On the Adversarial Robustness of Mixture of Experts

Joan Puigcerver
Google Research

Rodolphe Jenatton
Google Research

Carlos Riquelme
Google Research

Pranjal Awasthi
Google Research

Srinadh Bhojanapalli
Google Research

Abstract

Adversarial robustness is a key desirable property of neural networks. It has been empirically shown to be affected by their sizes, with larger networks being typically more robust. Recently, Bubeck and Sellke [3] proved a lower bound on the Lipschitz constant of functions that fit the training data in terms of their number of parameters. This raises an interesting open question, do—and can—functions with more parameters, but not necessarily more computational cost, have better robustness? We study this question for sparse Mixture of Expert models (MoEs), that make it possible to scale up the model size for a roughly constant computational cost. We theoretically show that under certain conditions on the routing and the structure of the data, MoEs can have significantly smaller Lipschitz constants than their dense counterparts. The robustness of MoEs can suffer when the highest weighted experts for an input implement sufficiently different functions. We next empirically evaluate the robustness of MoEs on ImageNet using adversarial attacks and show they are indeed more robust than dense models with the same computational cost. We make key observations showing the robustness of MoEs to the choice of experts, highlighting the redundancy of experts in models trained in practice.

1 Introduction

Adversarial robustness refers to prediction robustness of a given machine learning model to adversarial, but bounded, changes to the input. Neural networks trained with standard classification objectives have been shown to have poor adversarial robustness [25], a property attributed to their overparametrization. Conversely, in practice, larger models, that have more parameters and higher computation cost, have shown better robustness [20, 26, 1, 12].

In a recent work, Bubeck and Sellke [3] studied this phenomenon from a theoretical perspective, by analyzing the relationship between model size and its Lipschitz constant - which measures the sensitivity of a function to changes in the input. In particular the authors proved that any function with P parameters, that memorizes N input data points in D dimensions, has a Lipschitz constant of at least $\mathcal{O}(\sqrt{\frac{N \cdot D}{P}})$. This shows that, on a given dataset, larger models can have better robustness (smaller Lipschitz constant). Note that as this is only a lower bound, it does not guarantee that larger models will indeed have a smaller Lipschitz constant. Interestingly, the result is agnostic to other properties of the function, such as its computation cost, or specific architecture.

Given the above result it is natural to wonder: can one increase the model size, without increasing the computation cost, and enjoy better robustness? An increasingly popular class of such models are Mixture of Experts (MoE) [23]. MoE models have multiple expert sub-models, and a routing function that selects for each input a small subset of the experts and routes the input only to them. Neural networks with MoE layers, such as Switch Transformer [10] in NLP, and V-MoE [22] in Computer Vision, have been shown to achieve superior performance in comparison to their dense counterparts,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

by allowing one to scale the model size without increasing the computation cost. In this paper we study the following question: are MoE models more adversarially robust than dense models?

In general, sparse MoE models are not continuous, and hence not even smooth. Small changes to the input can result in the router selecting a different expert for a given example. Unfortunately, in some cases these experts may be very different, resulting in large changes to the output. Another factor that affects the robustness of MoE models is the geometry of input data and its routing –how the data is divided between experts. This decides what data an expert is trained on and hence their robustness. Finally, it has been observed that –unless certain auxiliary losses are also applied to encourage balancedness– the number of used experts tends to collapse to very few, and the remaining ones are just ignored [22, 10]. Given these stability issues, it is a priori *not* clear if MoEs, despite having more parameters, are more robust than dense models.

To theoretically study the effect of both of these factors, (1) router stability and (2) data routing, on robustness we consider the *smooth* MoE models, where each expert output is weighted by their routing probability. In particular, MoE models where all experts are simultaneously applied to every input. We analyze models with fixed routing proving that MoEs with *linear experts* can achieve better robustness if the data is well separated and routed accordingly. In the extreme case, MoEs with E experts can have a smaller Lipschitz constant by a factor of $1/\sqrt{E}$ compared to equivalent dense models. We also characterize the effect of the difference between experts in terms of robustness, showing that MoEs can have a high Lipschitz constant when two experts are very different for inputs that weigh the experts similarly and highly. We show that both these factors, data routing and difference between relevant experts, characterize the Lipschitz constant of MoE models.

We next evaluate robustness experimentally by using adversarial attacks [20]. We experimentally show that on the ImageNet dataset [6], MoEs are more robust than dense models that have the same computation cost. Interestingly, we observe that adversarial attacks result in significant routing changes of the inputs, but do not result in lower robustness than dense models. This suggests that in practice, MoEs are robust to the choice of the experts to an extent. We next perform standard adversarial training of both the dense and the MoE models and again observe that MoEs are more robust against adversarial attacks.

The main contributions of the current work are as follows:

1. We propose a simple theoretical framework to understand the robustness of Mixture of Experts (MoE) models. We provide general and non-trivial sufficient conditions under which MoEs are provably more robust than their dense counterparts for the linear experts setting.
2. We perform extensive experiments to demonstrate that in practice MoEs indeed enjoy better robustness than dense models to norm bounded adversarial attacks. We also uncover intriguing properties of adversarial attacks for MoE models and show that even for robust MoE models, the attacks very often change the routing of data points, thereby pointing to a high degree of redundancy in such models.

2 Preliminaries

2.1 Mixture of Experts

Mixture of Experts combine the outputs of (sub-)models, i.e., the *experts*, via a weighted sum of their outputs [14, 15, 28, 9]. *Sparse* Mixture of Experts condition the weights of the sum in the inputs, and activate only K (out of E) experts, where K is typically a very small integer compared to E (such as $K = 1$ or $K = 2$) [23, 10]. This form of conditional computation models allows one to easily increase the number of parameters in the model (roughly) independently of its compute cost [21]. This approach has been recently applied to significantly increase the model size and quality of models used in Natural Language Processing [23, 18, 19, 10, 8] and Computer Vision applications [22, 27].

More concretely, we define the sparse and smooth (or dense) versions of mixtures of experts below. While previous empirical works have sometimes selected more than one expert per input ($K > 1$), for simplicity we constrain ourselves to the case where only one expert is selected for each input ($K = 1$). Recently, practical models also tend to this setup to match the FLOPs of dense models [10].

Sparse MoEs. In these models only one expert is selected for a given example $\boldsymbol{x}$ based on its routing probabilities $p_i(\boldsymbol{x})$. These models are not continuous functions of the input. A small change in the input can lead to large changes in the output, due to changes in selected experts. A sparse MoE layer

is defined as:

$$f(\boldsymbol{x}) = \sum_{i=1}^{E} \mathbf{1}_{\{p_i(\boldsymbol{x}) \geq p_j(\boldsymbol{x}), \forall j \neq i\}} \; f_i(\boldsymbol{x}). \tag{1}$$

Here $f_i(\boldsymbol{x})$ are the individual expert functions. $\mathbf{1}$ is the indicator function that takes the value of 1 if the condition is satisfied and 0 else. Note that to do tie breaking in the above definition, in case multiple experts have the same maximum probability, we simply sample uniformly one of the experts with the maximum probability. In practice, MoE models can have multiple sparse layers in combination with dense layers. Further, random noise is sometimes added to the p_i's [23, 22].

We refer to MoEs with only 1 expert as **dense** models. In dense models, which are the standard in neural networks literature, same function is applied to all input examples. In sparse MoEs introduced above, different part of network / expert is applied to each input example depending on which expert selected. This flexibility allows us to train larger sparse MoEs, that have the same computational cost as the dense models, as only a part of MoE is activated/selected for each input, with better accuracy than dense models [10, 22].

Smooth MoEs. To theoretically analyze robustness, we consider a smooth mixture of expert models, as they are continuous functions. In this case, the output of each expert f_i, is weighted according to its routing probabilities:

$$f(\boldsymbol{x}) = \sum_{i=1}^{E} p_i(\boldsymbol{x}) f_i(\boldsymbol{x}). \tag{2}$$

The routing probabilities $p_i(\boldsymbol{x})$ are usually computed by a routing layer, e.g., $p_i(\boldsymbol{x}) = \sigma(\boldsymbol{S}\boldsymbol{x})_i$, where σ is the softmax and $\boldsymbol{S} \in \mathbb{R}^{E \times D}$ is a trainable variable. Such linear routing layers are the common choice for MoE models in practice [10].

Though these models have recently been shown to achieve state of art performance for tasks in NLP [10] and vision [22], there have not been many works analyzing their robustness. Recently, Allingham et al. [2] empirically studied robustness of MoEs to natural perturbations in data (e.g., evaluation across different corrupted versions of CIFAR10 [17] and ImageNet [6]), and showed MoEs are more robust for these natural perturbations than the corresponding dense models. To the best of our knowledge ours is the first work theoretically analyzing robustness of MoEs, and empirically exploring it in the context of adversarial perturbations.

Load balancing loss To prevent MoEs from collapsing and always selecting the same expert for all inputs, it is customary during training to add a *load balancing loss*, that encourages equal fraction of inputs being routed to different experts. We define and present these losses in detail in appendix E.

