AT for MoE network. Specifically, we modify the conventional (AT) to

$$\begin{aligned} \underset{\boldsymbol{\psi}}{\text{minimize}} \quad & \ell_{\text{TRADES}}(\boldsymbol{\psi}, \boldsymbol{\phi}^*(\boldsymbol{\psi}); \mathcal{D}) \\ \text{subject to} \quad & \boldsymbol{\phi}^*(\boldsymbol{\psi}) = \arg\min_{\boldsymbol{\phi}} \ell_{\text{TRADES}}(\boldsymbol{\psi}, \boldsymbol{\phi}; \mathcal{D}), \end{aligned} \tag{1}$$

where the model parameters of MoE-CNN $\boldsymbol{\theta}$ are split into the lower-level optimization variables $\boldsymbol{\phi}$ for routers and the upper-level optimization variables $\boldsymbol{\psi}$ for the MoE backbone network, and $\ell_{\text{TRADES}}(\boldsymbol{\psi}, \boldsymbol{\phi}; \mathcal{D})$ denotes the TRADES-type training loss defined in (AT) by replacing $\boldsymbol{\theta}$ with $(\boldsymbol{\phi}, \boldsymbol{\psi})$. Compared to (AT), our proposal (1) has the following differences. **First**, robustifying MoE network ($\boldsymbol{\psi}$) is now explicitly coupled with routers' optimization through the lower-level solution $\boldsymbol{\phi}^*(\boldsymbol{\psi})$. **Second**, our proposal addresses the robustness adaptation problem pointed out in **Insight 3** since the lower-level optimization of (1) enables fast adaptation of $\boldsymbol{\phi}$ to the current MoE network $\boldsymbol{\psi}$ like meta-learning [55]. **Third**, since ℓ_{TRADES} is involved at both optimization levels of (1), the embedded attack generation problem (*i.e.*, maximization over $\boldsymbol{\delta}$ in AT) needs to be solved at each level but corresponding to different victim models, *i.e.*, $(\boldsymbol{\psi}, \boldsymbol{\phi})$ and $(\boldsymbol{\psi}, \boldsymbol{\phi}^*(\boldsymbol{\psi}))$, respectively.

To solve problem (1), we adopt a standard alternating optimization (AO) method [56]. Compared with other kinds of BLO algorithms [57], AO is the most computationally efficient. Our extensive experiments in Sec. 5 will show that AO is effective to boost the adversarial robustness of MoE-CNN and achieve improvements over baseline methods and models by a substantial margin. The key idea of AO is to alternatively optimize the lower-level and the upper-level problem, during which variables defined in another level

Algorithm 1 The AdvMoE algorithm

Initialize: backbone network ψ, routers ϕ, batch size b, attack generation step K.
for Iteration $t = 0, 1, \ldots,$ **do**
  Pick different random data batches $\mathcal{B}_\psi$ and $\mathcal{B}_\phi$ for backbone and router training
  Lower-level ϕ-update (with fixed ψ): Given ψ, update ϕ by minimizing ℓ_{TRADES} using K-step PGD attack [24] generator and SGD (with $\mathcal{B}_\phi$)
  Upper-level ψ-update (with fixed ϕ): Given ϕ, update ψ by minimizing ℓ_{TRADES} using K-step PGD attack generator and SGD (with $\mathcal{B}_\psi$)
end for

are fixed. We term the resulting algorithmic framework as Adversarially robust learning for MoE-CNN (AdvMoE); see Algorithm 1 for a summary.

We highlight that AdvMoE will train robust routers and robust MoE pathways to 'accommodate' each other. In contrast to the conventional AT framework, AdvMoE delivers the coupled $\phi^*(\psi)$ and ψ, where both parts make a concerted effort to improve the overall robustness. We also remark that AdvMoE does not introduce additional hyper-parameters, since in practice we found routers and experts can share the same learning rate and schedules. More implementation details are provided in Appendix B. In the meantime, we remark that since our proposal is a BLO with non-convex lower and upper-level objectives (1). It is difficult to prove the convergence of AdvMoE. Existing theoretical analysis of BLO typically relies on strongly convex assumptions of lower-level problems [58, 59]. Although without a proper theoretical analysis framework, our method converges well in practice (see Appendix C).

