A Provably Effective Method for Pruning Experts in Fine-tuned Sparse Mixture-of-Experts

Mohammed Nowaz Rabbani Chowdhury [1] **Meng Wang** [1] **Kaoutar El Maghraoui** [2] **Naigang Wang** [2]
Pin-Yu Chen [2] **Christopher Carothers** [3]

Abstract

The sparsely gated mixture of experts (MoE) architecture sends different inputs to different subnetworks (experts), through trainable routers. MoE reduces the training computation significantly for large models, but its deployment can be still memory/computation expensive for some downstream tasks. Model pruning is a popular approach to reduce inference computation, but its application in MoE architecture is largely unexplored. To the best of our knowledge, this paper provides the first provably efficient technique for pruning experts in fine-tuned MoE models. We theoretically prove that prioritizing the pruning of the experts with a smaller change of the router's l_2 norm from the pre-trained model guarantees the preservation of test accuracy, while significantly reducing the model size and the computational requirements. Although our theoretical analysis is centered on binary classification tasks on simplified MoE architecture, our expert pruning method is verified on large vision MoE models such as V-MoE and E^3-MoE fine-tuned on benchmark datasets such as CIFAR-10, CIFAR-100, and ImageNet.

1. Introduction

In deep learning, a typical approach is to adapt the same large pre-trained model (often based on the Transformer architecture (Vaswani et al., 2017)) to a plethora of downstream tasks. This method circumvents the substantial cost associated with training separate models for each task. This approach has consistently achieved state-of-the-art results in many application domains (Devlin et al., 2019; Dosovitskiy et al., 2020; Wortsman et al., 2022; Yu et al., 2022). Notably, a larger pre-trained model usually leads to better performance (Chen et al., 2022b; Chowdhery et al., 2023; Dehghani et al., 2023). However, an increase in model size typically leads to a proportional rise in the computational resources required for training, scaling linearly with the number of parameters.

The sparsely gated mixture of experts (MoE) has been introduced to reduce the training cost of large models (Shazeer et al., 2017; Lepikhin et al., 2020). The MoE layer in a Transformer encoder block replaces the single feed-forward network (FFN) module with multiple FFN modules, referred to as the *experts*. Each expert is associated with a trainable router that selectively activates the expert based on the input tokens. The sparse computation over the tokens allows for the expansion of the model size with only a sub-linear increase in the training compute (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022).

Due to their large model sizes, the deployment of MoE models still needs significant memory requirements, and inference compute (Riquelme et al., 2021; Zhou et al., 2022), which may restrict their applications in resource-constrained environments. Because different experts often learn diverse features (Lepikhin et al., 2020; Riquelme et al., 2021; Fedus et al., 2022), and not all experts are essential for a particular downstream task (Riquelme et al. (2021), Figure 29), pruning irrelevant and redundant experts may reduce the inference compute and memory requirement of a fine-tuned MoE model while maintaining the inference accuracy.

Despite the extensive research on neural network pruning (Han et al., 2015; 2016b;a; Frankle & Carbin, 2018; Li et al., 2020a; Frantar & Alistarh, 2023), pruning experts in the MoE architecture has not been much explored. To the best of our knowledge, only recent work by Chen et al. (2022a) and Koishekenov et al. (2023) prune non-essential experts, which are identified by the total number of tokens received by each expert throughout the fine-tuning stage or over the validation set after fine-tuning. These pruning methods can reduce memory requirements by reducing the number of

[1]Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, NY, USA [2]IBM Research, Yorktown Heights, NY, USA [3]Department of Computer Science, Rensselaer Polytechnic Institute, NY, USA. Correspondence to: Mohammed Nowaz Rabbani Chowdhury <chowdm2@rpi.edu>, Meng Wang <wangm7@rpi.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

experts and the communication among experts but cannot reduce the inference compute because tokens are routed to other experts after pruning. Moreover, no theoretical analysis has been provided, and it is not clear whether the total token count is the correct measure to identify essential experts. A fundamental question remains to be open:

For a given downstream task, what characteristic of an MoE layer provably separates the essential experts so that removing the rest does not hurt generalization?

This paper addresses this question empirically and theoretically on a binary supervised classification task[1]. We analyze the training dynamics of the routers and experts of a pre-trained MoE layer during the fine-tuning stage. Our analysis reveals that prioritizing the pruning of the experts with smaller l_2 *norm* change of the router's weights from the pre-trained model guarantees the preservation of test accuracy, while significantly reducing the model size and the computational requirements. The significance of our contributions are summarized as follows:

1. Theoretical generalization analysis: To the best of our knowledge, this paper provides **the first provably effective technique for pruning experts in MoE models**. We theoretically prove that experts who learned task-relevant features (that determine labels) have larger changes of their router's l_2 norm during the fine-tuning stage than those experts that learn task-irrelevant features. We then prove that pruning experts with smaller changes of their router's l_2 norm can guarantee the pruning of irrelevant and redundant experts, while maintaining the same test accuracy in a wide range of pruning ratios.

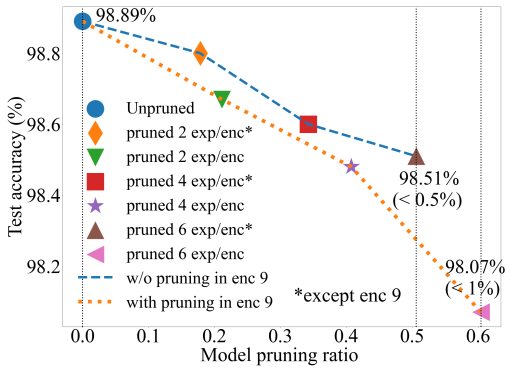


Figure 1: Generalization performance of the pruned VMoE on CIFAR-10 **with** post-pruning fine-tuning. 'pruned 2 exp/enc' implies pruning two experts from each MoE encoder.

2. Empirical validation: We provide **experimental demonstration of the proposed pruning technique's effectiveness on state-of-the-art vision MoE models**. We evaluate on several vision MoE (VMoE) (Riquelme et al., 2021) and ensembles of vision MoE (known as the efficient ensemble of experts, E^3) (Allingham et al., 2022) models with thousands of millions of parameters, fine-tuned on benchmark datasets such as CIFAR-10, CIFAR-100, and ImageNet. For example, as shown in Figure 1 our method can prune 75% of the experts of the fine-tuned V-MoE model on CIFAR-10 to reduce 60% of the memory requirements, while maintaining the model accuracy within 1% of the un-pruned model. Moreover, the method can reduce 40% of the inference FLOPs and 40% of inference time (see Section 5 for details). Furthermore, our pruning technique is designed for seamless integration with contemporary digital hardware accelerators, such as GPUs and TPUs, and can be implemented without the need for specialized software or hardware modifications.

[1]We consider binary tasks for the simplicity of the analysis. Our analysis can be extended to the multi-class case at the cost of higher complexity in the analysis.

2. Related Works

Mixture-of-Experts. The sparsely gated MoE sends different tokens of an input sequence to different experts in language models (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022; Du et al., 2022) and vision models (Riquelme et al., 2021; Puigcerver et al., 2022; Allingham et al., 2022). To address the challenge of load balancing among the experts (Lewis et al., 2021), *expert-choice routing* is introduced in Zhou et al. (2022) so that, instead of selecting experts for each token, the router selects tokens for each expert.

Despite the empirical success of MoE, its theoretical analysis is underexplored except for a few recent works. Specifically, Chen et al. (2022c) provides the theoretical generalization analysis of MoE with sample-level routing, and Chowdhury et al. (2023) proves the computational efficiency of modern MoE with patch-level routing.

Pruning deep neural networks. Network pruning has been widely explored recently to reduce the computation and memory cost of deep neural networks (DNN) (Han et al., 2016a; Luo et al., 2017; Lee et al., 2019b; Liu et al., 2021a; Jaiswal et al., 2023) Unstructured pruning methods prune individual weights (Han et al., 2015; 2016b; Frankle & Carbin, 2018; Wang et al., 2020; Liu et al., 2021b), while structured pruning methods remove neurons, channels, and layers (Li et al., 2016; Tung & Mori, 2018; Nonnenmacher et al., 2021). Although unstructured pruning can lead to a smaller model size because of the higher flexibility in pruning, the resulting irregular sparsity can lead to computational and memory overhead in practice. In contrast, structured pruning produces more regular and structured architecture, leading to possibly better hardware utilization and computational efficiency.

For large Transformer-based models, some focus on pruning pre-trained models (Chen et al., 2020; Zafrir et al., 2021; Li et al., 2024a) while others focus on task-specific pruning during or after the fine-tuning stage (Wang et al., 2020; Li et al., 2020b; Sanh et al., 2020). Note that all of these methods are compatible with our method and can be implemented together to further compress MoE models.

Convergence and Generalization Analyses of Neural Networks. The Neural Tangent Kernel (NTK) based approaches (Jacot et al., 2018; Lee et al., 2019a; Du et al., 2019; Allen-Zhu et al., 2019b; Li et al., 2022) assume the model weights stay close to the initialization during the training process, which might not reflect the practical learning dynamics. The model estimation approach (Zhong et al., 2017; Zhang et al., 2020b;a; Fu et al., 2020; Zhang et al., 2023; Li et al., 2024b) requires the input data to satisfy the Gaussian distribution. Recent works based on the feature learning framework (Daniely & Malach, 2020; Shalev-Shwartz et al., 2020; Shi et al., 2021; Allen-Zhu & Li, 2022; Zhang et al., 2022; Li et al., 2023; Allen-Zhu & Li, 2023; Chowdhury et al., 2023) can better characterize the practical learning dynamics in which the neural network gradually learns important features and discards unimportant features. Our theoretical analysis follows the feature learning framework.

3. Method

3.1. The Mixture-of-Experts Architecture

An MoE layer consists of multiple position-wise FFN modules referred to as experts, each associated with a router. The routers are collectively referred to as the gating network. Generally, MoE is implemented in the transformer encoder and decoder. As input samples in a transformer are tokenized, the MoE layer receives tokens as input.