2.2 Adversarial robustness

Adversarial robustness models the susceptibility of a function to adversarial perturbations to its input [25, 20]. More concretely, given a function f, loss function ℓ, and an input $\boldsymbol{x} \in \mathbb{R}^D$, we can write the *adversarial loss* incurred at $\boldsymbol{x}$ with label $\boldsymbol{y}$ as follows.

$$\max_{\boldsymbol{z}: \|\boldsymbol{z}\| \leq \epsilon} \ell(f(\boldsymbol{x} + \boldsymbol{z}), \boldsymbol{y}). \tag{3}$$

Here ϵ is the attack radius per input. For the norm constraint ($\|\boldsymbol{z}\|$), popular choices in practice, are ℓ_∞ and ℓ_2 norms [20, 4]. Adversarial accuracy is the accuracy of the model on data with perturbations that satisfy the above equation, i.e., when the loss function ℓ is the $0/1$ classification loss.

Since optimizing equation 3 for neural networks is computationally hard, popular methods to find adversarial examples use local search approaches such as gradient ascent. The Fast Gradient Signed Method (FGSM) is perhaps the simplest one, with only one gradient update step [11]. For ℓ_∞ bounded perturbations the FGSM update is the following:

$$\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{x} + \epsilon \text{ sgn } (\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), \boldsymbol{y})). \tag{4}$$

sgn() is the sign function. The Projected Gradient Descent (PGD) method is a stronger attack that does multiple (τ) gradient ascent steps to find the perturbation [20]. The update rule for ℓ_∞ bounded perturbations is as follows, starting with $\boldsymbol{x}^0 = \boldsymbol{x}$.

$$\boldsymbol{x}^{t+1} = \Pi_{\boldsymbol{x}+\mathcal{C}} \left[\boldsymbol{x}^t + \alpha \text{ sgn } (\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), \boldsymbol{y})) \right], \forall t \in [0, \tau - 1]. \tag{5}$$

α is chosen to be $\frac{\epsilon}{\tau}$. Here $\mathcal{C}$ is the constraint set $\{\boldsymbol{z} : \|\boldsymbol{z}\|_\infty \leq \epsilon\}$ and Π is the projection operator.

Existing works have established that neural networks are sensitive to adversarial attacks resulting in a significant drop in their accuracy [25]. Interestingly, increasing the size of neural networks, thereby increasing their capacity, leads to an improvement in adversarial accuracy, showing that larger neural networks are more robust [20, 26, 1, 12]. In a recent work, Bubeck and Sellke [3] studied this phenomenon theoretically. They proved a lower bound on the Lipschitz constant of any function that fits the training data that scales inversely $\frac{1}{\sqrt{P}}$ with the number of function parameters P. However it is not clear if functions that have more parameters, but not necessarily more computation cost (e.g. MoEs), can achieve better robustness. In this paper we analyze adversarial robustness of MoEs, both theoretically and experimentally.

Notation. We use small bold letters, e.g., $\boldsymbol{x}$, to denote vectors and capital bold letters to denote matrices, e.g., $\boldsymbol{W}$. Scalars are denoted with plain letters. $[N]$ denotes the integer set from 1 to N. $\|.\|$ denotes the ℓ_2 norm unless specified otherwise.

3 Robustness analysis

In this section, we present our main results regarding the robustness of the mixture of expert models. We first present a bound on the Lipschitz constant of MoEs for general expert functions f_i. While this is an upper bound for general functions it still highlights the two key components that affect the robustness of MoE models.

3.1 Router stability

In this section, we compute the Lipschitz constant of smooth MoEs with learnable routing. Let,

$$f(\boldsymbol{x}) = \sum_{i=1}^{E} p_i(\boldsymbol{x}) f_i(\boldsymbol{x}) \;\text{ where }\; p_i(\boldsymbol{x}) = \frac{\exp(\langle \boldsymbol{s}_i, \boldsymbol{x}\rangle)}{\sum_{j=1}^{E} \exp(\langle \boldsymbol{s}_j, \boldsymbol{x}\rangle)} \tag{6}$$

and $\{\boldsymbol{s}_i\}_{i\in[E]}$, with each $\boldsymbol{s}_i$ in $\mathbb{R}^D$, are learnable variables that decide the routing of an example to different experts. We then prove the following upper bound on the Lipschitz constant of the MoE models with learnable routing.

Lemma 1. *Let $\{f_i\}_{i\in[E]}$ be smooth functions with Lipschitz constants $\{L_{f_i}\}_{i\in[E]}$ and let L_f be the Lipschitz constant of f. Let $\bar{\boldsymbol{s}}(\boldsymbol{x}) = \sum_j p_j(\boldsymbol{x})\boldsymbol{s}_j$. Then $f(\boldsymbol{x})$ in equation 6 satisfies,*

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\| \leq \sum_{i=1}^{E} p_i(\boldsymbol{x}) L_{f_i} + \left\| \sum_{i=1}^{E} p_i(\boldsymbol{x}) f_i(\boldsymbol{x})(\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x})) \right\|,$$

and hence

$$L_f \leq \max_{i\in[E]} \{L_{f_i}\} + \sup_x \left\| \sum_{i=1}^{E} p_i(\boldsymbol{x}) f_i(\boldsymbol{x})(\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x})) \right\|. \tag{7}$$

The above Lemma bounds the Lipschitz constant of MoE models with two terms that depend on (1) data routing, and (2) router stability. The first term above is from the individual Lipschitz constant of the experts. This depends on how the data is partitioned or routed to different experts, which we will discuss in more detail in the next section. The second term arises from the router used to compute probabilities for different experts. One may wonder if we can make this term arbitrarily large by increasing the norm of $\boldsymbol{x}$. This is not the case as increasing the norm of $\boldsymbol{x}$ usually results in the collapse of the routing probabilities to a single expert. This makes $p_i(\boldsymbol{x})(\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x}))$ zero for all experts $i \in [E]$.

To study the above more concretely, we consider the case of $E = 2$ experts. In this setting the second term above reduces to $\|p_1(\boldsymbol{x})f_1(\boldsymbol{x})(\boldsymbol{s}_1 - \bar{\boldsymbol{s}}(\boldsymbol{x})) + (1 - p_1(\boldsymbol{x}))f_2(\boldsymbol{x})(\boldsymbol{s}_2 - \bar{\boldsymbol{s}}(\boldsymbol{x}))\|$. Now, in the extreme case, where $p_1(\boldsymbol{x})$ is either 0 or 1, the above term collapses to 0 as $p_i(\boldsymbol{x})f_i(\boldsymbol{x})(\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x}))$ is 0 in that case. This is also expected as for examples that are routed to an expert with high probability, the main dominating factor is the individual expert Lipschitz constant, as small perturbations do not cause much change to the probabilities.

A more interesting setting is when $p_1(\boldsymbol{x})$ is $\frac{1}{2}$. In this setting, small perturbations to the input can cause the model to drastically change the weight between the experts, and the above term reduces to $\|\frac{1}{4} \cdot (f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})) \cdot (\boldsymbol{s}_1 - \boldsymbol{s}_2)\|$. Hence MoE models can suffer from large Lipschitz constants if the two experts are very different for points on the boundary, i.e., if $|f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})| \gg 0$. Alternately if $f_1(\boldsymbol{x}) \approx f_2(\boldsymbol{x})$ for points on the boundary (i.e. $p_1(\boldsymbol{x}) \approx \frac{1}{2}$) then this term is small.

We will later see in the experiments that MoEs trained in practice often have good router stability and changes to routing do not result in much degradation of the final accuracy.

3.2 Data routing

In this section we will study the effect of data routing on the Lipschitz constant of the individual experts – the first term in RHS of equation 7 (lemma 1). In particular we will try to address how large can $\max_{i\in[E]}\{L_{f_i}\}$ be in comparison to the Lipschitz constant of a dense model trained on the same data. Towards this end, we will theoretically analyze the robustness of MoE models with a pre-determined fixed routing and linear experts, in the light of the structure of the data.

3.2.1 Setup

We consider the setting of fixed routing, where we assume the routing of individual examples to experts is pre-determined and fixed. Moreover, we consider *linear models* as experts.

More specifically, we consider N input points of dimension D stacked in the matrix $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, together with the corresponding vector of targets $\boldsymbol{y} \in \mathbb{R}^N$. In the dense case (i.e., no experts), we would like to find a linear model $\boldsymbol{w}^* \in \mathbb{R}^D$ that minimizes

$$\min_{\boldsymbol{w}\in\mathbb{R}^D} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2. \tag{8}$$

The least-squares solution [13] for this problem that minimizes the MSE loss is $\boldsymbol{w}^* = (\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top\boldsymbol{y}$. Here † denotes the pseudo inverse. On the other hand, for mixture of experts, let the dataset be split into E subsets $\mathcal{S}_1, \cdots, \mathcal{S}_E$, with $\boldsymbol{X}_i, \boldsymbol{y}_i$ denoting the data in set $\mathcal{S}_i$. Now let the data from the set $\mathcal{S}_i$ be routed to the expert i. Below we write the objective for each expert:

$$\min_{\boldsymbol{w}_i\in\mathbb{R}^D} \|\boldsymbol{X}_i\boldsymbol{w}_i - \boldsymbol{y}_i\|^2 \ \text{ for } i \in [E]. \tag{9}$$

Similarly, let $\boldsymbol{w}_i^* = (\boldsymbol{X}_i^\top\boldsymbol{X}_i)^\dagger\boldsymbol{X}_i^\top\boldsymbol{y}_i$ be the optimal solution for expert i that minimizes the MSE.