5. Experiments

In this section, we will demonstrate the effectiveness of our proposed AdvMoE approach on diverse datasets and models. We will also make an in-depth analysis of the router utility and the expert selection distribution for AdvMoE-trained MoE-CNN.

5.1. Experiment Setup

Model and dataset setups. To implement MoE-CNN and other baselines, we conduct experiments on ResNet-18 [60], Wide-ResNet-28-10 [61], VGG-16 [62], and DenseNet [63]. Towards fair assessment, our performance comparison between different model types is restricted to using the same model scale parameter r (see Fig. 1 for an example). By doing so, an input example will leverage the same amount of model parameters for decision-making. For MoE-CNN, we consider $N = 2$ experts with $r = 0.5$ by default, see Appendix B for more details. Dataset-wise, we focus on the commonly used ones to evaluate the adversarial robustness of image classification [24, 25, 64], including `CIFAR-10` [65], `CIFAR-100` [65], `TinyImageNet` [66], and `ImageNet` [66].

Baselines. To make our performance comparison informative and comprehensive, we consider three kinds of baselines that are fairly comparable to (AdvMoE). ① AT (S-Dense): we apply AT to S-Dense; ② AT (Sparse): we apply the robustness-aware (structured) sparse mask learning method [50] to obtain Sparse-CNN; ③ AT (MoE): we directly apply AT to MoE-CNN, which co-trains the routers and backbone network. Note this method is also adopted in the latest robust training algorithm [30] for ViT-based MoE architectures. It is worth noting that the above baselines use the same number of model parameters as the pathway of MoE-CNN during model prediction. In addition, we cover ④ AT (Dense) (applying AT to Dense) to acquire a robustness performance reference. Yet, we remark that it is *not* quite fair to directly compare Dense with the aforementioned smaller counterparts, since the former uses a larger model scale ($r = 1.0$) at test-time inference.

Training and evaluation. We use TRADES [25] as the default robust training objective for all baselines. We also follow the literature [24, 25, 27, 64] to set the attack strength by $\epsilon = 8/255$ for `CIFAR-10` and `CIFAR-100`, and $\epsilon = 2/255$ for `TinyImageNet` and `ImageNet`. To implement AdvMoE (Algorithm 1), we mimic the TRADES training pipeline but conduct the proposed BLO routine to robustify routers and backbone parameters in an interactive mode. We adopt 2-step PGD attack [24] at training time for *all* the methods, supported by the recent work [67] showing its compelling performance in AT. We refer readers to Appendix B for more training details. During evaluation, we report standard accuracy (**SA**) on the clean test dataset and robust accuracy (**RA**) against test-time 50-step PGD attacks [24] with the attack strength same as the training values. We also report **GFLOPs** (FLOPS $\times 10^9$) as an indicator of the test-time inference efficiency.

5.2. Experiment Results

Overall performance. **Tab. 1** presents the overall performance of our proposed AdvMoE algorithm vs. baselines. We make several key observations below.

First, AdvMoE yields a significant robustness enhancement over all the baselines in every data-model setup. Specifically, AdvMoE consistently yields an improvement of around $1\% \sim 5\%$ on the robustness measured by RA against PGD attacks. Notably, AdvMoE can also outperform • AT (Dense) in most cases, around $1\% \sim 4\%$ robustness improvement (see highlighted results in green). This is remarkable since Dense ($r = 1.0$) is twice larger

Table 1. Performance overview of ADVMOE (our proposal) *vs.* baselines on various datasets and model backbone architectures. The model scale is fixed at $r = 0.5$ for Dense-CNN, Sparse-CNN and Moe-CNN (denoted with the symbol ○, since they are fairly comparable to each other) compared to Dense ($r = 1.0$, denoted with the symbol ●). For train- and test-time attack generations, we adopt an attack strength of $\epsilon = 8/255$ for `CIFAR-10` and `CIFAR-100`, and $\epsilon = 2/255$ for `TinyImageNet` and `ImageNet`. We evaluate RA (robust test accuracy) against 50-step PGD attack [24], SA (standard test accuracy), and GFLOPS (FLOPS $\times 10^9$) per test-time example (test-time inference efficiency) for each model. In each (dataset, backbone) setup, ① we highlight the best SA and RA over all baselines per model scale in **bold**, and ② we mark the performance better than AT (Dense) in green. Results in the format of $a \pm b$ provide the mean value a and its standard deviation b over 3 independent trials.