Let us denote $x = \left[x^{(1)^T}, x^{(2)^T}, ..., x^{(n)^T}\right] \in \mathbb{R}^{dn}$ as the input sample of an MoE layer tokenized into n tokens of dimension d. Here, $x^{(j)} \in \mathbb{R}^d$ denotes the j-th token where $j \in [n]$. The MoE layer generates the n corresponding tokens of dimension d' of the output $x_{out} = \left[x_{out}^{(1)^T}, x_{out}^{(2)^T}, ..., x_{out}^{(n)^T}\right]$. The layer with k experts computes the output $x_{out}^{(j)} \in \mathbb{R}^{d'}$ for the token $x^{(j)}$ as,

$$x_{out}^{(j)} = \sum_{s\in[k]} f_s\left(x^{(j)}\right) \text{ where,} \tag{1}$$

$$f_s\left(x^{(j)}\right) = W_2^{(s)}\sigma\left(W_1^{(s)^T} x^{(j)}\right) G_j^{(s)} \tag{2}$$

$f_s\left(x^{(j)}\right)$ is the output for the expert $s \in [k]$ for input token $x^{(j)}$. Here, $W_1^{(s)} \in \mathbb{R}^{d\times m}$ represents the hidden layer weights of the expert $s \in [k]$ with hidden dimension m, i.e. with m hidden neurons. We denote the neuron $r \in [m]$ of expert $s \in [k]$ (the r-th column of $W_1^{(s)}$) by $w_r^{(s)}$. $\sigma(\cdot)$ is the element-wise activation function, and $W_2^{(s)} \in \mathbb{R}^{d'\times m}$ is the output layer weights converting hidden representations into output tokens.

The gating network. $G_j^{(s)} \in [0, 1]$ is the output of the gating network, referred to as the *gating value* associated with the expert s and the j-th token. To show how $G_j^{(s)}$ is calculated, for the MoE layer with k experts, the gating network contains k corresponding trainable routers denoted as $\{w_s\}_{s=1}^k \in \mathbb{R}^d$. We follow the definition in Riquelme et al. (2021) that the *routing value* associated with the router w_s and token $x^{(j)}$ as $g_j^{(s)} := \langle w_s, x^{(j)}\rangle$.

The routing function is characterized as either *token-choice routing* (Fedus et al., 2022) or the *expert-choice routing* (Zhou et al., 2022). Token-choice routing selects the top l experts for each token, according to the routing values of that token over the k experts. Let $J_j \subset [k]$ with $|J_j| = l$ denote the index set of the top l experts for the j-th token. Then gating values of the top l experts are non-zero and calculated as the softmax function over their routing values, while gating values for other experts are zero, i.e.,

$$G_j^{(s)} := \begin{cases} e^{g_j^{(s)}} / \sum_{i\in J_j} e^{g_j^{(i)}} & \text{if } s \in J_j \\ 0 & \text{else} \end{cases}. \tag{3}$$

Similarly, expert-choice routing selects top l tokens for each expert, according to the routing values of that expert over the n tokens of the input. Let $J_s \subset [n]$ with $|J_s| = l$ denote the index set of the top l tokens for the s-th expert. The gating values are computed by

$$G_j^{(s)} := \begin{cases} e^{g_j^{(s)}} / \sum_{i\in J_s} e^{g_i^{(s)}} & \text{if } j \in J_s \\ 0 & \text{else} \end{cases}. \tag{4}$$

See Figure 2 for a visual description of the two routing methods.

Note that zero gating values in (3)-(4) are introduced for mathematical completeness. In the implementation of sparse computation, $f_s\left(x^{(j)}\right)$ is directly set as zero when s is not in J_s for token-choice routing or when j is not in J_s for expert-choice routing, without actually computing (2). That means tokens are only ***routed*** to the experts with non-zero gating values.

3.2. Expert Pruning Method in MoE

Let $\{w_s^{(0)}, W_1^{(s,0)}, W_2^{(s,0)}\}_{s=1}^k$ denote the pre-trained weights of an MoE layer. The model is fine-tuned using the stochastic gradient descent (SGD) with batch size B, the expert learning rate η_e for parameters $W_1^{(s)}$ and $W_2^{(s)}$, and the router learning rate η_r for w_s. Let

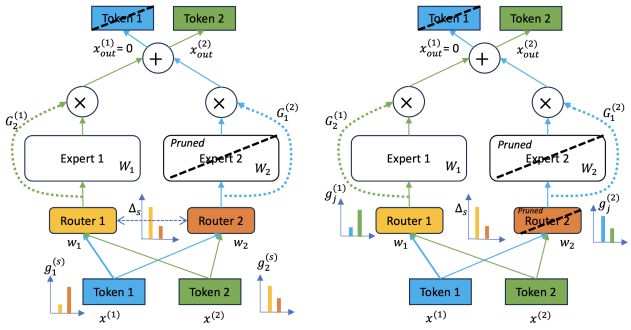


Figure 2: **Left:** Token-choice routing: each token selects experts based on the routing values over the experts. **Right:** Expert-choice routing: Each expert selects tokens based on the routing value over the tokens. In both cases, the experts with a smaller norm change of router's weights are pruned (Expert 2). The output tokens for the pruned experts are set to zero (Token 1 in the figure). In the left, the routers of the pruned experts are retained to calculate the gating value. In the right, the routers of the pruned experts are also pruned.

$\{w_s^{(T)}, W_1^{(s,T)}, W_2^{(s,T)}\}_{s=1}^k$ denote the corresponding fine-tuned weights after T iterations. We define the *change of router's l_2 norm* from the pre-trained model for any expert $s \in [k]$ after T training steps as,

$$\Delta_s^{(T)} := \|w_s^{(T)}\| - \|w_s^{(0)}\|$$

Let $S_{k'}$ denote the index set of remaining experts after pruning, where k' is the number of unpruned experts. $S_{k'}$ is selected based on $\Delta_s^{(T)}$. For example, $S_{k'}$ can contain experts with **top** k' values of $\{\Delta_s^{(T)}\}_{s=1}^k$. We define the *expert-pruning-ratio* of the pruned model as $\rho := 1 - (k'/k)$. Note that the computation of nonzero gating values $G_j^{(s)}$ is independent of the expert pruning. In token-choice routing, each token is only routed to those experts in $S_{k'}$ that are among the selected l experts for that token. The resulting gating values of token-choice routing with expert pruning is

$$G_j^{(s)} := \begin{cases} e^{g_j^{(s)}} / \sum_{i \in J_j} e^{g_j^{(i)}} & \text{if } s \in J_j \cap S_{k'} \\ 0 & \text{else} \end{cases}. \quad (5)$$

In expert-choice routing, because the routers select l tokens for each expert, if an expert is pruned, its corresponding router is also pruned. The resulting gating values of expert-choice routing with expert pruning is

$$G_j^{(s)} := \begin{cases} e^{g_j^{(s)}} / \sum_{i \in J_s} e^{g_i^{(s)}} & \text{if } j \in J_s \text{ and } s \in S_{k'} \\ 0 & \text{else} \end{cases}. \quad (6)$$

See Figure 2 for a visual description of the pruning method for the two routing techniques.

The model after pruning can be directly employed in a downstream task. It can also be fine-tuned again post-pruning using SGD with batch size B and learning rates η_e and η_r for T' iterations before deployment.

4. Theoretical Guarantees of the Expert Pruning Method

4.1. Key Theoretical Findings

We consider the setup that some tokens represent task-specific features that determine the data label in the downstream tasks, while some other tokens represent task-irrelevant features that do not affect the label. Before presenting our analysis setup and the formal theoretical results, we first present the key insights.

(I) Experts learning task-specific features have a large change in router's norm, while experts not learning task-specific features have a small change in router's norm. We theoretically show that the experts that learned task-specific features for the downstream task to a sufficient extent after pre-training continue to learn the task-specific features during fine-tuning, which leads to a large *change of the router's l_2 norm* in the fine-tuned model (Lemma 4.1). In contrast, the experts that do not learn task-specific features will still only learn irrelevant features during fine-tuning and have a small *change of the router's l_2 norm* in the fine-tuned model.

(II) Post-pruning fine-tuning promotes unpruned experts to learn task-specific features. We show that the routers of the unpruned experts always selects task-specific patterns. As a result, post-pruning fine-tuning can promote the neurons of these experts to learn task-specific features.

(III) The pruned model provides guaranteed generalization for a wide range of pruning ratios. Using the above two findings, we formally show all the experts not learning task-specific features can be pruned without hurting generalization accuracy. Moreover, if the pruned model can be further fine-tuned, one can prune up to $1 - O(1/k)$ fraction of experts with the least changes of router's l_2 norm while maintaining the generalization accuracy.

4.2. The Analysis Setup

Network architecture. We consider fine-tuning a pre-trained MoE layer with k experts on a binary classification task. Any input samples (x, y) is drawn from an unknown distribution $\mathcal{D}$ where $y \in \{+1, -1\}$. The output of the

fine-tuned model is defined as,

$$f(x) := \sum_{j=1}^{n} x_{out}^{(j)} = \sum_{j=1}^{n} \sum_{s\in[k]} f_s\left(x^{(j)}\right) \tag{7}$$

Here, the output token dimension, $d' = 1$. We replace the output layer weights $W_2^{(s)}$ of the pre-trained model by a fixed classification head $a^{(s)}\vec{1}$ where $\vec{1}$ is a vector of 1s with dimension m and where $a^{(s)}$ is generated from $\text{Unif}(\{+1, -1\})$. In other words, each expert is positively or negatively connected to the model's output. Following the typical setup of theoretical generalization analysis of neural networks (Li & Liang, 2018; Brutzkus et al., 2018; Allen-Zhu et al., 2019a; Arora et al., 2019; Zhang et al., 2022), $a^{(s)}$ is not updated during training. The activation function $\sigma(\cdot)$ is the rectified linear unit (ReLU) i.e. $\sigma(z) = \text{ReLU}(z) := \max(z, 0)$ for any $z \in \mathbb{R}$. We analyze the expert-choice routing MoE with $l = O(1)$ and $k = O(\sqrt{d})$, but our theoretical results are also validated in experiments on token-choice routing MoEs.

Training method. The learning method minimizes the hinge loss, $\hat{l}(f(x), y) = \max(1 - yf(x), 0)$ while the gradient is evaluated on $l(f(x), y) = 1 - yf(x)$, same as the setting in Zhang et al. (2022).We employ vanilla SGD using a batch size of B for T training steps with learning rate η_e and η_r in the experts and the routers, respectively.

Pruning model. Let us denote S_1 and S_2 as the two sets of experts positively and negatively connected to the output, respectively, i.e., $S_1 := \{s \in [k] : a^{(s)} = +1\}$ and $S_2 := \{s \in [k] : a^{(s)} = -1\}$. We prune the experts based on the *change of router's l_2 norm* separately over the set S_1 and S_2. More specifically, given an expert pruning ratio ρ, we retain the experts with top $|S_1|(1-\rho)$ values of $\{\Delta_s^{(T)}\}_{s\in S_1}$ and top $|S_2|(1-\rho)$ values of $\{\Delta_s^{(T)}\}_{s\in S_2}$ to construct the set $S_{k'}$ while pruning rest of the experts.