3.2.2 Analysis

Before introducing our analysis relating the Lipschitz constant of the dense and MoE models, we first present two examples to better understand how routing affects Lipschitz constants. In particular we consider a simple setting with $E = 2$ experts. Let $\boldsymbol{X}^\top = [\boldsymbol{X}_1^\top, \boldsymbol{X}_2^\top]$, be the data routed to the two experts. Recall that the Lipschitz constant of the function $f(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{w}$ is $\|\boldsymbol{w}\|$.

Case $\boldsymbol{X}_1 \perp \boldsymbol{X}_2$. In this setting we get

$$\begin{aligned}\|\boldsymbol{w}^*\| = \|(\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top\boldsymbol{y}\| = \|(\boldsymbol{X}_1^\top\boldsymbol{X}_1)^\dagger\boldsymbol{X}_1^\top\boldsymbol{y}_1 + (\boldsymbol{X}_2^\top\boldsymbol{X}_2)^\dagger\boldsymbol{X}_2^\top\boldsymbol{y}_2\| &= \|\boldsymbol{w}_1^* + \boldsymbol{w}_2^*\| \\ &\geq \max(\|\boldsymbol{w}_1^*\|, \|\boldsymbol{w}_2^*\|)\end{aligned}$$

The second equality follows from the folowing steps - 1) $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{X}_1^T\boldsymbol{X}_1 + \boldsymbol{X}_2^T\boldsymbol{X}_2$, 2) $(\boldsymbol{X}_1^T\boldsymbol{X}_1 + \boldsymbol{X}_2^T\boldsymbol{X}_2)^\dagger = (\boldsymbol{X}_1^T\boldsymbol{X}_1)^\dagger + (\boldsymbol{X}_2^T\boldsymbol{X}_2)^\dagger$, and 3) $(\boldsymbol{X}_1^T\boldsymbol{X}_1)^\dagger\boldsymbol{X}_2^T = 0$, where steps 2 and 3 follow from the orthogonality of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ (see [16]). Hence, experts have smaller Lipschitz constant when data routed to different experts lies in orthogonal subspaces.

Case $\boldsymbol{X}_1 = -\boldsymbol{X}_2$ and $\boldsymbol{y}_1 = \boldsymbol{y}_2$. In this case, we see that:

$$\begin{aligned}\|\boldsymbol{w}^*\| = \|(\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top\boldsymbol{y}\| &= \|(\boldsymbol{X}_1^\top\boldsymbol{X}_1 + \boldsymbol{X}_2^\top\boldsymbol{X}_2)^\dagger\boldsymbol{X}^\top\boldsymbol{y}\| \\ &= \|(\boldsymbol{X}_1^\top\boldsymbol{X}_1 + \boldsymbol{X}_2^\top\boldsymbol{X}_2)^\dagger(\boldsymbol{X}_1^\top\boldsymbol{y}_1 - \boldsymbol{X}_1^\top\boldsymbol{y}_1)\| = 0 \leq \min(\|\boldsymbol{w}_1^*\|, \|\boldsymbol{w}_2^*\|).\end{aligned}$$

Hence experts have worse Lipschitz constant when the data routed to different experts is aligned. These two simple examples illustrate under which conditions experts have an advantage over a single dense model, and when they do not.

We now present our main result. To capture this relation between the data geometry and the routing, we introduce the following quantities. Let $\{\boldsymbol{U}_i\}_{i\in[E]}$ be the projection matrices onto orthogonal subspaces in $\mathbb{R}^D$. One way to construct them is by first taking the singular vectors of $\boldsymbol{X}^\top\boldsymbol{X}$ and assigning them to the set i with the largest projection. This guarantees that $\boldsymbol{U}_i$ is orthogonal to $\boldsymbol{U}_j$ with $j \neq i$, and they have greatest alignment with subspace spanned by $\boldsymbol{X}_i$.

We first define a quantity to capture how well $\boldsymbol{U}_i$ captures the span of the data subset $\boldsymbol{X}_i$.

Definition 1 (In-subspace distance: ϵ_1)**.** $\exists\epsilon_1 \geq 0$, $\|\boldsymbol{U}_i\boldsymbol{U}_i^\top(\boldsymbol{X}^\top\boldsymbol{X})^\dagger - (\boldsymbol{X}_i^\top\boldsymbol{X}_i)^\dagger\|_2 \leq \epsilon_1, \forall i \in [E]$.

Here $\|.\|_2$ for a matrix denotes the spectral norm. ϵ_1 is small when, $\forall i, \boldsymbol{U}_i$ captures $\boldsymbol{X}_i$ perfectly.

Next we define the projection distance between data from two different subsets.

Definition 2 (Cross-subspace distance: ϵ_2)**.** $\exists\epsilon_2 \geq 0$ *such that for any* $\boldsymbol{z}$ *in the span of* $\boldsymbol{X}_j^\top$*, we have* $\|(\boldsymbol{X}_i^\top\boldsymbol{X}_i)^\dagger\boldsymbol{z}\| \leq \epsilon_2\|\boldsymbol{z}\|, \forall i \neq j$.

Here ϵ_2 is small if the data in different subsets $\boldsymbol{X}_i$ lies in orthogonal subspaces.

Theorem 1. *Let* $\boldsymbol{w}^*$ *be the minimizer of equation 8 and* $\{\boldsymbol{w}_i^*\}_{i\in[E]}$ *be the minimizers of equation 9. Then,*

$$\|\boldsymbol{w}^*\|^2 \geq \sum_{i=1}^{E}(\lfloor\|\boldsymbol{w}_i^*\| - \epsilon_1\|\boldsymbol{X}^\top\boldsymbol{y}\| - \epsilon_2\sum_{j\neq i}\|\boldsymbol{X}_j^\top\boldsymbol{y}_j\|\rfloor_+)^2,$$

where $\lfloor\cdot\rfloor_+$ *denotes the projection onto non-negative numbers.*

The above result lower bounds the Lipschitz constant of the dense model $\|\boldsymbol{w}^*\|$ in terms of Lipschitz constants of the experts $\|\boldsymbol{w}_i^*\|$ in the MoE model. We present the proof of this theorem in appendix A.

In the case where $\epsilon_1 = \epsilon_2 = 0$, we obtain the following bound $\|\boldsymbol{w}^*\| \geq \sqrt{\sum_{i=1}^{E}\|\boldsymbol{w}_i^*\|^2}$. Assuming a balanced setting where all experts have the same norm parameters, we get the scaling $\|\boldsymbol{w}^*\| \approx \mathcal{O}(\sqrt{E}) \cdot \max_{i\in[E]}\{\|\boldsymbol{w}_i^*\|\}$. Hence in this setting, experts have a Lipschitz constant that is smaller by a factor of $\frac{1}{\sqrt{E}}$ compared to dense models. This happens when the data lies in E orthogonal subspaces and the data from each subspace is routed to the same expert. This shows that MoE models can have significantly smaller Lipschitz constant than their dense counterparts, while having the same computation cost. For MoEs in practice, data routed to different experts does display some clustering of the features (see Figure 7 in Riquelme et al. [22]).

As data in different partitions $\boldsymbol{X}_i$ gets more aligned, ϵ_1 and ϵ_2 increase, and reduce the gap between the Lipschitz constant of the dense and the MoE models. This reduces the gap in the Lipschitz constant of the dense model and the experts. In the extreme case, if all the datapoints are the same, then ϵ_1 and ϵ_2 are large, eliminating this difference. In appendix D, we give a complementary result relating the Lipschitz constants of the experts to that of the dense model when both ϵ_1 and ϵ_2 can possibly be large.

Connections to Bubeck and Sellke [3]. Bubeck and Sellke [3] proved a universal lower bound on the Lipschitz constant of a function required to δ-memorize N training samples in D dimension.

$$L_f \geq \tilde{\Omega}\left(\delta\sqrt{\frac{ND}{P}}\right).$$

For the linear model considered above, this reduces to $\tilde{\Omega}\left(\delta\sqrt{N}\right)$ as the number of parameters is D. As MoEs have E times more parameters, they have a lower bound of $\tilde{\Omega}\left(\delta\sqrt{\frac{N}{E}}\right)$, i.e., the lower bound on the Lipschitz constant of MoEs is smaller by a factor of $\sqrt{\frac{1}{E}}$. Our result shows that this lower bound is in fact achievable, and hence tight, when the data lies in E orthogonal subspaces and data from each subspace is routed to the same expert.

4 Experiments

4.1 Setup

We compare the robustness of Vision Transformer (ViT) [7] and Vision MoE (V-MoE) [22] models against adversarial attacks. In particular, we use the ViT-B/32 and V-MoE-B/32 models through all the experiments. These models have the same backbone architecture but the latter replaces one in every two feedforward layers with a *sparse* Mixture of Experts, selecting $K = 2$ out of $E = 32$ feedforward experts that are applied on each token[1]. One could argue that the router in the V-MoE model introduces an overhead that should be accounted for. Thus, we have trained a bigger version of the dense ViT-B/32 (which we coin ViT-B++/32) that reaches roughly the same predictive performance as the V-MoE model, but has higher cost. The cost of evaluating an image on the dense ViT models is 8.9 and 17.9 GFLOPs and runtime cost, respectively; and 12.4 GFLOPs on the V-MoE model. Appendix B contains additional experimental details.