Method	Backbone	RA (%)	SA (%)	GFLOPS(#)	Method	Backbone	RA (%)	SA (%)	GFLOPS (#)
CIFAR-10									
● AT (Dense)	ResNet-18	50.13±0.13	82.99±0.11	0.54	● AT (Dense)	WRN-28-10	51.75±0.12	83.54±0.15	5.25
○ AT (S-Dense)		48.12±0.09	80.18±0.11	0.14 (74%↓)	○ AT (S-Dense)		50.66±0.13	82.24±0.10	1.31 (75%↓)
○ AT (Sparse)		47.93±0.17	**80.45**±0.13	0.14 (74%↓)	○ AT (Sparse)		48.95±0.14	82.44±0.17	1.31 (75%↓)
○ AT (MoE)		45.57±0.51	78.84±0.75	0.15 (72%↓)	○ AT (MoE)		46.73±0.46	77.42±0.73	1.75 (67%↓)
○ **ADVMOE**		**51.83** ±0.12	80.15±0.11	0.15 (72%↓)	○ **ADVMOE**		**55.73** ±0.13	**84.32** ±0.18	1.75 (67%↓)
● AT (Dense)	VGG-16	46.19±0.21	82.18±0.23	0.31	● AT (Dense)	DenseNet	44.52±0.14	74.97±0.19	0.07
○ AT (S-Dense)		45.72±0.18	**80.10**±0.16	0.07 (77%↓)	○ AT (S-Dense)		38.07±0.13	69.63±0.11	0.02 (71%↓)
○ AT (Sparse)		46.13±0.15	79.32±0.18	0.07 (77%↓)	○ AT (Sparse)		37.73±0.13	67.35±0.12	0.02 (71%↓)
○ AT (MoE)		43.37±0.46	76.49±0.65	0.12 (61%↓)	○ AT (MoE)		35.21±0.74	64.41±0.81	0.03 (57%↓)
○ **ADVMOE**		**49.82** ±0.11	80.03±0.10	0.12 (61%↓)	○ **ADVMOE**		**39.97**±0.11	**70.13**±0.15	0.03 (57%↓)
CIFAR-100									
● AT (Dense)	ResNet-18	27.23±0.08	58.21±0.12	0.54	● AT (Dense)	WRN-28-10	27.90±0.13	57.60±0.09	5.25
○ AT (S-Dense)		26.41±0.16	57.02±0.14	0.14 (74%↓)	○ AT (S-Dense)		26.30±0.10	56.80±0.08	1.31 (75%↓)
○ AT (Sparse)		26.13±0.14	57.24±0.12	0.14 (74%↓)	○ AT (Sparse)		25.83±0.16	57.39±0.14	1.31 (75%↓)
○ AT (MoE)		22.72±0.42	53.34±0.61	0.15 (72%↓)	○ AT (MoE)		22.94±0.55	53.39±0.49	1.75 (67%↓)
○ **ADVMOE**		**28.05** ±0.13	**57.73**±0.11	0.15 (72%↓)	○ **ADVMOE**		**28.82** ±0.14	**57.56**±0.17	1.75 (67%↓)
● AT (Dense)	VGG-16	22.37±0.15	52.36±0.17	0.31	● AT (Dense)	DenseNet	21.72±0.13	48.64±0.14	0.07
○ AT (S-Dense)		20.58±0.13	**48.89**±0.14	0.07 (77%↓)	○ AT (S-Dense)		16.86±0.21	39.97±0.11	0.02 (71%↓)
○ AT (Sparse)		21.12±0.22	48.03±0.17	0.07 (77%↓)	○ AT (Sparse)		17.72±0.14	41.03±0.16	0.02 (71%↓)
○ AT (MoE)		19.34±0.43	45.51±0.75	0.12 (61%↓)	○ AT (MoE)		14.45±0.45	36.72±0.71	0.03 (57%↓)
○ **ADVMOE**		**21.21**±0.21	48.33±0.17	0.12 (61%↓)	○ **ADVMOE**		**23.31** ±0.11	**48.97** ±0.14	0.03 (57%↓)
Tiny-ImageNet									
● AT (Dense)	ResNet-18	38.17±0.14	53.81±0.16	2.23	● AT (Dense)	WRN-28-10	38.82±0.15	55.30±0.19	21.0
○ AT (S-Dense)		36.29±0.16	52.15±0.13	0.55 (75%↓)	○ AT (S-Dense)		37.09±0.12	54.83±0.16	5.26 (75%↓)
○ AT (Sparse)		36.11±0.13	50.75±0.17	0.55 (75%↓)	○ AT (Sparse)		37.32±0.14	54.32±0.23	5.26 (75%↓)
○ AT (MoE)		34.41±0.31	47.73±0.41	0.75 (68%↓)	○ AT (MoE)		33.31±0.41	49.91±0.52	7.44 (65%↓)
○ **ADVMOE**		**39.99** ±0.12	**53.31**±0.14	0.75 (68%↓)	○ **ADVMOE**		**40.15** ±0.15	**55.18**±0.09	7.44 (65%↓)
ImageNet									
● AT (Dense)	ResNet-18	44.64±0.14	60.32±0.15	1.82	● AT (Dense)	WRN-28-10	45.13±0.14	60.97±0.16	16.1
○ AT (S-Dense)		41.19±0.16	58.32±0.12	0.48 (74%↓)	○ AT (S-Dense)		41.72±0.15	58.98±0.18	4.04 (75%↓)
○ AT (Sparse)		40.87±0.15	58.22±0.13	0.48 (74%↓)	○ AT (Sparse)		39.88±0.18	**59.21**±0.14	4.04 (75%↓)
○ AT (MoE)		35.57±0.73	55.47±0.66	0.67 (63%↓)	○ AT (MoE)		37.42±0.44	56.44±0.71	5.15 (68%↓)
○ **ADVMOE**		**43.32** ±0.12	**59.72**±0.17	0.67 (63%↓)	○ **ADVMOE**		**46.82** ±0.11	58.87±0.07	5.15 (68%↓)