The data model[2]. The tokens are drawn from a fixed orthonormal pattern set, denoted by $\mathcal{P}$, which includes d patterns in $\mathbb{R}^d$, where $d = \Omega(n)$. $\mathcal{P}$ includes two *task-specific* patterns, denoted by o_1 and o_2, which determine the labels for class-1 ($y = +1$) and class-2 ($y = -1$), respectively. Each input sample x contains exactly one task-specific pattern, which determines the label. $\mathcal{P}$ also contains $d - 2$ *task-irrelevant* patterns, denoted by $\{q_i\}_{i=1}^{d-2}$, which do not affect labels. Each q_i appears in both classes with the same probability. The probability of q_i appearing in x can vary for different i, but is $O(1/d)$ for all i. We denote the data generating distribution by $\mathcal{D}$.

[2]Assumptions on the structure of the input data are required to analyze the feature-learning dynamics of neural networks. Similar assumptions have been made in many recent works (Brutzkus & Globerson, 2021; Shi et al., 2021; Karp et al., 2021; Allen-Zhu & Li, 2022; Chen et al., 2022c; Zhang et al., 2022; Li et al., 2022; Zhang et al., 2022; Allen-Zhu & Li, 2023; Chowdhury et al., 2023).

Experts' proficiency measure. We introduce a probability measure to quantify the quality of a router's capability of selecting task-specific features. Let $p_1^{(s,t)}$ and $p_2^{(s,t)}$ denote the *proficiency measure* of router s at iteration t in selecting the task-specific feature o_1 and o_2 with a gating value of at least $1/l$, respectively. Specifically,

$$p_1^{(s,t)} := \mathbb{P}[(x, +1) \sim \mathcal{D} : \exists j \in J_s^{(t)} \text{ s.t. } x^{(j)} = o_1 \text{ and, } G_j^{(s,t)} \geq 1/l]$$

Likewise for $p_2^{(s,t)}$. The larger values indicate the higher chances of expert s in selecting task-specific features.

4.3. Main Generalizalization Results of Expert Pruning

The generalization results of the pruned model without and with post-pruning fine-tuning are summarized in Theorems 4.3 and 4.5. We also present three important lemmas that lead to the theorems. Lemmas 4.1 and 4.2 jointly show that the *change of router's l_2 norm* in an important expert that learns task-specific features is significantly larger than the l_2 norm change of the router of unimportant expert that learns task-irrelevant features. Lemma 4.4 shows that post-pruning fine-turning can promote the unpruned experts to be specialized in learning task-specific features.

Lemma 4.1 (Important experts become more specialized)**.** *Suppose the expert learning rate η_e, the router learning rate η_r, the batch-size B, and the number of iterations T satisfy*

$$\eta_r = O(\eta_e/mdl^2), B = \Omega(l^2 d^2) \tag{8}$$

$$T = \Omega(l^2\sqrt{d\log l}/\eta_e). \tag{9}$$

For any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, we have

(i) $p_1^{(s,T)} = 1$,
(ii) for every $(x, +1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_1$,
(iii) $\langle w_r^{(s,T)}, o_1\rangle = \Omega(l\sqrt{d\log l})$, for a constant fraction $r \in [m]$,
(iv) $\Delta_s^{(T)} > \frac{3}{2}\log l$. The counterpart results also hold for experts in S_2.

Lemma 4.1 shows that expert s in the pre-trained model that learns the *task-specific* feature (say, o_1) to some extent (important experts, i.e., $p_1^{(s,0)} = \Omega(1)$) will become more specialized in learning the task-specific features after fine-tuning T iterations, and the norm of the corresponding router has a significant increase. Specifically, (i) indicates that the router s will always select o_1 as one of the l tokens in all class-1 samples. (ii) shows that for a class-1 sample x, the gating value that corresponds to the token o_1 is large, at least $1/2$. (iii) shows that a constant fraction of neurons in expert s has a large component along the direction of the

task-specific feature. (iv) shows that the router norm has a significant increase after fine-tuning.

In contrast, Lemma 4.2 shows that experts that do not learn the downstream task-specific features (unimportant experts) will still only learn task-irrelevant features after fine-tuning, and the norm changes of the corresponding routers are relatively small.

Lemma 4.2 (Unimportant experts stay unimportant)**.** *Suppose (8) and (9) hold. For any expert $s \in S_1$ such that $p_1^{(s,0)} = O(1/d)$, we have (i) $p_1^{(s,T)} = O(1/d)$, (ii) $\Delta_s^{(T)} = O(\log^2 l/\sqrt{d}) + O(l^4 \log^2 l/d^2)$. The counterpart results also hold for experts in S_2.*

Lemma 4.2 (i) indicates that the router of the fine-tuned expert still can only select task-specific features for a very small fraction of data, less than $O(1/d)$, while the router fails to select task-specific features for the dominating fraction of the samples. Lemma 4.2 (ii) shows that the router norm stays small. That is because l is $O(1)$, much smaller than d.

Next we will present the generalization result of expert pruning without post-pruning fine-tuning. Let $f^{(T)}(x)$ denote the output of the unpruned model in equation (7) after T SGD iterations. Let $\hat{f}^{(T;\rho)}(x)$ denote the resulting pruned model where ρ (in $[0, 1)$) fraction of experts are pruned.

Theorem 4.3 (Generalization of pruned model with no post-pruning fine-tuning)**.** *Suppose (8) and (9) hold, the number of experts $k = O(\sqrt{d})$, and at least γ fraction of $s \in S_1$ with $p_1^{(s,0)} = O(1/d)$ and $s' \in S_2$ with $p_2^{(s',0)} = O(1/d)$. Then, we have for any $0 \le \rho \le \gamma$,*

$$\mathbb{P}\left[\forall (x, y) \sim \mathcal{D} : y\hat{f}^{(T;\rho)}(x) > 0\right] = 1. \tag{10}$$

Theorem 4.3 shows that the pruned model with no post-pruning fine-tuning can still achieve zero generalization if up to γ fraction of experts with the smallest router l_2-norm change are removed. The intuition is that those γ fraction of experts do not learn task-specific features, and, thus, removing these experts does not affect generalization.

We next analyze the impact of post-pruning fine-tuning. If the pruned model is fine-tuned again for T' iterations, let $\hat{f}^{(T;\rho,T')}(x)$ denote resulting model after post-pruning fine-tuning. Lemma 4.4 shows that the routers with large norm change are specialized in selecting task-specific patterns. If these experts remain in the model after pruning, post-pruning fine-tuning can ensure that the hidden neurons of the experts are trained to learn task-specific features.

Lemma 4.4 (Post-pruning fine-tuning promotes experts to learn task-specific features)**.** *Suppose (8) and (9) hold. For any expert $s \in S_1$ such that $\Delta_s^{(T)} > \frac{3}{2}\log l$, we have*
(i) $p_1^{(s,T)} = 1$,
(ii) for every $(x, +1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_1$,
Moreover, after pruning experts with the ratio $\rho \ge \gamma$, if $s \in S_{k'}$, and the number of post-pruning fine-tuning steps satisfies

$$T' = \Omega(kl^2\sqrt{\log l}/\eta_e) \tag{11}$$

then we have

$$\langle w_r^{(s,T,T')}, o_1\rangle = \Omega(kl^2\sqrt{\log l}) \tag{12}$$

for a constant fraction $r \in [m]$, where $w_r^{(s,T,T')}$ is the resulting weights of neuron r in expert s. The counterpart results hold for experts in S_2.

The first half of Lemma 4.4 shows that if expert s has large router norm change after pre-pruning fine-tuning, then router s is specialized in selecting task-specific patterns and the corresponding gating values for task-specific patterns are large. The second of Lemma 4.4 shows that if we prune all unimportant experts and then continue to further fine-tuning expert s, then a constant fraction of neurons in expert s will be specialized in learning task-specific features. Therefore, if we allow post-pruning fine-tuning, the tolerable pruning rate can be large, as described in Theorem 4.5.

Theorem 4.5 (Generalization of pruned model with post-pruning fine-tuning)**.** *Suppose (8), (9) and (11) hold, and the number of experts $k = O(\sqrt{d})$. Then for any $\rho \le 1 - O(1/k)$,*

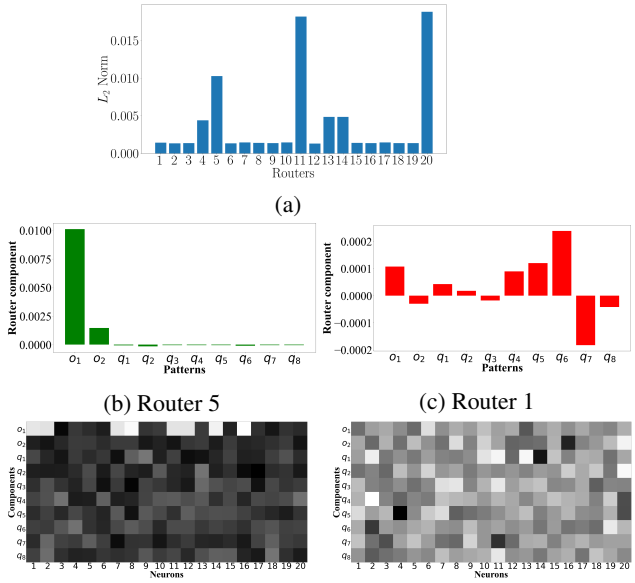
$$\mathbb{P}\left[\forall (x, y) \sim \mathcal{D} : y\hat{f}^{(T;\rho,T')}(x) > 0\right] = 1. \tag{13}$$

Theorem 4.5 guarantees that a high expert-pruning-ratio ($1 - O(1/k)$) is achievable without hurting the generalization performance using post-pruning fine-tuning. The intuition is that post-pruning fine-tuning allows some experts to become more specialized in learning task-specific patterns, and therefore, pruning more experts does not hurt generalization overall. The required number of post-pruning fine-tuning iterations T' is less than pre-pruning fine-tuning iterations T, because $k = O(\sqrt{d})$.

5. Experimental Results

5.1. Experiments on Synthetic Data

We verify our theoretical findings on synthetic data for the analyzed model. The data are generated by following the description given in section 4.2. We selected $d = 200$ and $n = 100$ to generate the data. We select 20 experts in the MoE layer (i.e., $k = 20$), each selecting five tokens (i.e., $l = 5$). The first ten experts are positively connected to the output, while the last ten are negatively connected. We initialize the routers and the neurons of the experts randomly following zero-mean Gaussian distribution with very small



(a)

(b) Router 5

(c) Router 1

(d) Expert 5

(e) Expert 1

Figure 3: (a) The norm of the post-training router weights, (b)(c) Projections of router weights to different directions, (d)(e) Projections of neuron weights to different directions (larger pixel intensity represents larger component of the router weights).

variance (1×10^{-8} and 1×10^{-4}, respectively) and train using SGD up to zero training error.