We pre-train our models on the private dataset JFT-300M [24] for 7 epochs (517 859 steps with a batch size of 4 096 images), using an image resolution of 224×224 pixels, and standard data augmentation (inception crop and horizontal flips). Since JFT-300M is a multi-label dataset, we minimize the sigmoid cross-entropy loss. V-MoE also adds auxiliary losses to encourage a balanced load for all experts; we used the same recipe as in [22]. In both cases we use Adam ($\beta_1 = 0.9, \beta_2 = 0.999$), with a peak learning rate of $8 \cdot 10^{-4}$, reached after a linear warm-up of 10^4 steps and then linearly decayed to a final value of 10^{-5}. Weight decay of 0.1 was used on all parameters. This is the same pre-training protocol as that used in [7, 22].

After pre-training, the models are fine-tuned on ImageNet [6], at a resolution of 384×384 pixels and the same data augmentations as before, for a total of 10^4 steps, using a batch size of 4 096 images. SGD with Momentum ($\mu = 0.9$) is used for fine-tuning, with a peak learning rate of 0.03, reached after a linear warm-up of 500 steps, and followed with cosine decay to a final value of 10^{-5}. The norm of the flattened vector of gradients is clipped to a maximum value of 10. Since ImageNet images have a single label, we minimize the softmax cross-entropy loss during fine-tuning. The same fine-tuning protocol was adopted in [7, 22].

We evaluate the adversarial robustness of both the pre-trained and fine-tuned models, by means of PGD adversarial attacks [20]. We maximize the corresponding loss (sigmoid or softmax cross-entropy), varying the ℓ_∞ norm constraint on the input image, for a total of $\tau = 40$ steps.

4.2 Adversarial robustness of V-MoEs

During pre-training the ViT-B/32 model achieves a precision at 1 of 39.3%, and the V-MoE-B/32 achieves a precision-at-1 of 43.5% (conversely, the false discovery rate at 1 is 60.7% and 56.5%, respectively). After fine-tuning, the classification error achieved by each model on ImageNet is 19.3% and 17.8%, respectively.

Figure 1 shows in solid lines the false discovery rate (left) and the classification error rate (right) as a function of the ℓ_∞ constraint. Despite the fact that the V-MoE model contains a router that makes discrete choices among the experts conditioned on the input, which could potentially lead to a severe weakness against the adversarial attacker, we can observe that it follows the same trend as the base dense ViT model. It is able to preserve a lower error over a wide range of ℓ_∞ values. A larger version of the dense model matching the quality of the V-MoE has a much higher cost.

In addition, we also fine-tuned the base ViT and V-MoE models on ImageNet using PGD adversarial training. We use the same recipe as above but we perform a PGD attack of 10 steps on the input images, with fixed $\ell_\infty = \frac{8}{255}$, before computing the gradients of the model parameters and updating them. The classification error on the original ImageNet dataset achieved by each model after adversarial fine-tuning is 51.7% and 49.8%, for both models respectively. Figure 1 (right) shows in dashed lines the classification error when these models are evaluated against an adversarial attacker using different ℓ_∞ constraints. As the ℓ_∞ increases, the benefit of adversarial fine-tuning to preserve accuracy is shown in both cases. Once again, both models report similar trends, and the V-MoE model shows better robustness for a wide range of ℓ_∞ values.

[1] Token refers to an element in the sequence input, which is obtained by projecting input image patches [7]

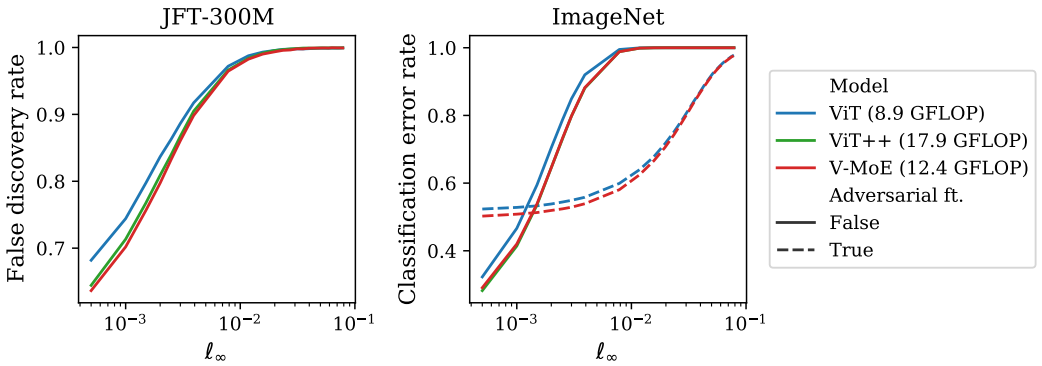


Figure 1: False discovery rate on JFT-300M (left) and classification error rate on ImageNet (right) as a function of the ℓ_∞ used in the adversarial attacks. Dashed lines depict models fine-tuned with PGD adversarial training. Although the V-MoE model contains several sparse MoE layers making discrete choices on their respective inputs, it shows lower error under adversarial attacks than the base ViT model. A much bigger and slower ViT model is needed to roughly match the V-MoE.

4.3 Effect of the adversarial attacks on the selected experts

As described in section 3.1, in the region close to the decision boundary of the router, if two experts have very different outputs, the Lipschitz constant of the MoE model could be much higher than that of a similar dense model. If the index of the selected experts changes significantly, but the model still shows a reasonably high accuracy under adversarial attacks (compared to a dense model), this would suggest that the outputs of the two selected sets of experts do not differ much.

Figure 2 shows the rate of changes in the router as a function of the ℓ_∞ used in the adversarial attack, on the different layers of the V-MoE-B/32 model that have a MoE layer. For each token processed by the model, we compute the intersection-over-union (IoU) of the selected set of experts before and after the adversarial attack. We average the IoU across all processed tokens and define the rate of routing changes as the complement of the average IoU.

In dashed lines we report the rate of changes of a V-MoE model fine-tuned using PGD adversarial training. Not only the model has a better accuracy against adversarial attacks when the ℓ_∞ increases, as reported in figure 1, but the rate of routing changes is also generally lower across all layers.

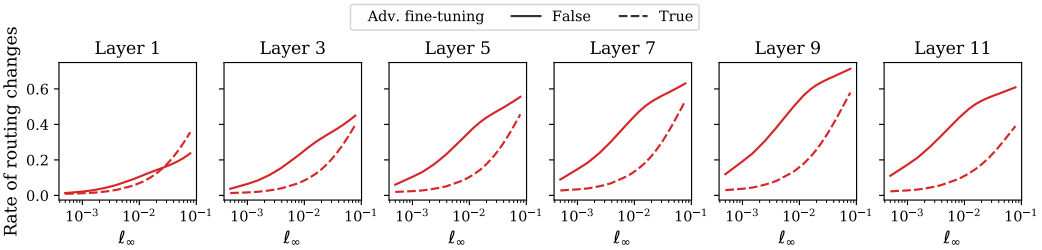


Figure 2: Rate of changes in the selection of experts as a function of the ℓ_∞ used in the adversarial attacks on ImageNet against a V-MoE model. The dashed line depicts a V-MoE model fine-tuned with PGD adversarial training. A significant fraction of experts change *in each MoE layer*. Given that figure 1 shows the V-MoE model keeping its advantage over the dense ViT, we hypothesize that the output of different experts for inputs close to the decision boundary of the router is similar.

Despite the fact that a significant fraction of choices change *on each layer* as the ℓ_∞ increases, figure 1 shows that the V-MoE model still keeps its advantage against adversarial attacks over the ViT model. This suggests that in the regions close to a decision boundary of the router, the two corresponding experts have a similar output, hence preventing the full V-MoE model from being less robust in practice. Conversely, the fact that the V-MoE model has a better base quality than the ViT counterpart, suggests that the experts are not equivalent for regions far away from the decision boundary, otherwise it would be reduce to a dense model.

4.4 Attacking the router's auxiliary losses

Sparse MoE models usually employ auxiliary losses to balance the load among all experts. In particular, in the implementation of V-MoEs, if the load of the experts is highly unbalanced, the experts receiving significantly more tokens than the average could ignore all tokens that exceed the expert's capacity, potentially leading to a significantly worse performance. The question is whether an adversarial attacker can exploit this property in practice. Figure 3 (left) shows the false discovery rate on JFT-300M, for a V-MoE model when the router's auxiliary losses are maximized in the adversarial attack, together with the corresponding cross-entropy loss. We use the same weight for the auxiliary losses as the one used to train the models. The figure shows that attacking the auxiliary loss does not offer any significant advantage for the attacker. The results are analogous in terms of the classification error on ImageNet (not shown here in interest of space).