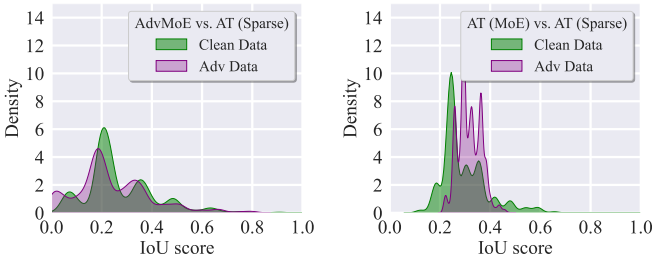
than an MoE pathway ($r = 0.5$). **Second**, we observe that ADVMOE has a preference on wider models. For instance, when WRN-28-10 (the widest model architecture in experiments) is used, ADVMOE yields better robustness over the Dense counterpart across all the dataset setups. **Third**, we also observe that the direct AT application to MoE-CNN, *i.e.*, AT (MoE), is worse than AT (S-Dense) and ADVMOE in all setups. This is consistent with our findings in Sec. 4. We remark that although the usefulness of AT (MoE) was exploited in [30] for the MoE-type ViT, it is *not* effective for training MoE-type CNNs anymore. **Fourth**, ADVMOE can retain the high inference efficiency for MoE-CNN, as evidenced by the GFLOPS measurements in Tab. 1. Compared to S-Dense, MoE-CNN introduces minor computational overhead due to the routing system. However, it saves more than 50% of the inference cost vs. Dense. This implies that our proposal ADVMOE can preserve the efficiency merit of the MoE structure while effectively improving its adversarial robustness.