We present the norms of the post-training routerweights for the 20 routers in Figure 3a. One can see that in both groups, a few routers have significantly larger weight norms than others, e.g., experts 4 and 5 in the positive group, experts 11 and 20 in negative group. Figure 3b and 3c visualize the projection of router weights in different directions. The weight of Router 5 has a significant norm, and has a large component along the o_1 direction. In contrast, the weight of Router 1 has a small norm, and it does not have a significant component along o_1. Figure 3d and 3e show the projection of the neuron weights of experts 5 and 1 in these directions. One can see that a constant fraction of neurons in expert 5 have a large component along o_1, while neurons in expert 1 do not. We have similar results for the negative group, see section B.1 in the appendix. These results are consistent with Lemmas 4.1 and 4.2.

5.2. Experiments on State-of-the-art Vision MoE Models

We implement the proposed *change in router's l_2 norm* based expert pruning method in state-of-the-art vision MoE models fine-tuned on several benchmark datasets such as CIFAR-10, CIFAR-100 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015). More specifically, we implement the method to prune the released fine-tuned sparse vision MoE (V-MoE) by Riquelme et al. (2021). Moreover, we present results on the released fine-tuned efficient ensembles of experts in MoE (E^3-MoE) by Allingham et al. (2022).

The V-MoE model. The model contain 12 transformer encoders (encoders 0, 1, ..., and 11). Every odd encoder (i.e., encoders 1, 3, 5, 7, 9, and 11) is the MoE encoder. Each MoE encoder contains 8 experts (i.e., $k = 8$).

The E^3-MoE model. The model contain 8 transformer encoders (i.e., encoders 0, 1, ..., and 7), where the last two encoders contain the ensemble of MoEs (i.e., encoders 5 and 7). There are 2 MoE ensembles in the model and each MoE ensembles contains 4 experts per MoE encoder (i.e., $k = 4$).

Both of these models implement the *token-choice* routing with $l = 2$ and $l = 1$, respectively. More details can be found in section A of appendix.

Pruning and post-pruning fine-tuning details. We use both the pre-trained and the fine-tuned versions of the released model to calculate the change in the router's norm for each expert. We apply our pruning method over each MoE layer separately in V-MoE model while applying the method separately over the experts in each ensemble group of each MoE encoder in the E^3-MoE model. We implement the pruned model in parallel into two NVIDIA RTX A5000 GPUs for inference and the post-pruning fine-tuning. For post-pruning fine-tuning, as the model size is large, we divide the batch size into half from the original case for CIFAR-10 and CIFAR-100. However, the number of steps is the same. For the same reason, we divide the original batch-size by 32 folds so as the learning rate for ImageNet and hence increase the post-training fine-tuning steps by 32 times of the original. Rest of the hyperparameters are same as in the original fine-tuning process described by the authors.

5.2.1. Results

Generalization performance. Here, we present the generalization performance of the pruned models in terms of the *model pruning ratio*, which is defined as the ratio of the number of parameters pruned to the total number of parameters of the unpruned model. Primarily, we verify the effectiveness of our method compared to the random pruning baseline. As shown in Figure 4a for CIFAR-10, our method can effectively select the task-specific experts as it retains the original performance while the random pruning of experts falls sharply after a small pruning ratio. Figure 4b presents generalization results on CIFAR-10 without post-pruning fine-tuning. As we can see, the method can prune 50% of the experts (35% of the whole model) while maintaining accuracy within 1% of the unpruned model when we do not prune from the penultimate MoE layer. However,

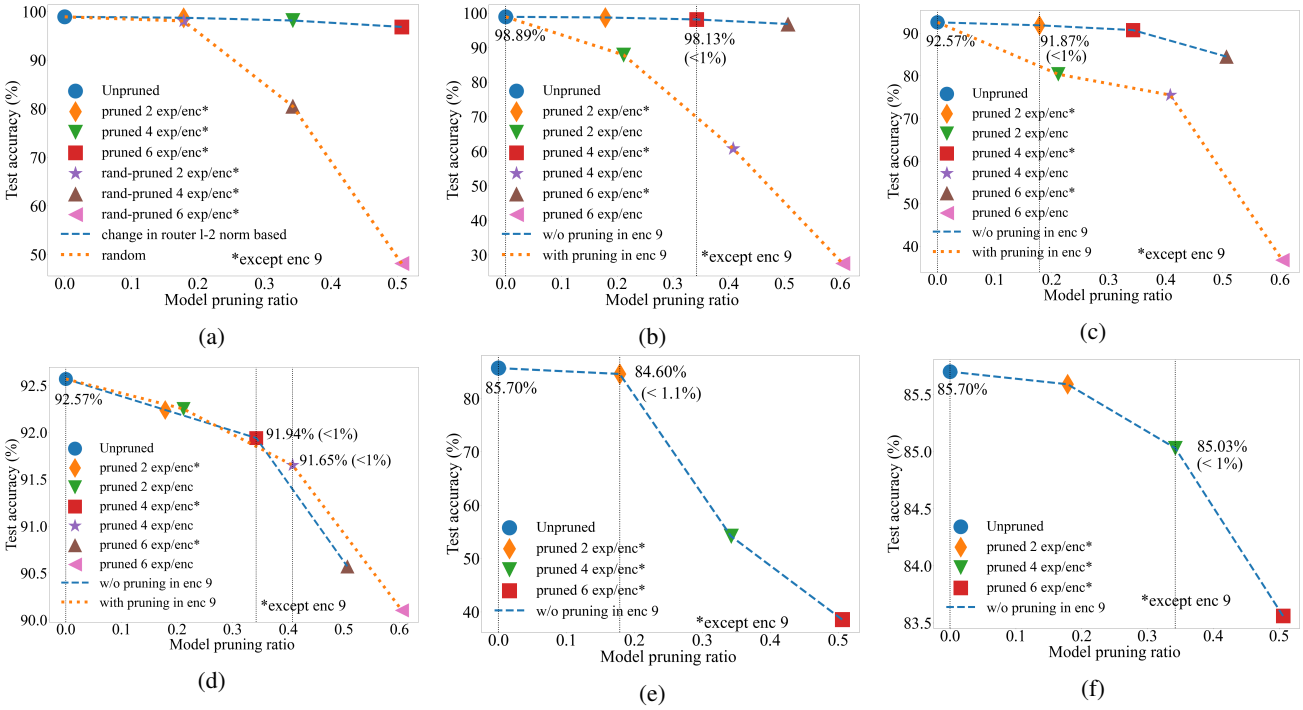


Figure 4: Generalization performance of the pruned **V-MoE** models: (a) Comparison with random pruning on CIFAR-10, (b) On CIFAR-10 **w/o** post-pruning fine-tuning, (c) On CIFAR-100 w/o post-pruning fine-tuning, (d) On CIFAR-100 **with** post-pruning fine-tuning, (e) On ImageNet **w/o** post-pruning fine-tuning, (f) On ImageNet **with** post-pruning fine-tuning

as described in Riquelme et al. (2021), due to the extra sensitivity of the penultimate MoE layer, pruning in the layer is not feasible without post-pruning fine-tuning. However, as shown in Figure 1, we can overcome the limitation by post-pruning fine-tuning and prune up to the maximum possible percentage (75% of the experts, 60% of the whole model)[3].

We found similar results for pruning V-MoE on CIFAR-100 (50% of the experts; 41% of the whole model is pruned) and ImageNet (42% of the experts; 35% of the whole model is pruned) and E^3-MoE on CIFAR-10 and CIFAR-100 (75% of the experts; 45% of the whole model is pruned for both of the datasets). The results for V-MoE on CIFAR-100 and ImageNet are presented in Figure 4c, Figure 4d and Figure 4e, Figure 4f, respectively. The results for E^3-MoE are presented in section B.2 of Appendix. As we can see, when the size of the downstream task goes up (1000 class classification in ImageNet compared to 100 class classification in CIFAR-100 and 10 class classification in CIFAR-10) the maximum allowable pruning ratio goes down as more experts are now important for the larger sized task.

Comparison with other methods. We compare our proposed method of pruning experts based on the *change in router's l_2 norm* with some other potential methods for pruning experts in MoE. Before presenting the results, we briefly describe the methods we consider for comparison as:
(I) **Importance score**. As mentioned in section 1, there are only two recent works (Chen et al., 2022a; Koishekenov et al., 2023) that explored the expert pruning of MoE. Both of them used the *fraction of the total number of tokens received* as the metric to order the experts according to their importance. Specifically, Koishekenov et al. (2023) defines the *importance score* of any expert $s \in [k]$ over the validation set as follows:

$$\text{Importance score}(s) := \text{top1}(s) \times \text{conf}(s)$$

Here, $\text{top1}(s)$ is the fraction of the total tokens received as top-1 tokens by the expert s, and $\text{conf}(s)$ is the confidence value of the expert s, which is essentially the average gating value of the top-1 tokens received by the expert.
We also consider two magnitude-based pruning metrics:
(II) **Router magnitude**. The absolute l_2 norm of the router.
(III) **Average neuron magnitude**. The average value of the l_2 norm of the hidden neurons of the expert.
Similar to the *change of router's l_2 norm* based method proposed in this paper, we also consider
(IV)**Average change of neuron magnitude**. The average value of the change in l_2 norm of the neurons of the experts.

Figure 5a, 5b, 5c and, Figure 5d, 5e, 5f present the comparative results for pruning V-MoE on CIFAR-10 and ImageNet, respectively while the corresponding results on CIFAR-100

[3]pruning more than 6 experts out of 8 experts is not allowed for $l = 2$.