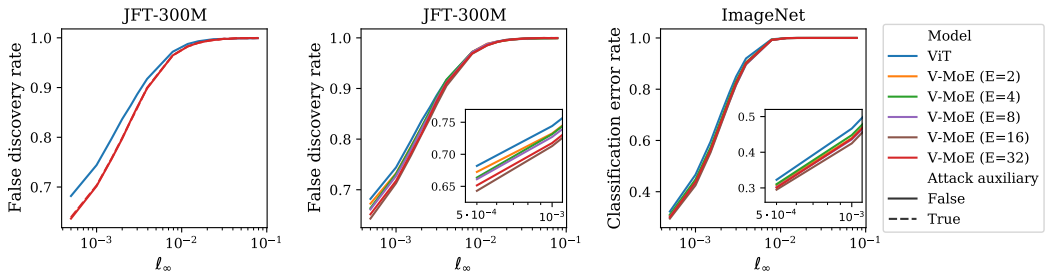


Figure 3: **(Left)** False discovery rate on JFT-300M for a ViT-B/32 and a V-MoE-B/32 models (with 32 experts). Dashed lines represent the results when the attacker targets both the cross-entropy and the auxiliary losses used by the V-MoE routers to balance the load among experts in each layer. The auxiliary losses of V-MoE do not present a disadvantage against an adversarial attacker. **(Center and Right)** False discovery rate on JFT-300M and classification error rate on ImageNet for a ViT-B/32 and several V-MoE-B/32 models (with increasing number of total experts). On JFT-300M the quality of the model against adversarial attacks improves as more experts are used. After fine-tuning on ImageNet, the number of experts becomes increasingly redundant.

4.5 Increasing the model size by increasing the total number of experts

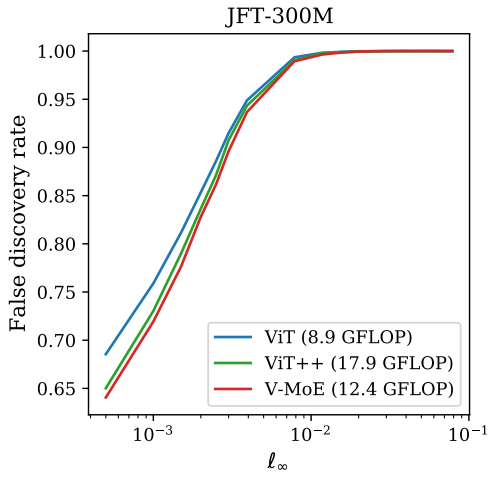
In section 1 we asked if, given that the lower bound on the Lipschitz constant given by Bubeck and Sellke [3] is agnostic to the computational cost of the function, could we make a model more robust by increasing its model size without increasing the total cost? Here we measure how much increasing the total number of experts in a V-MoE model helps against adversarial attacks. Notice that increasing the total number of experts E does not make the V-MoE model more expensive. Figure 3 (center and right) shows the false discovery rate on JFT-300M and the classification error on ImageNet for an increasing number of experts. All V-MoE models select $K = 2$ experts for each token.

On the one hand, increasing the total number of experts improves the robustness on JFT-300M up to $E = 16$ experts. The curves for $E = 16$ and $E = 32$ are highly overlapping, thus any difference is most likely due to noise in the training and fine-tuning process. On the other hand, when the V-MoEs are fine-tuned on ImageNet, all models with more than two experts achieve roughly the same accuracy under adversarial attacks.

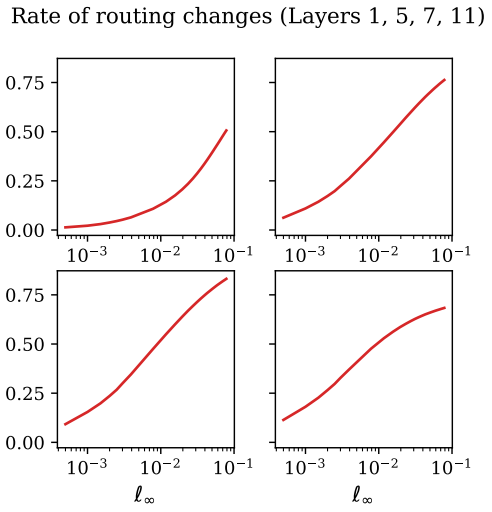
This shows that, although the results presented in section 3 showing better robustness of MoEs require some assumptions, the conclusions hold to some extent in real scenarios. Increasing the number of parameters by growing the number of experts is an effective way of improving the model's robustness.

4.6 Robustness against AutoPGD attacks

We conducted additional experiments using a more sophisticated adversarial attack, AutoPGD [5], which selects the step size to use in each update of the attack. We also increased the number of steps performed in the attack, using $\tau = 100$ steps. Figure 4a shows the results achieved by two dense (ViT-B/32 and ViT-B++/32) models and a sparse model (V-MoE-B/32). Although AutoPGD is slightly more effective as an adversarial attack against all methods (for example, with $\ell_\infty = 10^{-3}$ the false discovery rate on JFT-300M of ViT-B/32 is 0.744 using PGD and 0.757 using AutoPGD), the trend of all models is identical to that represented in figure 1.



(a) False discovery rate on JFT-300M (left) and classification error rate on ImageNet (right) as a function of the ℓ_∞, using AutoPGD and yielding the same conclusions as the simpler PGD method.



(b) Rate of changes in the selection of experts as a function of the ℓ_∞ used in the adversarial attacks on ImageNet against a V-MoE model.

Figure 4b shows the rate of routing changes in the different MoE layers in the V-MoE-B/32 model. This figure is analogous to figure 2 which shows the results for the standard PGD attack. Compared to it, AutoPGD is able to change a higher fraction of the selected experts. For example, the maximum fraction of changes in Layer 1 using PGD was near 0.2, while AutoPGD increases it up to around 0.5.

AutoPGD offers the same conclusion as the PGD attacks: models using sparse MoE layers offer better robustness against adversarial attacks (per GFLOP) than dense models, despite the fact that the router itself can be quite sensible to these attacks.

5 Conclusion

In this work we analyzed the adversarial robustness of MoEs showing their advantage over dense models, with more experts leading to better robustness, both theoretically and empirically. We showed how the properties of the data and its routing plays and important role in learning robust MoEs. While there is some evidence that routing learned by MoEs in practice display some clustering of the features in some layers of the model (see Figure 7 in Riquelme et al. [22]), it is currently not explicitly encouraged during training. Hence developing smarter routing strategies that take data geometry into account can be an interesting direction of future work. Currently our analysis is limited to linear models, extending this to general models and deriving the dependency of optimal routing on them is another promising research direction.

We have also shown that, for inputs that weigh two experts similarly, if the two expert values are very different, then the MoEs can suffer from higher Lipschitz constant. However, for models trained in practice, we saw their predictions to be relatively stable, despite significant changes in choice of experts, highlighting potential redundancy of learned experts. However too much redundancy, with all experts learning similar functions is a waste of capacity and can affect model performance. Hence it is an interesting research problem to balance robustness and accuracy of MoEs by controlling the redundancy of experts.

Acknowledgments and Disclosure of Funding

All authors would like to thank Sven Gowal for sharing their AutoPGD attack codebase. We also thank Neil Houlsby, Basil Mustafa, and André Susano Pinto for providing insightful feedback while working on this project.

References

[1] J.-B. Alayrac, J. Uesato, P.-S. Huang, A. Fawzi, R. Stanforth, and P. Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019.

[2] J. U. Allingham, F. Wenzel, Z. E. Mariet, B. Mustafa, J. Puigcerver, N. Houlsby, G. Jerfel, V. Fortuin, B. Lakshminarayanan, J. Snoek, D. Tran, C. R. Ruiz, and R. Jenatton. Sparse moes meet efficient ensembles. *CoRR*, abs/2110.03360, 2021. URL https://arxiv.org/abs/2110.03360.

[3] S. Bubeck and M. Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[5] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/croce20b.html.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[9] D. Eigen, M. Ranzato, and I. Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[10] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[12] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[13] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[14] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[15] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

[16] P. Kovanic. On the pseudoinverse of a sum of symmetric matrices with applications to estimation. *Kybernetika*, 15(5):341–348, 1979.

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[18] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[19] M. Lewis, S. Bhosale, T. Dettmers, N. Goyal, and L. Zettlemoyer. Base layers: Simplifying training of large, sparse models. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6265–6274. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lewis21a.html.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[21] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

[22] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby. Scaling Vision with Sparse Mixture of Experts. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/48237d9f2dea8c74c2a72126cf63d933-Paper.pdf.

[23] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[24] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[25] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[26] C. Xie and A. Yuille. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2020.

[27] F. Xue, Z. Shi, F. Wei, Y. Lou, Y. Liu, and Y. You. Go wider instead of deeper. *arXiv preprint arXiv:2107.11817*, 2021.