Robust evaluation on AutoAttack [68]. In Tab. 2, we provide additional experiments evaluated by AutoAttack [68] (termed **RA-AA**), a popular robustness evaluation benchmark [69]. The experiment setting in Tab. 2 follows **Tab. 1**. We report RA-AA on `CIFAR-10` and `CIFAR-100` with ResNet-18 and WRN-28-10. As we can see, although AutoAttack leads to a lower RA-AA compared to RA evaluated using PGD attacks (termed

Table 2. Robustness overview evaluated with AutoAttack [68] (**RA-AA**) on various datasets and model backbone architectures. Other settings strictly follow Tab. 1. The values of RA-PGD, SA, and GFLOPS are repeated from Tab. 1 for better comparison.

Method	Backbone	RA-PGD (%)	RA-AA (%)	SA (%)	GFLOPS(#)	Method	Backbone	RA-PGD (%)	RA-AA (%)	SA (%)	GFLOPS (#)
CIFAR-10											
● AT (Dense)	ResNet-18	50.13±0.13	44.72±0.15	82.99±0.11	0.54	● AT (Dense)	WRN-28-10	51.75±0.12	45.13±0.12	83.54±0.15	5.25
○ AT (S-Dense)		48.12±0.09	42.24±0.13	80.18±0.11	0.14 (74%↓)	○ AT (S-Dense)		50.66±0.13	44.14±0.10	82.24±0.10	1.31 (75%↓)
○ AT (Sparse)		47.93±0.17	42.11±0.11	**80.45**±0.13	0.14 (74%↓)	○ AT (Sparse)		48.95±0.14	43.97±0.11	82.44±0.17	1.31 (75%↓)
○ AT (MoE)		45.57±0.51	40.42±0.19	78.84±0.75	0.15 (72%↓)	○ AT (MoE)		46.73±0.46	41.11±0.23	77.42±0.73	1.75 (67%↓)
○ **AdvMoE**		**51.83** ±0.12	**45.13** ±0.07	80.15±0.11	0.15 (72%↓)	○ **AdvMoE**		**55.73** ±0.13	**45.89** ±0.11	**84.32** ±0.18	1.75 (67%↓)
CIFAR-100											
● AT (Dense)	ResNet-18	27.23±0.08	23.11±0.06	58.21±0.12	0.54	● AT (Dense)	WRN-28-10	27.90±0.13	23.45±0.11	57.60±0.09	5.25
○ AT (S-Dense)		26.41±0.16	22.11±0.13	57.02±0.14	0.14 (74%↓)	○ AT (S-Dense)		26.30±0.10	22.23±0.13	56.80±0.08	1.31 (75%↓)
○ AT (Sparse)		26.13±0.14	21.89±0.11	57.24±0.12	0.14 (74%↓)	○ AT (Sparse)		25.83±0.16	21.97±0.09	57.39±0.14	1.31 (75%↓)
○ AT (MoE)		22.72±0.42	16.33±0.25	53.34±0.61	0.15 (72%↓)	○ AT (MoE)		22.94±0.55	17.87±0.24	53.39±0.49	1.75 (67%↓)
○ **AdvMoE**		**28.05** ±0.13	**23.33** ±0.06	**57.73**±0.11	0.15 (72%↓)	○ **AdvMoE**		**28.82** ±0.14	**23.57** ±0.12	**57.56**±0.17	1.75 (67%↓)

RA-PGD), AdvMoE still outperforms AT (S-Dense), AT (Sparse), and AT (MoE) consistently, evidenced by the **bold** numbers in the RA-AA columns.



(a) AdvMoE (b) AT (MoE)

Figure 6. The distribution of the intersection of union (IoU) scores of the input-specific pathways generated by AdvMoE (a) and AT (MoE) (b) vs. the static mask found by AT (Sparse). The distribution over the clean test set and the adversarial test set is plotted for AT (MoE) and AdvMoE on setting (ResNet-18, CIFAR-100). Other settings are aligned with Tab.1.

MoE-CNN trained by AdvMoE enjoys better router utility. Based on the results above and the preliminary studies in Sec. 4, we next peer into the performance difference achieved by AT (Sparse), AT (MoE), and AdvMoE from the perspective of pathway diversities. We ask:

① What is the relationship between the dynamic pathways generated by the routers trained by AdvMoE and the static mask optimized by AT (Sparse)? ② What is the difference between the routing decisions using AdvMoE and AT (MoE), and how does it impact the performance?