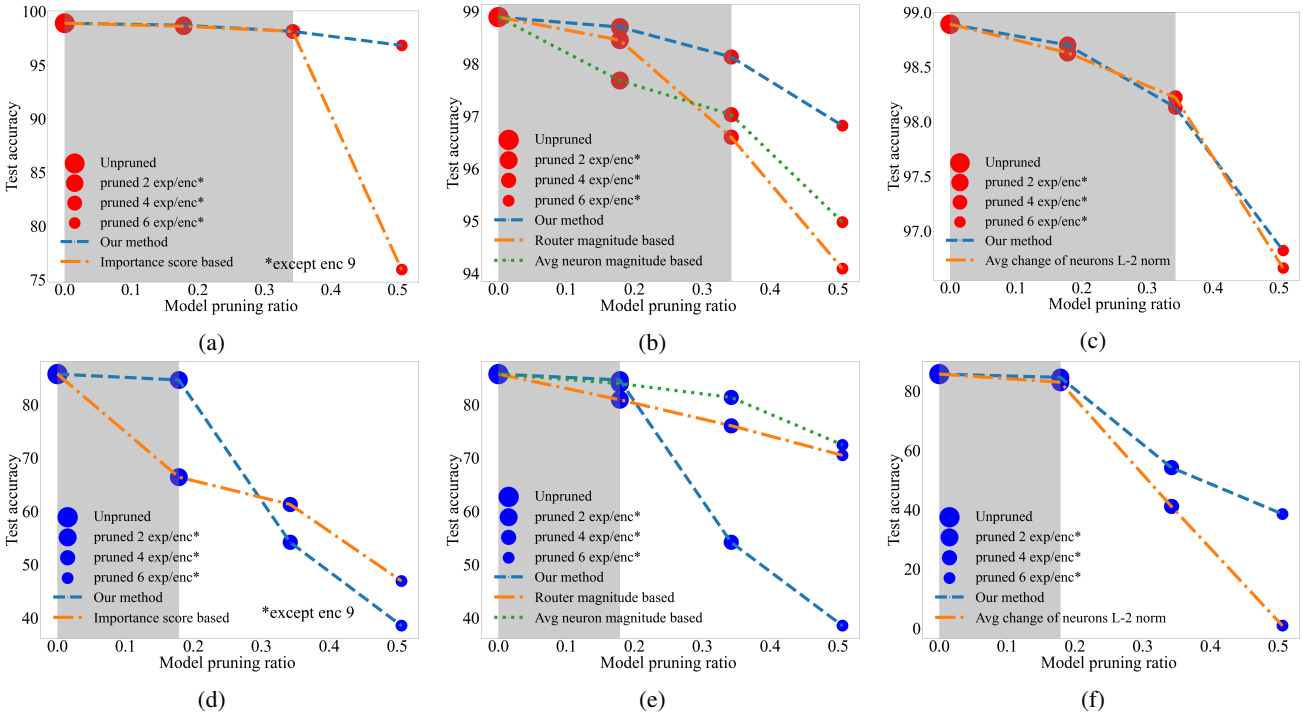


Figure 5: Comparison between different expert pruning methods: (a) vs. importance score on CIFAR-10, (b) vs. absolute magnitude on CIFAR-10, (c) vs. average change-in-neurons-magnitude on CIFAR-10, (d) vs. importance score on ImageNet, (e) vs. absolute magnitude on ImageNet, (f) vs. average change-in-neurons-magnitude on ImageNet

are presented in section B.3 of Appendix. As we can see, the proposed method consistently provides an upper bound to all the other methods described above, at least within the range of acceptable model pruning ratio (model performance is within 1% of the unpruned model, the gray area).

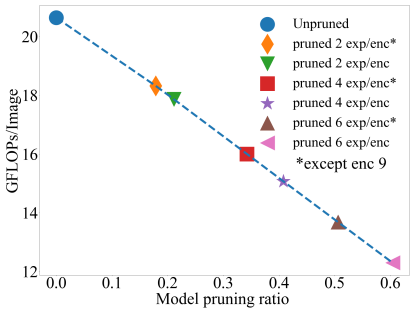


Figure 6: FLOPs per image in V-MoE on CIFAR-10

Inference efficiency. We measure the inference efficiency of the pruned model in terms of FLOPs per image and the inference time. Figure 6 shows the linear reduction of inference FLOPs per image with the increment of pruning ratio in pruning V-MoE on CIFAR-10. By combining the results of this figure with the results in Figure 1, we can infer that the proposed pruning method can reduce 40% of the inference FLOPs. We have similar results showing the linear reduction of inference time (see section B.4 in Appendix for details). The linear reduction of FLOPs and time indicates the prompt implementation of the method without any loss of estimated efficiency (e.g., due to the overhead of the dedicated software kernels or hardware) as our results are generated from the computation on typical modern GPUs without any requirement of special software (e.g., dedicated CUDA kernels). We have similar results for pruning E^3-MoE on CIFAR-100 (linear reduction; 15% reduction of FLOPs and time for the 1% tolerance in accuracy) and pruning V-MoE on ImageNet (linear reduction; 14% reduction of FLOPs and time for the 1% tolerance in accuracy), see section B.4 in appendix.

6. Conclusion

MoE allows faster pre-training of large deep models. However, the inference compute is at the same order as in its dense counterpart. In this paper, we theoretically and empirically investigated pruning irrelevant and redundant experts from the fine-tuned MoE model. We show that pruning experts based on the change in the router's norm can provably maintain the generalization accuracy. The results are verified in the state-of-the-art vision MoE models. Future works include co-implementing the method with other network compression techniques such as neurons pruning, fine-grained structured sparsity, low rank factorization and quantization.

Acknowledgements

This work was supported by IBM through the IBM-Rensselaer Future of Computing Research Collaboration. We thank all anonymous reviewers for their valuable comments and suggestions.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Allen-Zhu, Z. and Li, Y. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 977–988. IEEE, 2022.

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Uuf2q9TfXGA.

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019a.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019b.

Allingham, J. U., Wenzel, F., Mariet, Z. E., Mustafa, B., Puigcerver, J., Houlsby, N., Jerfel, G., Fortuin, V., Lakshminarayanan, B., Snoek, J., Tran, D., Ruiz, C. R., and Jenatton, R. Sparse moes meet efficient ensembles. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=i0ZM36d2qU. Expert Certification, Expert Certification.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Brutzkus, A. and Globerson, A. An optimization and generalization analysis for max-pooling networks. In *Uncertainty in Artificial Intelligence*, pp. 1650–1660. PMLR, 2021.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*, 2018.

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., and Carbin, M. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.

Chen, T., Huang, S., Xie, Y., Jiao, B., Jiang, D., Zhou, H., Li, J., and Wei, F. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*, 2022a.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2022b.

Chen, Z., Deng, Y., Wu, Y., Gu, Q., and Li, Y. Towards understanding the mixture-of-experts layer in deep learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022c. URL https://openreview.net/forum?id=MaYzugDmQV.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Chowdhury, M. N. R., Zhang, S., Wang, M., Liu, S., and Chen, P.-Y. Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6074–6114. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/chowdhury23a.html.

Daniely, A. and Malach, E. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33:20356–20365, 2020.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

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

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

Frantar, E. and Alistarh, D. SparseGPT: Massive language models can be accurately pruned in one-shot. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Fu, H., Chi, Y., and Liang, Y. Guaranteed recovery of one-hidden-layer neural networks via cross entropy. *IEEE transactions on signal processing*, 68:3225–3235, 2020.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., and Dally, W. J. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016a.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016b.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jaiswal, A. K., Liu, S., Chen, T., Ding, Y., and Wang, Z. Instant soup: Cheap pruning ensembles in a single pass can draw lottery tickets from large models. In *International Conference on Machine Learning*, pp. 14691–14701. PMLR, 2023.

Karp, S., Winston, E., Li, Y., and Singh, A. Local signal adaptivity: Provable feature learning in neural networks beyond kernels. *Advances in Neural Information Processing Systems*, 34:24883–24897, 2021.

Koishekenov, Y., Berard, A., and Nikoulina, V. Memory-efficient NLLB-200: Language-specific expert pruning of a massively multilingual machine translation model. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3567–3585, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.198. URL https://aclanthology.org/2023.acl-long.198.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507. Springer, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, Canadian Institute For Advanced Research, 2009.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019a.

Lee, N., Ajanthan, T., and Torr, P. Snip: single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*. Open Review, 2019b.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic

sharding. In *International Conference on Learning Representations*, 2020.

Lewis, M., Bhosale, S., Dettmers, T., Goyal, N., and Zettlemoyer, L. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Li, B., Kong, Z., Zhang, T., Li, J., Li, Z., Liu, H., and Ding, C. Efficient transformer-based large scale language representations using hardware-friendly block structured pruning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3187–3199, 2020a.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2016.

Li, H., Wang, M., Liu, S., Chen, P.-Y., and Xiong, J. Generalization guarantee of training graph convolutional networks with graph topology sampling. In *International Conference on Machine Learning*, pp. 13014–13051. PMLR, 2022.

Li, H., Wang, M., Liu, S., and Chen, P.-Y. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jClGv3Qjhb.

Li, H., Wang, M., Lu, S., Cui, X., and Chen, P.-Y. Training nonlinear transformers for efficient in-context learning: A theoretical learning and generalization analysis. *arXiv preprint arXiv:2402.15607*, 2024a.

Li, H., Zhang, S., Zhang, Y., Wang, M., Liu, S., and Chen, P.-Y. How does promoting the minority fraction affect generalization? a theoretical study of one-hidden-layer neural network on group imbalance. *IEEE Journal of Selected Topics in Signal Processing*, 2024b.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31, 2018.

Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D., and Gonzalez, J. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on machine learning*, pp. 5958–5968. PMLR, 2020b.

Liu, L., Zhang, S., Kuang, Z., Zhou, A., Xue, J.-H., Wang, X., Chen, Y., Yang, W., Liao, Q., and Zhang, W. Group fisher pruning for practical network compression. In *International Conference on Machine Learning*, pp. 7021–7032. PMLR, 2021a.

Liu, Z., Li, F., Li, G., and Cheng, J. Ebert: Efficient bert inference with dynamic structured pruning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4814–4823, 2021b.

Luo, J.-H., Wu, J., and Lin, W. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

Nonnenmacher, M., Pfeil, T., Steinwart, I., and Reeb, D. Sosp: Efficiently capturing global correlations by second-order structured pruning. In *International Conference on Learning Representations*, 2021.

Puigcerver, J., Jenatton, R., Riquelme, C., Awasthi, P., and Bhojanapalli, S. On the adversarial robustness of mixture of experts. *Advances in Neural Information Processing Systems*, 35:9660–9671, 2022.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34: 8583–8595, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Sanh, V., Wolf, T., and Rush, A. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389, 2020.

Shalev-Shwartz, S. et al. Computational separation between convolutional and fully-connected networks. In *International Conference on Learning Representations*, 2020.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Shi, Z., Wei, J., and Liang, Y. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. In *International Conference on Learning Representations*, 2021.

Tung, F. and Mori, G. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7873–7882, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Wohlwend, J., and Lei, T. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6151–6162, 2020.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=Ee277P3AYC.

Zafrir, O., Larey, A., Boudoukh, G., Shen, H., and Wasserblat, M. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754*, 2021.