[28] S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012. doi: 10.1109/TNNLS.2012.2200299.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see sections 1 and 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work analyzes adversarial robustness of popular models.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please see sections sections 3.1 and 3.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] Please see appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The pre-training data is proprietary and cannot be publicly shared. Appendix B includes a description of the software stack used and a link to the software that we used to run all experiments.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We notice fairly stable adversarial accuracy metrics across different settings.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The hardware specifications and the total compute used in the experiments are detailed in appendix B.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please see section 4.
   (b) Did you mention the license of the assets? [No] Part of the data that we use is proprietary. The public data that we use (ImageNet) is publicly available after registration on `https://image-net.org`, and broadly used by the Machine Learning and Computer Vision communities.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Part of code we use is proprietary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs

Proof of lemma 1. We have by chain rule

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{i=1}^{k} p_i(\boldsymbol{x}) \nabla f_i(\boldsymbol{x}) + \sum_{i=1}^{k} \nabla p_i(\boldsymbol{x}) f_i(\boldsymbol{x}).$$

Hence,

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\| \leq \sum_{i=1}^{k} p_i(\boldsymbol{x}) L_{f_i} + \|\sum_{i=1}^{k} \nabla p_i(\boldsymbol{x}) f_i(\boldsymbol{x})\|,$$

where

$$\begin{aligned}
\nabla p_i(\boldsymbol{x}) &= \nabla \frac{\exp(\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle)}{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle)} \\
&= \frac{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle) \cdot \exp(\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle) \boldsymbol{s}_i - \exp(\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle) \cdot \sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle) \boldsymbol{s}_j}{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle) \cdot \sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle)} \\
&= \frac{\exp(\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle) \boldsymbol{s}_i}{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle)} - p_i \frac{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle) \boldsymbol{s}_j}{\sum_j \exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle)} \\
&= p_i \boldsymbol{s}_i - p_i \sum_j p_j \boldsymbol{s}_j.
\end{aligned}$$

Let $\bar{\boldsymbol{s}}(\boldsymbol{x}) = \sum_j p_j(\boldsymbol{x}) \boldsymbol{s}_j$. Substituting this in the above equation gives us

$$\begin{aligned}
\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\| &\leq \sum_{i=1}^{k} p_i(\boldsymbol{x}) L_{f_i} + \|\sum_{i=1}^{k} p_i(\boldsymbol{x}) f_i(\boldsymbol{x}) (\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x}))\| \\
&\leq \max_i \{L_{f_i}\} + \max_x \|\sum_{i=1}^{E} p_i(\boldsymbol{x}) f_i(\boldsymbol{x}) (\boldsymbol{s}_i - \bar{\boldsymbol{s}}(\boldsymbol{x}))\|.
\end{aligned}$$

The last inequality follows from Holder's inequality and $\sum_{i=1}^{k} p_i(\boldsymbol{x}) = 1$. □

Proof of theorem 1. Since $\boldsymbol{U}_i$ are orthogonal to each other, we can decompose $\|\boldsymbol{w}^*\|$ as follows.

$$\|\boldsymbol{w}^*\|^2 = \|(\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\|^2 = \|(\sum_i \boldsymbol{U}_i \boldsymbol{U}_i^\top)(\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\|^2 = \sum_i \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\|^2. \tag{10}$$

From definition 1 we have, for any unit norm vector $\boldsymbol{z}$,

$$\begin{aligned}
\epsilon_1 &\geq \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger - (\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger\|_2 \\
&\geq \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{z} - (\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{z}\| \\
&\geq \left| \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{z}\| - \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{z}\| \right|
\end{aligned}$$

The last step follows from triangle inequality. This implies

$$\|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{z}\| \leq \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{z}\| + \epsilon_1.$$

Hence, by setting $\boldsymbol{z} = \boldsymbol{X}^\top \boldsymbol{y} / \|\boldsymbol{X}^\top \boldsymbol{y}\|$, we see that

$$\|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| \geq \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| - \epsilon_1 \|\boldsymbol{X}^\top \boldsymbol{y}\|.$$

Recall that $\boldsymbol{X}^\top \boldsymbol{y} = \sum_{j=1}^{E} \boldsymbol{X}_j^\top \boldsymbol{y}_j$. Then using triangle inequality, and the above inequality, we get,

$$\begin{aligned}
\|\boldsymbol{w}_i^*\| &= \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{X}_i^\top \boldsymbol{y}_i\| \\
&\leq \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| + \sum_{j \neq i} \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{X}_j^\top \boldsymbol{y}_j\| \\
&\leq \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| + \epsilon_1 \|\boldsymbol{X}^\top \boldsymbol{y}\| + \sum_{j \neq i} \|(\boldsymbol{X}_i^\top \boldsymbol{X}_i)^\dagger \boldsymbol{X}_j^\top \boldsymbol{y}_j\| \\
&\leq \|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| + \epsilon_1 \|\boldsymbol{X}^\top \boldsymbol{y}\| + \epsilon_2 \sum_{j \neq i} \|\boldsymbol{X}_j^\top \boldsymbol{y}_j\|.
\end{aligned}$$

The last step follows from definition 2. Thus,

$$\|\boldsymbol{U}_i \boldsymbol{U}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top \boldsymbol{y}\| \geq \|\boldsymbol{w}_i^*\| - \epsilon_1 \|\boldsymbol{X}^\top \boldsymbol{y}\| - \epsilon_2 \sum_{j \neq i} \|\boldsymbol{X}_j^\top \boldsymbol{y}_j\|.$$

Substituting this in equation 10 gives us the result.

$$\|\boldsymbol{w}^*\|^2 \geq \sum_i (\lfloor \|\boldsymbol{w}_i^*\| - \epsilon_1 \|\boldsymbol{X}^\top \boldsymbol{y}\| - \epsilon_2 \sum_{j \neq i} \|\boldsymbol{X}_j^\top \boldsymbol{y}_j\| \rfloor_+)^2.$$

□

B Experiments details

B.1 Architectures

Table 1 contains the details of the architectures used in section 4. All V-MoE models have replaced one in every two feedforward layers in the original ViT-B/32 architecture, with a MoE of feedforward layers, and select only $K = 2$ experts out of the E available experts. All models were pre-trained on JFT-300M, at a resolution of 224×224 pixels, for a total number of 7 epochs, using the same batch size and optimizer settings as used by Dosovitskiy et al. [7] and Riquelme et al. [22] (see details section 4). We used the B/32 variants of ViT and V-MoE since these are the "base" configurations suggested in the respective papers.

In order to match the quality of a dense ViT and a sparse V-MoE model, we also trained a bigger version of ViT-B/32, which we refer to as ViT-B++/32. The values for the number of layers, number of attention heads, embedding dimension and the hidden dimension of the FFN, are simply an interpolation between the ViT-B/32 and the ViT-L/32 corresponding values.

We implemented all models, training and evaluation code using JAX[2] and FLAX[3]. Although we cannot release the models used in the experiments, since they are pre-trained on proprietary data, the code used in all of them is available at `http://github.com/google-research/vmoe`.

B.2 Compute resources

We use TPUv3 to train and evaluate our models. In particular, we used 32 TPUv3 cores for pre-training and fine-tuning the models. In most of the evaluation experiments against adversarial attacks we used 8 TPUv3 cores, since the compute needed to perform the evaluation against adversarial attack is much lower. The total training cost (in terms of total training runtime and FLOPs) of ViT-B/32 and V-MoE-B/32 can be found in [22].

ViT-B/32 is pre-trained for a total of 27.6 TPUv3-core-days, performing 56.1 ExaFLOPs. V-MoE-B/32 (with 32 experts) is trained for a total of 54.9 TPUv3-core-days and performs a total of 76.1 ExaFLOPs. Finally, the larger dense ViT-B++/32 is trained for a total of 78.3 TPUv3-core-days and performs 113.4 ExaFLOPs.

The GFLOPs are computed automatically by JAX. For instance, to compute the GFLOPs used for each individual image during the evaluation of the models, the code snippet in listing 1 is used.

[2] `https://github.com/google/jax`
[3] `https://github.com/google/flax`

Table 1: Architecture details of all the models used in the experiments. We specify here the number of layers in the Transformer encoder, the number of heads used in the multi-head attention, the embedding dimension (i.e. token size), the hidden size in the feedforward (FFN) layers and experts, the selected and total number of experts (when applicable), the active/total number of parameters, and the cost of evaluation per image (at a resolution of 224×224 pixels). Precision-at-1 on the pre-training dataset (JFT-300M) and classification error rate on ImageNet (ILSVRC2012) are also reported.

Name	ViT-B/32	ViT-B++/32	V-MoE-B/32				
			E=2	E=4	E=8	E=16	E=32
Layers	12	18			12		
Heads	12	14			12		
Embedding dim.	768	896			768		
FFN dim.	3072	3584			3072		
Selected/Total experts	—	—	2/2	2/4	2/8	2/16	2/32
Active parameters (M)	102.1	193.6	130.5	130.5	130.5	130.5	130.6
Total parameters (M)	102.1	193.6	130.5	187.1	300.5	527.2	980.6
Eval. GFLOPs/image	8.9	17.9	12.1	12.2	12.2	12.3	12.4
JFT-300M P@1 (%)	39.3	42.9	40.5	41.4	42.7	43.6	43.5
ILSVRC2012 error (%)	19.3	17.2	18.9	18.7	18.0	17.8	17.8

Listing 1: Code snippet used to compute the evaluation GFLOPs per image.

```
client = jax.lib.xla_bridge.get_backend()
# eval_step_fn is the function called to evaluate one batch.
m = jax.xla_computation(eval_step_fn)(...).as_hlo_module()
analysis = jax.lib.xla_client._xla.hlo_module_cost_analysis(client, m)
eval_gflops_per_image = analysis['flops'] / 10**9 / batch_size
```