Regarding ①, we investigate the cosine similarity between the pathways generated by training methods, either AT (MoE) or AdvMoE, and the static mask found by AT (Sparse). Since the latter can be regarded as a single pathway used for all the data, we term it '*mask pathway*' in contrast to '*MoE pathway*'. We calculate the intersection of union (**IoU**) score between the MoE pathway and the mask pathway under each testing dataset (the clean or adversarial version). **Fig. 6** presents the IoU distributions based on the clean and adversarial test datasets (**Fig. 6a** for AdvMoE and **Fig. 6b** for AT (MoE)). We remark that a smaller IoU score indicates a larger discrepancy between the MoE pathway and the mask pathway. As we can see, the IoU distribution of AdvMoE vs. AT (Sparse) in Fig. 6a shifts closer to 0 compared with Fig. 6b. This observation applies to both standard and adversarial evaluation and suggests that AdvMoE (our proposal) has a better capability than AT (MoE) to re-build input-specific MoE pathways, which are more significantly different from the input-agnostic mask pathway identified by the pruning-based method, AT (Sparse).

Regarding ②, we observe from Fig. 6 that the routers learned by AT (MoE) are more fragile to adversarial attacks compared to AdvMoE, as evidenced by the less intersection area of adversarial data vs. clean data. This is also aligned with **Insight 3** in Sec.4. Moreover, the routing policy learned by AdvMoE is more diverse than AT (MoE), as indicated by the latter's density-concentrated IoU scores. In contrast, the distribution of AdvMoE is dispersed with a smaller peak value. Therefore, regarding the expert utility, AdvMoE is able to assign the inputs to a larger group of pathways than AT (MoE), making better use of experts.

A coupling effect of expert number N and per-expert model scale r on AdvMoE. Recall that there exist two key parameters involved in MoE-CNN (**Fig. A1**): *(a)* the number of experts N, and *(b)* the model scale r that defines the per-expert (or per-pathway) model capacity. Given the backbone model (*e.g.*, ResNet-18 in this experiment), a larger N paired with a small r implies that each expert may only have limited model capacity, *i.e.*, corresponding to a less number of channels. Regardless of N, if $r = 1$, the full backbone network will be used to form the identical decision pathway.

Fig. 7 shows the RA of MoE-CNN trained by AdvMoE vs. the model scale parameter r at different values of N. Two insightful observations can be drawn. **First**, there exists an MoE regime (*e.g.*, $N < 8$ and $r \in [0.5, 0.9]$), in which AdvMoE can outperform AT (Dense) (*i.e.*, $r = 1$) by a substantial margin. This shows the benefit of MoE in adversarial robustness. However, if the number of experts becomes larger (*e.g.*, $N = 10$), the increasing diversity of MoE pathways can raise the difficulty of routers' robustification and thus hampers the performance of AdvMoE (see $N = 10$ and $r = 0.8$ in **Fig. 7**). **Second**, there exists an

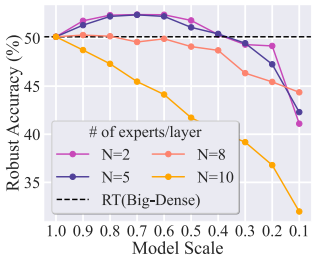


Figure 7. Performance of ADVMOE under CIFAR-10 using ResNet-18 as the backbone network for different values of expert number N and model scale r. The black dash line denotes the performance of Dense (*i.e.* $r = 1$).