Zhang, S., Wang, M., Liu, S., Chen, P.-Y., and Xiong, J. Fast learning of graph neural networks with guaranteed generalizability: one-hidden-layer case. In *International Conference on Machine Learning*, pp. 11268–11277. PMLR, 2020a.

Zhang, S., Wang, M., Xiong, J., Liu, S., and Chen, P.-Y. Improved linear convergence of training CNNs with generalizability guarantees: A one-hidden-layer case. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2622–2635, 2020b.

Zhang, S., Wang, M., Chen, P.-Y., Liu, S., Lu, S., and Liu, M. Joint edge-model sparse learning is provably efficient for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhang, S., Li, H., Wang, M., Liu, M., Chen, P.-Y., Lu, S., Liu, S., Murugesan, K., and Chaudhury, S. On the convergence and sample complexity analysis of deep q-networks with ϵ-greedy exploration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pp. 4140–4149. PMLR, 2017.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V. Y., Dai, A. M., Chen, Z., Le, Q. V., and Laudon, J. Mixture-of-experts with expert choice routing. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jdJo1HIVinI.

A. More Details on V-MoE and E^3-MoE

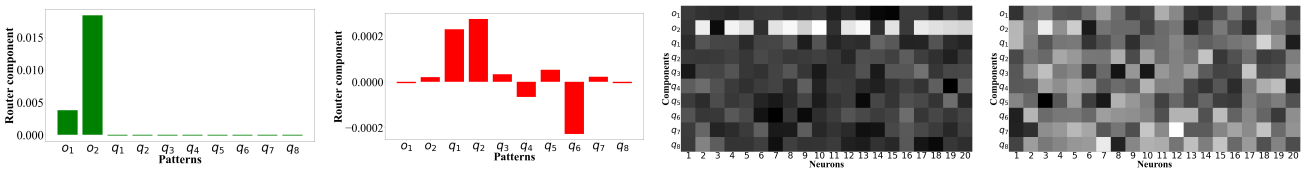
The V-MoE models. The released set of V-MoE models by Riquelme et al. (2021) includes the pre-trained version of the models on ImageNet-21k (Kolesnikov et al., 2020) and the two fine-tuned versions on CIFAR-10 and ImageNet-1k (i.e., ILSVRC2012), respectively. The models contain 12 transformer encoders (encoders 0, 1, ..., and 11). Every odd encoder (i.e., encoders 1, 3, 5, 7, 9, and 11) is the MoE encoder. Each MoE encoder contains 8 experts (i.e., $k = 8$). The input images are divided into 16×16 patches and then transformed into the encoder's hidden dimension of 768 (i.e., $d = 768$). The MoE hidden dimension is 3072 (i.e., $m = 3072$). The CIFAR-10 fine-tuned version contains 65 tokens for an input image (i.e., $n = 65$). The ImageNet fine-tuned version contains 577 tokens for an input image (i.e., $n = 577$). The MoE routers in the model implement token-choice routing where each token is routed to 2 experts (i.e., $l = 2$). The total number of parameters in the model is roughly 979 million. SGD with momentum and cosine learning rate decay is implemented to obtain the fine-tuned models. The fine-tuning steps are 1000 and 10,000 for CIFAR-10 and ImageNet, respectively. More details can be found in Riquelme et al. (2021).

The E^3-MoE models. Allingham et al. (2022) proposed the efficient ensembles of experts in MoE (i.e., E^3-MoE) to improve the generalization performance of V-MoE. The architecture implements the ensembles of experts where, in each MoE encoder the experts are divided into multiple ensembles. The released set of E^3-MoE models includes the pre-trained version on ImageNet and the fine-tuned version on CIFAR-100. The models contain 8 transformer encoders (i.e., encoders 0, 1, ..., and 7), where the last two encoders contain the ensemble of MoEs (i.e., encoders 5 and 7). There are 2 MoE ensembles in the model and each MoE ensembles contains 4 experts per MoE encoder (i.e., $k = 4$). The input images are divided into 32×32 patches and then transformed into encoder's hidden dimension of 512 (i.e., $d = 512$) where the MoE hidden dimension is 2048 (i.e., $m = 2048$). The number of tokens per image in each encoder is 65 (i.e., $n = 65$). The MoE routers implement the token-choice routing, selecting one expert per token (i.e., $l = 1$). Total number of parameters in the model is roughly 167 million. Again, SGD with momentum and cosine learning rate decay is implemented during fine-tuning for 2000 steps.

B. More Experimental Results

B.1. On Synthetic Data

As described in section 5.1, here we present results for experts in the negative group. Figure 7a and 7b present the components for large router (router 20) and small router (router 12), respectively. Figure 7c and 7d present the components for neurons of the corresponding experts.



(a) (b) (c) (d)

Figure 7: Verification of the theoretical findings on synthetic data: (a) Router's components for large router (router 20), (b) Router's components for small router (router 12), (c) Neurons components for expert with large router (expert 20), (d) Neurons components for expert with small router (expert 12) (larger pixel intensity represents larger component of the router weights)

B.2. Generalization Performance On E^3-MoE

We present generalization results of the pruned E^3-MoE models fine-tuned on CIFAR-10 and CIFAR-100 in Figure 8a, 8b and Figure 8c, 8d, respectively. As we can see and as reported in the main paper, similar results are found for the pruned E^3-MoE models as for the pruned V-MoE models. More specifically, for the pruned E^3-MoE, 75% of the experts and 45% of the whole model can be pruned for both of the datasets while maintaining generalization performance within 1% of the unpruned model. We do not test the generalization performance of the pruned E^3-MoE on ImageNet as the released E^3-MoE model is pre-trained on ImageNet itself.

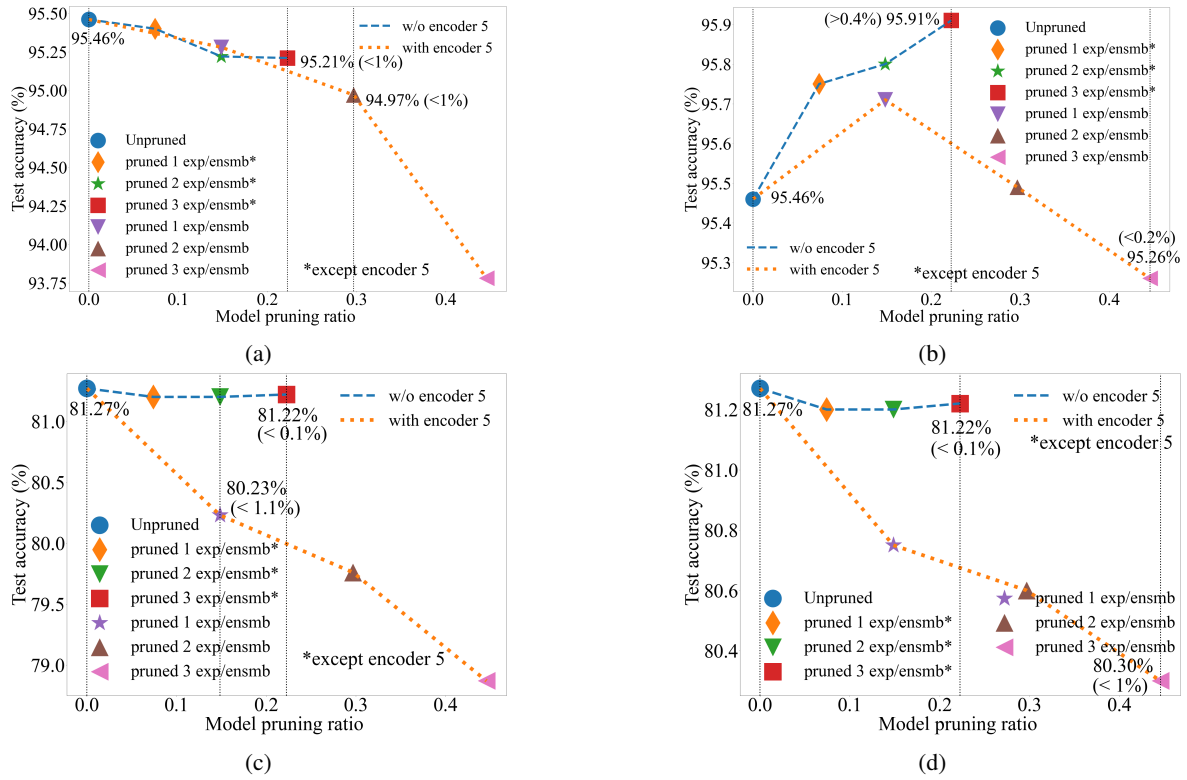


Figure 8: Generalization performance of the pruned **E^3-MoE** models: (a) On CIFAR-10 **without** post-pruning fine-tuning, (b) On CIFAR-10 with post-pruning fine-tuning, (c) On CIFAR-100 **without** post-pruning fine-tuning, (d) On CIFAR-100 **with** post-pruning fine-tuning

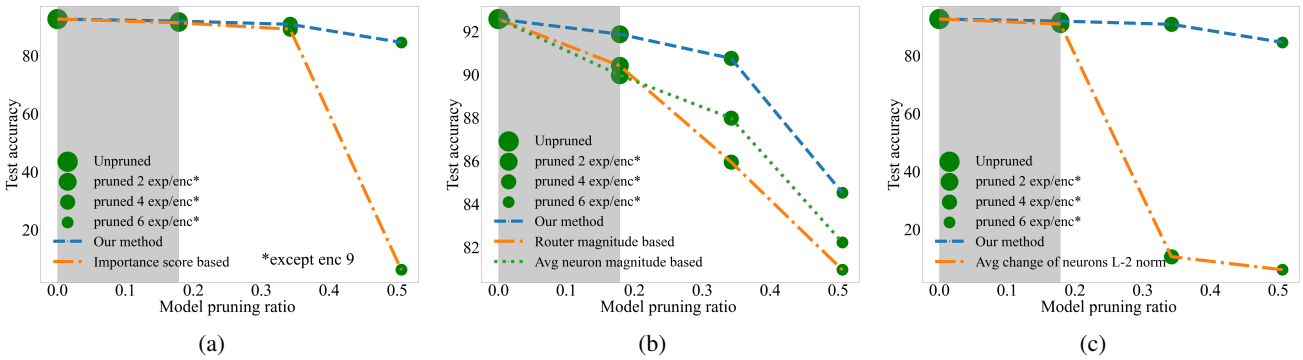


Figure 9: Comparison between different expert pruning methods on CIFAR-100 for pruning V-MoE: (a) vs. importance score, (b) vs. absolute magnitude, (c) vs. average change-in-neurons-magnitude

B.3. Comparison with Other Methods for Pruning V-MoE on CIFAR-100

Figure 9a, 9b and, 9c presents the comparative results for pruning V-MoE on CIFAR-100 among different expert pruning methods. The proposed method outperforms all other methods for this dataset.