C Adversarial examples

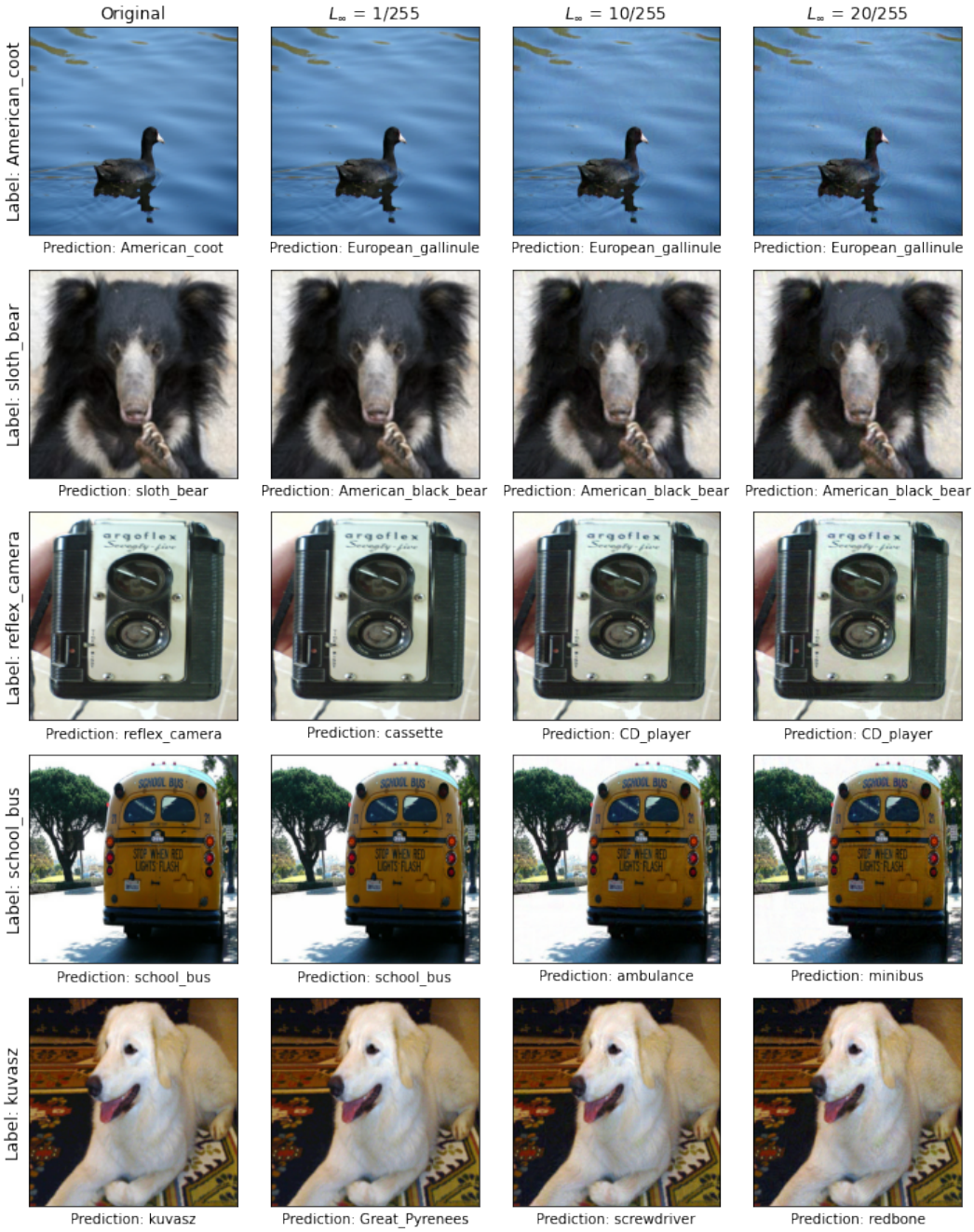


Figure 5: ImageNet adversarial examples generated when attacking a V-MoE-B/32, for different values of L_{inf}. Although the differences are hard to spot (better seen in a monitor, zooming in the images), they are sufficient to change the prediction of the model.

D A complementary result to theorem 1

We now present our complementary result. In essence, in a regression setting, our result tries to capture how different the sub-problems tackled by the experts should be in order to get improved robustness compared with a dense approach.

In what follows, we use the shortcuts $\boldsymbol{Z} = \boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{D\times D}$ and $\boldsymbol{Z}_i = \boldsymbol{X}_i^\top \boldsymbol{X}_i \in \mathbb{R}^{D\times D}$ for $i \in [E]$. Also, we recall that each $\boldsymbol{X}_i$ is in $\mathbb{R}^{N_i\times D}$ with $\sum_{i=1}^E N_i = N$.

Theorem 2. *Let $\boldsymbol{w}^*$ be the minimizer of equation 8 and $\{\boldsymbol{w}_i^*\}_{i\in[E]}$ be the minimizers of equation 9. For all $i \in [E]$, let us assume that there exist $\beta_i \in \mathbb{R}^D$ and $\boldsymbol{r}_i \in \mathbb{R}^{N_i}$ such that we can write*

$$\boldsymbol{y}_i = \boldsymbol{X}_i\beta_i + \boldsymbol{r}_i$$

where the vectors $\boldsymbol{r}_i$'s are residual terms that can for instance account for non-linear effects in $\boldsymbol{X}_i$ and/or some stochastic noise. Further assume the $\{\boldsymbol{Z}_i\}_{i\in[E]}$ to be invertible and define

$$\eta_i = \boldsymbol{Z}_i^{-1}\boldsymbol{X}_i^\top \boldsymbol{r}_i \quad \textit{and} \quad \eta = \boldsymbol{Z}^{-1}\boldsymbol{X}^\top \boldsymbol{r} \ \textit{ with } \ \boldsymbol{r} = [\boldsymbol{r}_1, \dots, \boldsymbol{r}_E] \in \mathbb{R}^N.$$

Let us denote the (normalized) difference between any two vectors (β_i, β_j) as $\Delta_{ij} \in \mathbb{R}^D$, that is

$$\textit{for any } \ i \neq j, \ \ \Delta_{ij} = \boldsymbol{Z}^{-1}\boldsymbol{Z}_j(\beta_i - \beta_j),$$

and consider the cumulative difference with respect to the vector β_i as

$$\textit{for any } \ i \in [E], \ \ \Delta_i = \sum_{j\neq i} \Delta_{ij} \in \mathbb{R}^D.$$

Consider $\gamma \in (0, 1]$. If the structures of the underlying sub-problems on $\{(\boldsymbol{X}_i, \boldsymbol{y}_i)\}_{i\in[E]}$ differ sufficiently from each other in the following sense

$$\textit{for all } i \in [E], \ \ \|\Delta_i\| \geq \|\beta_i + \eta\| + \frac{1}{\sqrt{\gamma}}\|\beta_i + \eta_i\|,$$

then it holds that

$$\max_{i\in[E]} \|\boldsymbol{w}_i^*\|^2 \leq \gamma\|\boldsymbol{w}^*\|^2.$$

In other words, the experts have smaller Lipschitz constants than the dense model by a factor γ. As a corollary, if we can control the residual terms $\boldsymbol{r}_i$'s such that

$$\|\eta\| \leq \min_{i\in[E]} \|\beta_i\| \ \textit{ and for all } i \in [E], \ \|\eta_i\| \leq \|\beta_i\|,$$

then the same conclusion is implied by the simpler sufficient condition

$$\min_{i\in[E]} \left\{\|\Delta_i\| - 4/\sqrt{\gamma} \cdot \|\beta_i\|\right\} \geq 0.$$

Proof of theorem 2. As a first comment, since we assume that all the $\boldsymbol{Z}_i$'s are invertible, then so is $\boldsymbol{Z} = \sum_{i=1}^E \boldsymbol{Z}_i$. We readily have for $i \in [E]$

$$\boldsymbol{w}_i^* = \boldsymbol{Z}_i^{-1}\boldsymbol{X}_i^\top \boldsymbol{y}_i = \beta_i + \boldsymbol{Z}_i^{-1}\boldsymbol{X}_i^\top \boldsymbol{r}_i = \beta_i + \eta_i.$$

Moreover, for any $i \in [E]$, we have

$$\begin{aligned}
\boldsymbol{w}^* &= \boldsymbol{Z}^{-1}\boldsymbol{X}^\top \boldsymbol{y} \\
&= \boldsymbol{Z}^{-1}\sum_{j=1}^E \boldsymbol{X}_j^\top \boldsymbol{y}_j \\
&= \boldsymbol{Z}^{-1}\sum_{j=1}^E (\boldsymbol{Z}_j\beta_j + \boldsymbol{X}_j^\top \boldsymbol{r}_j) \\
&= \left\{\boldsymbol{Z}^{-1}\sum_{j=1}^E \boldsymbol{Z}_j(\beta_i - \boldsymbol{Z}_j^{-1}\boldsymbol{Z}\Delta_{ij})\right\} + \eta \\
&= \boldsymbol{Z}^{-1}\left\{\sum_{j=1}^E \boldsymbol{Z}_j\right\}\beta_i - \sum_{j=1}^E \Delta_{ij} + \eta \\
&= \beta_i - \Delta_i + \eta \\
&= \boldsymbol{w}_i^* - \Delta_i + \eta - \eta_i.
\end{aligned}$$

To summarize, for any $i \in [E]$, it therefore holds

$$\|\boldsymbol{w}^* - \boldsymbol{w}_i^*\| = \|\eta - \eta_i - \Delta_i\|.$$

Let us consider $\gamma \in (0, 1]$. We can observe that

$$\gamma\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}_i^*\|^2 = \gamma\|\boldsymbol{w}^* - \boldsymbol{w}_i^*\|^2 - (1+\gamma)\|\boldsymbol{w}_i^*\|^2 + 2\gamma(\boldsymbol{w}^*)^\top \boldsymbol{w}_i^*.$$