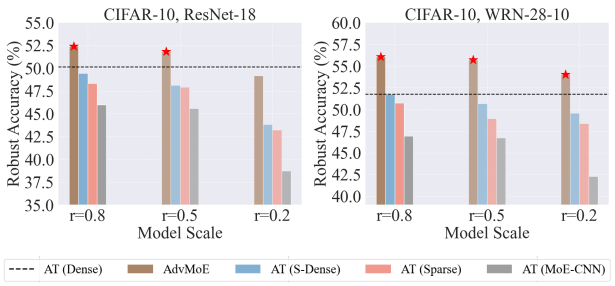


Figure 8. Robustness comparison of models trained with different methods under various model scale settings. Results higher than that of AT (Dense) are marked with ⋆. Other setups are aligned with Tab. 1. Please refer to Appendix C for exact numbers and GFLOPS comparisons.

ineffective MoE regime (*e.g.*, $N \geq 8$ and $r < 0.5$), in which the performance of ADVMOE largely deviates from that of AT (Dense). In this regime, each expert consists only of a small number of channels, which restricts its robust training ability. Accordingly, both the increasing diversity of MoE pathways (large N) and the limited capacity per pathway (small r) could impose the difficulties of AT for MoE-CNN. In our experiments, we choose $r = 0.5$ and $N = 2$, which preserves the diversity of MoE pathways (*i.e.*, inference efficiency) and retains the effectiveness of robust training.

Performance with different model scales. To make sure the observations and conclusions from Tab. 1 are consistent across different values of the model scale parameter r, we repeated the experiments on (CIFAR-10, ResNet-18) and (CIFAR-10, WRN-28-10) using $r \in \{0.2, 0.5, 0.8\}$ to cover the {sparse, medium, dense} regimes with respect to Dense ($r = 1.0$). **Fig. 8** summarizes the obtained experiment results. As we can see, ADVMOE yields consistent robustness improvements over all the baselines, including Dense. And the improvement rises as the model scale r increases. This is not surprising as more parameters will be used when processing one input. Yet, a clear drawback brought by the larger model scale r is the increase of inference cost, evidenced by the GFLOPS numbers. When r turns to be large (like $r = 0.8$), the efficiency benefit brought by the pathway sparsification from MoE gradually vanishes. Thus, a medium sparsity ($r = 0.5$) is a better choice to balance the trade-off between performance and efficiency, which is thus adopted as our default setting.

Table 3. Performance on robust training for MoE-ViT with in the setup (ImageNet, DeiT-Tiny). Other settings follow Tab. 1.

Method	RA (%)	SA (%)	GFLOPS (#)
SOTA[30]	44.63	61.72	0.27
ADVMOE	**45.93**	61.67	0.27

Extended study: ADVMOE for ViT. To explore the capability of our proposal ADVMOE on ViT-based MoE models (MoE-ViT), **Tab. 3** presents additional results following the recently published SOTA baseline [30] for MoE-ViT. As we can see, ADVMOE is also applicable to MoE-ViT and can boost robustness over the SOTA baseline by over 1% RA improvement, while achieving a similar level of SA. Thus, although our work focuses on robust training for MoE-CNN, it has the promise of algorithmic generality to other MoE-based architectures. We defer a more comprehensive study in the future.

Additional experiments. We conduct ablation studies on (1) robustness evaluation using AutoAttack [68] (consistent findings can be drawn as PGD attacks), (2) attacks steps used in AT, and (3) additional explorations towards the coupling effect between the number of experts and the model scales. We refer readers to Appendix C for detailed results.

6. Conclusion

In this work, we design an effective robust training scheme for MoE-CNN. We first present several key insights on the defense mechanism of MoE-CNN by dissecting adversarial robustness through the lens of routers and pathways. We next propose ADVMOE, the first robust training framework for MoE-CNN via bi-level optimization, robustifying routers and pathways in a cooperative and adaptive mode. Finally, extensive experiments demonstrate the effectiveness of ADVMOE in a variety of data-model setups. Meanwhile, we admit that the ADVMOE requires roughly twice the computational capacity compared to the vanilla AT baseline due to alternating optimization that calls two back-propagations per step. Addressing this efficiency concern presents a meaningful avenue for future work.

Acknowledgement

The work of Y. Zhang, S. Chang and S. Liu was partially supported by National Science Foundation (NSF) Grant IIS-2207052 and Cisco Research Award. The work of Z. Wang is in part supported by the US Army Research Office Young Investigator Award (W911NF2010240).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. 2

[5] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[6] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*, 2021. 1

[7] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 1, 2

[8] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020. 1, 2

[9] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 1, 2

[10] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

[11] Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. Go wider instead of deeper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8779–8787, 2022. 2

[12] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*, 2022.