B.4. On the Inference Efficiency of V-MoE and E^3-MoE

We present the results in Figure 10a for inference time in predicting the whole CIFAR-10 test set by the pruned V-MoE. As we described in section 5, inference time falls linearly with pruning ratio. We also present results for linear reduction of FLOPs and inference time in the pruned V-MoE on CIFAR-100 and ImageNet in Figure 10b, 10c and Figure 10d, 10e

and, in the pruned E^3-MoE on CIFAR-10 and CIFAR-100 in Figure 10f,10g and Figure 10h,10i, respectively. The slight sublinearity in Figure 10g and 10i occurs from the parallel implementation of ensembles. The relatively lower percentage of reduction of FLOPs and time in E^3-MoE compared to V-MoE occurs from the fact that the unpruned E^3-MoE is much smaller than the unpruned V-MoE in terms of parameters (167 million compared to 979 million).

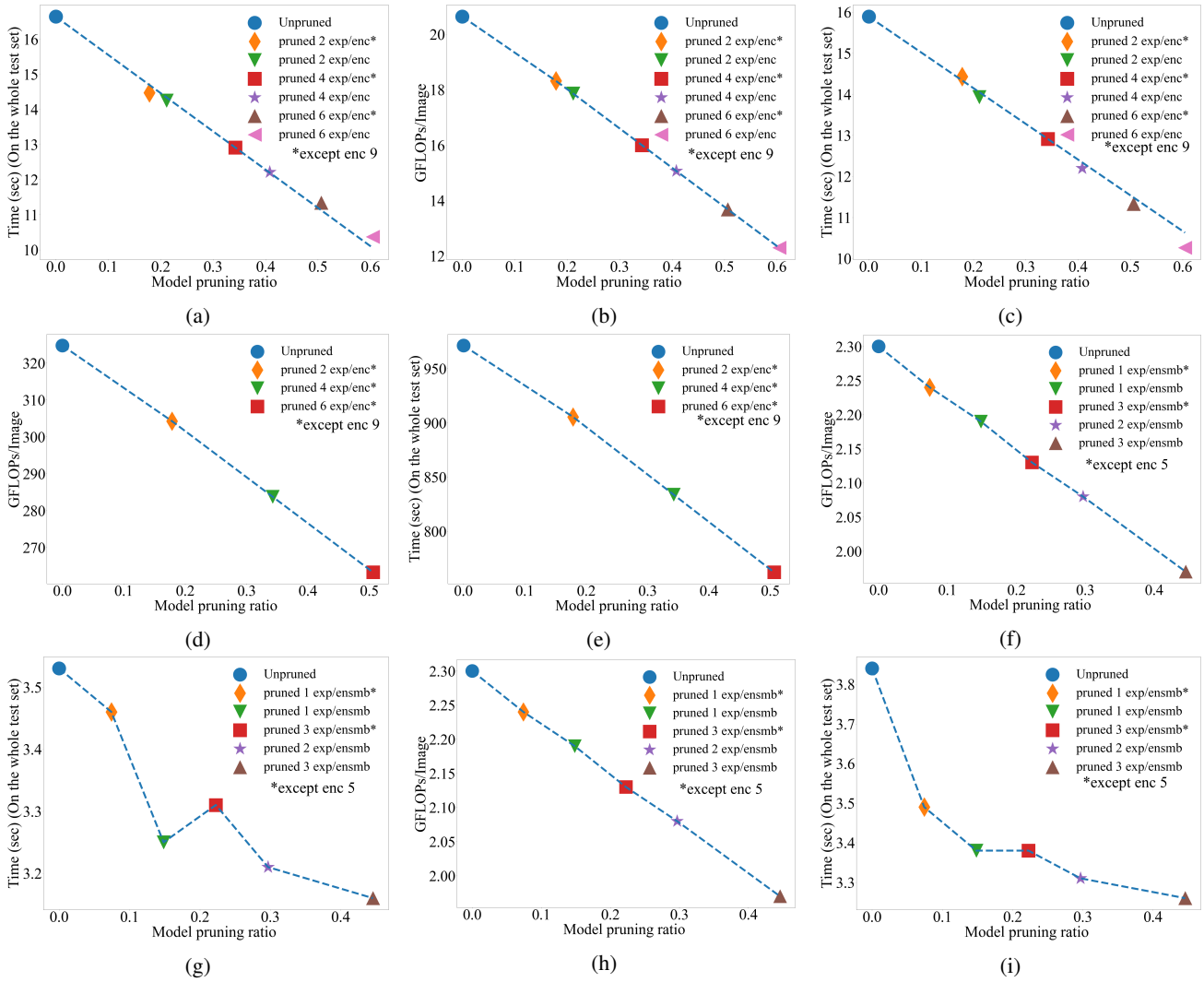


Figure 10: Inference cost of the pruned models: (a) Inference time in V-MoE on CIFAR-10, (b) FLOPs per Image in V-MoE on CIFAR-100, (c) Inference time in V-MoE on CIFAR-100, (d) FLOPs per Image in V-MoE on ImageNet, (e) Inference time in V-MoE on ImageNet, (f) FLOPs per Image in E^3-MoE on CIFAR-10, (g) Inference time in E^3-MoE on CIFAR-10, (h) FLOPs per Image in E^3-MoE on CIFAR-100, (i) Inference time in E^3-MoE on CIFAR-100

C. Preliminaries

As we analyze the *expert-choice* routing, for any fine-tuning step t, we re-write (7) as,

$$f^{(t)}(x) = \sum_{s=1}^{k} a^{(s)} \sum_{j \in J_s^{(t)}(x)} G_j^{(s,t)} \sum_{r=1}^{m} \mathrm{ReLU}\left(\langle w_r^{(s,t)}, x^{(j)} \rangle\right)$$

For any input (x, y) and iteration t, the gradient of the hidden neuron $r \in [m]$ of the expert $s \in [k]$ is calculated as,

$$\frac{\partial l^{(t)}(x,y)}{\partial w_r^{(s,t)}} = -y a^{(s)} \sum_{j \in J_s^{(t)}(x)} G_j^{(s,t)} x^{(j)} 1_{\langle w_r^{(s,t)}, x^{(j)} \rangle \geq 0}$$

and the gradient of the router of the expert $s \in [k]$ is calculated as,

$\frac{\partial l^{(t)}(x,y)}{\partial w_s^{(t)}} = -ya^{(s)} \sum_{j \in J_s^{(t)}(x)} \sigma_j^{(s,t)} G_j^{(s,t)} \sum_{i \in J_s^{(t)}(x) \setminus j} G_i^{(s,t)} (x^{(j)} - x^{(i)})$, where, $\sigma_j^{(s,t)} = \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,t)}, x^{(j)} \rangle)$

We express the batch gradient of SGD at iteration t for the batch $\mathcal{B}_t$ as, $\frac{\partial l}{\partial w_r^{(s,t)}} = \frac{1}{B} \sum_{x \in \mathcal{B}_t} \frac{\partial l^{(t)}(x,y)}{\partial w_r^{(s,t)}}$ for the expert and, $\frac{\partial l}{\partial w_s^{(t)}} = \frac{1}{B} \sum_{x \in \mathcal{B}_t} \frac{\partial l^{(t)}(x,y)}{\partial w_s^{(t)}}$ for the router.

We present the pruning algorithm considered for analysis at Algorithm 1[4].

Algorithm 1 The Expert Pruning Algorithm for the Theoretical Analysis

Input : Training data $\{(x_i, y_i)\}_{i=1}^N$, learning rates η_r and η_e, number of iterations T and T', batch-size B, expert-pruning-ratio ρ

Step-1: Initialize the pre-trained weights $\{w_s^{(0)}, w_r^{(s,t)}, a^{(s)}\}_{s \in [k], r \in [m]}$

Step-2: for $t = 0, 1, ..., T-1$ do:

$$w_s^{(t+1)} = w_s^{(t)} - \eta_r \frac{\partial l}{\partial w_s^{(t)}}, \forall s \in [k]$$

$$w_{r,s}^{(t+1)} = w_{r,s}^{(t)} - \eta_e \frac{\partial l}{\partial w_{r,s}^{(t)}}, \quad \forall r \in [m], s \in [k]$$

Step-3: Construct the set $S_{k'} = \{s \in S_1 : \Delta_s^{(T)} \in \text{TOP}_{(1-\rho)|S_1|}(\{\Delta_s\}_{s \in S_1})\} \bigcup \{s \in S_2 : \Delta_s^{(T)} \in \text{TOP}_{(1-\rho)|S_2|}(\{\Delta_s\}_{s \in S_2})\}$

Step-4: for $t = 0, 1, ..., T'-1$ do:

$$w_s^{(t+1)} = w_s^{(t)} - \eta_r \frac{\partial l}{\partial w_s^{(t)}}, \forall s \in S_{k'}$$

$$w_{r,s}^{(t+1)} = w_{r,s}^{(t)} - \eta_e \frac{\partial l}{\partial w_{r,s}^{(t)}}, \quad \forall r \in [m], s \in S_{k'}$$

Notations:

1. $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ hides factor $\log(\text{poly}(d))$ with a sufficiently large polynomial $poly(\cdot)$
2. With high probability (abbreviated as *w.h.p.*) refers to the probability $1 - \frac{1}{\text{poly}(d)}$

Definitions:

We denote,

- $G_1^{(s,t)}(x)$: Gating value of o_1 for the sample x at the expert s at time t
- $G_2^{(s,t)}(x)$: Gating value of o_2 for the sample x at the expert s at time t
- $G_q^{(s,t)}(x)$: Gating value of the task-irrelevant pattern $q \in \mathcal{P}$ for the sample x at the expert s at time t
- $l_q^{(s,t)}(x) := |\{j \in J_s^{(t)}(x) : x^{(j)} = q\}|$, is the number of copies of the task-irrelevant pattern $q \in \mathcal{P}$ in the set of top l tokens for the sample x at the expert s at time t

We restate the definition of the set, $S_1 := \{s \in [k] : a^{(s)} = +1\}$ and $S_2 := \{s \in [k] : a^{(s)} = -1\}$

[4]Here, $TOP_z(S)$ function provides the set of top z values of the set S.