We develop each term individually:

$$\gamma\|\boldsymbol{w}^* - \boldsymbol{w}_i^*\|^2 = \gamma\|\eta - \eta_i\|^2 + \gamma\|\Delta_i\|^2 - 2\gamma\Delta_i^\top(\eta - \eta_i) \tag{A}$$

with

$$-(1+\gamma)\|\boldsymbol{w}_i^*\|^2 = -(1+\gamma)\|\beta_i + \eta_i\|^2 \tag{B}$$

and using $\boldsymbol{w}^* = \beta_i - \Delta_i + \eta = \beta_i + \eta_i - \Delta_i + \eta - \eta_i$, we get

$$2\gamma(\boldsymbol{w}^*)^\top \boldsymbol{w}_i^* = 2\gamma\|\beta_i + \eta_i\|^2 - 2\gamma\Delta_i^\top(\beta_i + \eta_i) + 2\gamma(\beta_i + \eta_i)^\top(\eta - \eta_i). \tag{C}$$

Gathering **(A)+(B)+(C)**, we obtain the expression

$$\begin{aligned}\gamma\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}_i^*\|^2 &= \gamma\|\boldsymbol{w}^* - \boldsymbol{w}_i^*\|^2 - (1+\gamma)\|\boldsymbol{w}_i^*\|^2 + 2\gamma(\boldsymbol{w}^*)^\top \boldsymbol{w}_i^* \\ &= \gamma\|\Delta_i\|^2 - 2\gamma\Delta_i^\top(\beta_i + \eta) + \gamma\|\beta_i + \eta\|^2 - \|\beta_i + \eta_i\|^2 = \phi(\Delta_i).\end{aligned}$$

We want to understand the conditions on Δ_i to guarantee that

$$\gamma\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}_i^*\|^2 = \phi(\Delta_i) \geq 0.$$

To find a sufficient condition, we lower bound the quadratic form $\phi(\Delta_i)$ as follows

$$\begin{aligned}\text{For any } \Delta_i \in \mathbb{R}^D,\ \phi(\Delta_i) &\geq \gamma\|\Delta_i\|^2 - 2\gamma\|\Delta_i\|\|\beta_i + \eta\| + \gamma\|\beta_i + \eta\|^2 - \|\beta_i + \eta_i\|^2 \\ &= \gamma\left(\|\Delta_i\| - \text{root}_+\right) \cdot \left(\|\Delta_i\| - \text{root}_-\right)\end{aligned}$$

with

$$\text{root}_+ = \|\beta_i + \eta\| + \frac{1}{\sqrt{\gamma}}\|\beta_i + \eta_i\| \quad \text{and} \quad \text{root}_- = \|\beta_i + \eta\| - \frac{1}{\sqrt{\gamma}}\|\beta_i + \eta_i\|.$$

The lower bound is a standard polynomial of degree 2 and it is non-negative beyond its positive root

$$\|\Delta_i\| \geq \|\beta_i + \eta\| + \frac{1}{\sqrt{\gamma}}\|\beta_i + \eta_i\|.$$

We apply the exact same rationale for all $i \in [E]$ and take the minimum of the conditions to make them hold simultaneously.

Since $\gamma \in (0, 1]$ and $1/\sqrt{\gamma} \geq 1$, if we further assume that the residual terms are sufficiently small

- $\|\eta\| \leq \min_{i \in [E]} \|\beta_i\|$
- for all $i \in [E]$, $\|\eta_i\| \leq \|\beta_i\|$

then we get the simpler sufficient condition after using the triangle inequality

$$\|\Delta_i\| \geq \frac{4}{\sqrt{\gamma}}\|\beta_i\| \geq \|\beta_i + \eta\| + \frac{1}{\sqrt{\gamma}}\|\beta_i + \eta_i\|.$$

□

E Auxiliary losses for MoEs

In this section we present the Auxiliary losses used for training MoEs and attack objectives used for our experiments in section 4. Auxiliary losses are used to ensure the data is routed in a balanced way across different experts. In particular we use the losses from Riquelme et al. [22] which are defined below.

Recall that the router weights for a given token $\boldsymbol{x}$ and a given expert j, with routing parameters $\{\boldsymbol{s}_j\}_{j=1}^E$, were defined in equation 6 as:

$$p_j(\boldsymbol{x}) = \frac{\exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle)}{\sum_{j'=1}^E \exp(\langle \boldsymbol{s}'_j, \boldsymbol{x} \rangle)}$$

In practice, we use a noisy version of this router:

$$p_j(\boldsymbol{x}) = \frac{\exp(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle + \epsilon_j)}{\sum_{j'=1}^E \exp(\langle \boldsymbol{s}'_j, \boldsymbol{x} \rangle + \epsilon_{j'})}$$

with $\epsilon_j \sim \mathcal{N}(\mu = 0, \sigma = \frac{1}{E})$. The quantity $\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle$ is the routing *logit* for a given token and expert j.

Importance Loss. The importance of expert j for a batch of tokens $\boldsymbol{X}$ is simply defined as the sum of the router weights assigned to that expert over the tokens in the batch:

$$\text{Imp}_j(\boldsymbol{X}) = \sum_i p_j(\boldsymbol{x}_i) \tag{11}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top$ and $p_j(\boldsymbol{x}_i)$ is the routing weight assigned to the j-th expert for the i-th token.

We use the squared coefficient of variation of the importance distribution over experts, $\text{Imp}(\mathbf{X}) := \{\text{Imp}_j(\mathbf{X})\}_{j=1}^E$:

$$\mathcal{L}_{\text{Imp}}(\boldsymbol{X}) = \left(\frac{\text{std}(\text{Imp}(\boldsymbol{X}))}{\text{mean}(\text{Imp}(\boldsymbol{X}))} \right)^2 \tag{12}$$

Load Loss. For a given token, the load loss is based on the probability that the noisy logit of expert j is among the top-k logits when a new Gaussian sample is drawn. Let's define the k-th maximum noisy logit for a given token $\boldsymbol{x}$ as:

$$\tau_k(\boldsymbol{x}) = k\text{–max}_j \, \langle \boldsymbol{s}_j, \boldsymbol{x} \rangle + \epsilon_j \tag{13}$$

Then, the probability that the noisy logits of expert j are above the threshold if the noise is resampled, is given by the expression:

$$\pi_j(\boldsymbol{x}) = \mathbb{P}(\langle \boldsymbol{s}_j, \boldsymbol{x} \rangle + \epsilon \geq \tau_k(\boldsymbol{x})) \tag{14}$$

Analogous to the importance, the load of expert j is defined as the sum over the tokens in a given batch:

$$\text{Load}_j(\boldsymbol{X}) = \sum_i \pi_j(\boldsymbol{x}_i) \tag{15}$$

And the load loss corresponds to the squared coefficient of variation of the load distribution, with $\text{Load}(\boldsymbol{X}) := \{\text{Load}_j(\mathbf{X})\}_{j=1}^E$:

$$\mathcal{L}_{\text{Load}}(\boldsymbol{X}) = \left(\frac{\text{std}(\text{Load}(\boldsymbol{X}))}{\text{mean}(\text{Load}(\boldsymbol{X}))} \right)^2 \tag{16}$$

Final Loss. The final loss used for training the models includes the classification loss (i.e. cross-entropy) and both auxiliary losses with weights 0.005:

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + 0.005 \, \mathcal{L}_{\text{Imp}} + 0.005 \, \mathcal{L}_{\text{Load}} \tag{17}$$

F Experiment Metrics

Precision at 1 and False discovery rate In multi-label datasets (such as JFT), the traditional accuracy metric used for single-label classification is ill-defined. Usually, *Precision at k*, *Recall at k*, and other more general metrics are used in this setting. In particular, we use the *Precision at 1*.

Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]^\top, \boldsymbol{y}_i \in \mathbb{R}^C$ be the logits output by the model for a set of N examples, with true labels $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N]^\top, \hat{\boldsymbol{y}}_i \in \{0, 1\}^C$. *Precision at 1* is defined as:

$$\text{P@1} = \frac{1}{N} \sum_{i=1}^N \delta[\hat{\boldsymbol{y}}_{i, \arg\max_c \boldsymbol{y}_{i,c}} = 1] \tag{18}$$

where $\arg\max_c \boldsymbol{y}_{i,c}$ is essentially the index of the class with the highest logit, for a given example i, and δ is the Kronecker delta function. That is, the non-zero elements of the sum are those examples for which the highest logit corresponds to one of the true labels of such example. The false discovery rate is defined as $1 - P@1$.

Observe than when the examples have a single label, precision at 1 is equivalent to the accuracy, and the false discovery rate is equivalent to the error rate.

Rate of routing changes Given two sets A and B, the intersection over union (IoU) is defined as:

$$\text{IoU} = \frac{\text{card}(A \cap B)}{\text{card}(A \cup B)} \tag{19}$$

If the two sets are equal, IoU is 1. If the two sets do not share any element, IoU$is0$.

We use this metric in section 4.3 to compare how much the set of selected experts *change* when adversarial attacks are peformed on V-MoE models. Thus, the *rate of routing changes* is defined as $1 - \text{IoU}$.