[13] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. *arXiv preprint arXiv:2202.09368*, 2022.

[14] Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. *arXiv preprint arXiv:2206.04046*, 2022.

[15] Tianlong Chen, Zhenyu Zhang, AJAY KUMAR JAISWAL, Shiwei Liu, and Zhangyang Wang. Sparse moe with random routing as the new dropout: Training bigger and self-scalable models. In *International Conference on Learning Representations*, 2023. 1, 2

[16] Sam Gross, Marc'Aurelio Ranzato, and Arthur Szlam. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6865–6873, 2017. 1, 2

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[18] Alhabib Abbas and Yiannis Andreopoulos. Biased mixtures of experts: Enabling computer vision inference under data transfer limitations. *IEEE Transactions on Image Processing*, 29:7656–7667, 2020. 1, 2

[19] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, pages 516–532. Springer, 2016. 2

[20] Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner. Using mixture of expert models to gain insights into semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 342–343, 2020. 1, 2

[21] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 3

[22] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on S&P*, 2017. 2

[23] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016. 1, 2, 3

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 3, 4, 5, 6, 7

[25] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *ICML*, 2019. 2, 3, 6

[26] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019. 3

[27] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. 6

[28] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. *arXiv preprint arXiv:1905.00877*, 2019. 3

[29] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. 1, 2

[30] Joan Puigcerver, Rodolphe Jenatton, Carlos Riquelme, Pranjal Awasthi, and Srinadh Bhojanapalli. On the adversarial robustness of mixture of experts. *arXiv preprint arXiv:2210.10253*, 2022. 1, 3, 6, 7, 9

[31] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. In *Advances in Neural Information Processing Systems*, 2022. 2

[32] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, 2022.

[33] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *European Conference on Computer Vision*, 2022. 2

[34] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012. 2

[35] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[36] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[37] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[38] Ke Chen, Lei Xu, and Huisheng Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252, 1999.

[39] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pages 6265–6274. PMLR, 2021. 2

[40] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2

[41] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022. 2

[42] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022. 2

[43] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *arXiv preprint arXiv:2206.02770*, 2022. 2

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[45] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2

[46] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 2

[47] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pages 11278–11287. PMLR, 2020. 3

[48] Yihua Zhang, Yuguang Yao, Parikshit Ram, Pu Zhao, Tianlong Chen, Mingyi Hong, Yanzhi Wang, and Sijia Liu. Advancing model pruning via bi-level optimization. *arXiv preprint arXiv:2210.04092*, 2022. 13

[49] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. In *Advances in Neural Information Processing Systems*, pages 1283–1294, 2019.

[50] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 4, 6, 13

[51] Yonggan Fu, Qixuan Yu, Yang Zhang, Shang Wu, Xu Ouyang, David Cox, and Yingyan Lin. Drawing robust scratch tickets: Subnetworks with inborn robustness are found within randomly initialized networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[52] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *NeurIPS*, 2020.

[53] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *ICLR*, volume 1, 2021.

[54] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021. 3

[55] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 5

[56] James C Bezdek and Richard J Hathaway. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368, 2003. 5

[57] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv preprint arXiv:2101.11517*, 2021. 5

[58] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020. 6

[59] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. *arXiv preprint arXiv:2308.00788*, 2023. 6

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 13

[61] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 6, 13

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6, 13

[63] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6, 13

[64] Yihua Zhang, Guanhua Zhang, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing adversarial training through the lens of bi-level optimization, submitted to NeurIPS, 2021. 6

[65] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 6

[66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 6

[67] Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pages 26693–26712. PMLR, 2022. 6

[68] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. 7, 8, 9

[69] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020. 7

[70] Xiang Deng and Zhongfei Mark Zhang. Is the meta-learning idea able to improve the generalization of deep neural networks on the standard supervised learning? In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 150–157. IEEE, 2021. 13

[71] Resnet implementation in pytorch. https://github.com/kuangliu/pytorch-cifar/blob/master/models/resnet.py. 13