We restate the definition of the probability measures,

$p_1^{(s,t)} := \mathbb{P}\left[(x,y) \sim \mathcal{D} : \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_1 \text{ and } G_1^{(s,0)} \geq \frac{1}{l} | y = +1\right]$

and $p_2^{(s,t)} := \mathbb{P}\left[(x,y) \sim \mathcal{D} : \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_2 \text{ and } G_2^{(s,0)} \geq \frac{1}{l} | y = -1\right]$

For any x, $\sigma_j^{(s,t)}(x) := \sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,t)}, x^{(j)} \rangle)$ and, for any pattern $q \in \mathcal{P}$, $\sigma_q^{(s,t)} := \sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,t)}, q \rangle)$

$\sigma_1^{(s,t)} := \sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,t)}, o_1 \rangle)$ and $\sigma_2^{(s,t)} := \sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,t)}, o_2 \rangle)$

For any task-irrelevant pattern $q \in \mathcal{P}$, $\delta_{1,q}^{(s,t)}(x) := \sigma_1^{(s,t)} - \sigma_q^{(s,t)}$ and $\delta_{2,q}^{(s,t)}(x) := \sigma_2^{(s,t)} - \sigma_q^{(s,t)}$. Therefore, $\delta_{q,1}^{(s,t)} = -\delta_{1,q}^{(s,t)}$ and $\delta_{q,2}^{(s,t)} = -\delta_{2,q}^{(s,t)}$

For any two different task-irrelevant patterns $q, q' \in \mathcal{P}$, $\delta_{q,q'}^{(s,t)} := \sigma_q^{(s,t)} - \sigma_{q'}^{(s,t)}$

Note that, for any x and q, $l_q^{(s,t)}(x) G_q^{(s,t)} \leq 1$.

Let us define, $C_1 := \max\{\left|\left|w_s^{(0)}\right|\right|\}_{s \in [k]}$ and $C_2 := \max\{\left|\left|w_r^{(s,0)}\right|\right|\}_{s \in [k], r \in [m]}$.

WLOG, we analyze the case where $l \geq e^{2C_1}$. Therefore, $\forall s \in [k]$, $\left|\left|w_s^{(0)}\right|\right| \leq \frac{1}{2} \log l$.

Assumptions on the pre-trained model. Based on the experts' proficiency measure described in section 4.2, we define the important and unimportant experts for the downstream task in the pre-trained model.

Definition C.1 (Important and Unimportant experts in pre-trained model[5])**.** An expert $s \in [k]$ is *important* for the class-1 task-specific feature o_1 if $p_1^{(s,0)} = \Omega(1)$, and *unimportant* for the feature if $p_1^{(s,0)} = O(1/d)$. Similarly, an expert $s \in [k]$ is *important* for the class-2 task-specific feature o_2 if $p_2^{(s,0)} = \Omega(1)$, and *unimportant* for the feature if $p_2^{(s,0)} = O(1/d)$.

We assume the existence of at least one class-1 important expert in the set S_1 and at least one class-2 important expert in the set S_2 of the pre-trained model. This assumption is minimal as we need only one important expert per class.

For any class-1 important expert s, we assume that the router's component along o_1 (i.e., $\langle w_s^{(0)}, o_1 \rangle$) is minimally separated ($\Omega(1/d)$) from the components along any irrelevant pattern $q \in \mathcal{P}$ (i.e., $\langle w_s^{(0)}, q \rangle$), i.e.,
$\left|\langle w_s^{(0)}, o_1 - q \rangle\right| = C_p$ where, $C_p = \Theta(1/d)$, for any s such that $p_1^{(s,0)} = \Omega(1)$ and, for any $q \in \mathcal{P}$.

We further assume that, for any class-1 important expert s, at least a fraction of the m neurons (i.e., $\Omega(1)$ fraction) of the expert are minimally activated (i.e., $\langle w_r^{(s,0)}, o_1 \rangle \geq 0$) by o_1, i.e.,
$\left|\{r \in [m] : \langle w_r^{(s,0)}, o_1 \rangle \geq 0\}\right| = \Omega(1)$, for any s such that $p_1^{(s,0)} = \Omega(1)$.

We assume that similar assumptions hold for the class-2 important experts. Again, these two assumptions are minimal for an expert to be important for a task-specific pattern.

Components of the Router-gradients:

We calculate the components of the gradient of the router $s \in [k]$ for the input (x, y) along the task-specific patterns, i.e., o_1 and o_2 and along any task-irrelevant pattern $q \in \mathcal{P}$ at time t as follows:

[5]Our definitions of important and unimportant experts are minimal in the sense that (I) for an expert to be important for a particular downstream pattern, it should learn to route the pattern with at least the uniform gating value ($1/l$) for a minimal fraction of samples ($p_1^{(s,0)} = \Omega(1)$) during pre-training and, (II) if an expert does not learn to route the pattern at all ($p_1^{(s,0)} = O(1/d)$), it should be unimportant for the pattern.

$$\left\langle \frac{\partial l(x,y)}{\partial w_s^{(t)}}, q \right\rangle = \begin{cases} 0 & \text{if } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = q \\ y a^{(s)} l_q^{(s,t)} G_q^{(s,t)} \sum_{j \in J_s^{(t)}(x)/\{j: x^{(j)}=q\}} G_j^{(s,t)} \left(\sigma_j^{(s,t)} - \sigma_q^{(s,t)}\right) & \text{if } \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = q \end{cases} \tag{14}$$

$$\left\langle \frac{\partial l(x,y)}{\partial w_s^{(t)}}, o_1 \right\rangle = \begin{cases} 0 & \text{if } y = -1 \\ 0 & \text{if } y = +1 \text{ but } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_1 \\ a^{(s)} G_1^{(s,t)}(x) \sum_{i \in J_s^{(t)}(x)/\{j: x^{(j)}=o_1\}} G_j^{(s,t)} \left(\sigma_j^{(s,t)} - \sigma_1^{(s,t)}\right) & \text{if } y = +1 \text{ and } \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_1 \end{cases} \tag{15}$$

$$\left\langle \frac{\partial l(x,y)}{\partial w_s^{(t)}}, o_2 \right\rangle = \begin{cases} 0 & \text{if } y = +1 \\ 0 & \text{if } y = -1 \text{ but } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_2 \\ -a^{(s)} G_2^{(s,t)}(x) \sum_{i \in J_s^{(t)}(x)/\{j: x^{(j)}=o_2\}} G_j^{(s,t)} \left(\sigma_j^{(s,t)} - \sigma_2^{(s,t)}\right) & \text{if } y = -1 \text{ and } \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_2 \end{cases} \tag{16}$$

Components of the Neuron-gradients:

We calculate the components of the gradient of the hidden neuron $r \in [m]$ of the expert $s \in [k]$ for the input (x, y) along the task-specific patterns, i.e., o_1 and o_2 and along any task-irrelevant pattern $q \in \mathcal{P}$ at time t as follows:

$$\left\langle \frac{\partial l(x,y)}{\partial w_r^{(s,t)}}, q \right\rangle = \begin{cases} 0 & \text{if } \langle w_r^{(s,t)}, q \rangle < 0 \\ 0 & \text{if } \langle w_r^{(s,t)}, q \rangle \geq 0 \text{ but } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = q \\ -y a^{(s)} l_q^{(s,t)} G_q^{(s,t)} & \text{if } \langle w_r^{(s,t)}, q \rangle \geq 0, \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = q \end{cases} \tag{17}$$

$$\left\langle \frac{\partial l(x,y)}{\partial w_r^{(s,t)}}, o_1 \right\rangle = \begin{cases} 0 & \text{if } y = -1 \\ 0 & \text{if } y = +1 \text{ but } \langle w_r^{(s,t)}, o_1 \rangle < 0 \\ 0 & \text{if } y = +1 \text{ and } \langle w_r^{(s,t)}, o_1 \rangle \geq 0 \text{ but } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_1 \\ -a^{(s)} G_1^{(s,t)}(x) & \text{if } y = +1, \langle w_r^{(s,t)}, o_1 \rangle \geq 0 \text{ and } \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_1 \end{cases} \tag{18}$$

$$\left\langle \frac{\partial l(x,y)}{\partial w_r^{(s,t)}}, o_2 \right\rangle = \begin{cases} 0 & \text{if } y = +1 \\ 0 & \text{if } y = -1 \text{ but } \langle w_r^{(s,t)}, o_2 \rangle < 0 \\ 0 & \text{if } y = -1 \text{ and } \langle w_r^{(s,t)}, o_2 \rangle \geq 0 \text{ but } \nexists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_2 \\ a^{(s)} G_2^{(s,t)}(x) & \text{if } y = -1, \langle w_r^{(s,t)}, o_2 \rangle \geq 0 \text{ and } \exists j \in J_s^{(t)}(x) \text{ s.t. } x^{(j)} = o_2 \end{cases} \tag{19}$$

D. Proof of Lemma 4.1

Lemma D.1 (Full version of the **Lemma 4.1**). *Suppose the expert learning rate η_e such that, the router learning rate $\eta_r = O\left(\frac{\eta_e C_p}{m d l^2 C_2^2}\right)$, the batch-size $B = \tilde{\Omega}(l^2 d^2)$, and the number of iterations $T = \Omega\left(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}}\right)$. Then,*

For any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, we have

(i) $p_1^{(s,T)} = 1$,
(ii) for every $(x, +1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_1$,
(iii) $\langle w_r^{(s,T)}, o_1 \rangle = \Omega(l C_2 \sqrt{d \log l})$, for a constant fraction $r \in [m]$,
(iv) $\Delta_s^{(T)} > \frac{3}{2}\log l$.

and, for any expert $s \in S_2$ such that $p_2^{(s,0)} = \Omega(1)$, we have

(v) $p_2^{(s,T)} = 1$,
(vi) for every $(x, -1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_2$,
(vii) $\langle w_r^{(s,T)}, o_2 \rangle = \Omega(l C_2 \sqrt{d \log l})$, for a constant fraction $r \in [m]$,
(viii) $\Delta_s^{(T)} > \frac{3}{2}\log l$.

Proof. For any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, from Lemma E.4, for any task-irrelevant pattern q, $\delta_{1,q}^{(s,T'')} = \Omega\left(mC_2\right)$ and $|\delta_{(2,q)}^{(s,T'')}| = O(mC_2)$. Furthermore, for any different two task-irrelevant pattern q and q', $|\delta_{q,q'}^{(s,T'')}| = O(mC_2)$

Therefore, $\langle \frac{\partial l}{\partial w_s^{(T'')}}, o_1 \rangle \leq -\Omega(\frac{mC_2}{l}) + \tilde{O}(\frac{mC_2}{l\sqrt{B}})$,
as from Lemma E.4, $p_1^{(s,T'')} = p_1^{(s,0)}$, $\forall q$ s.t. $\langle w_s^{(T'')}, q \rangle < \langle w_s^{(T'')}, o_1 \rangle$; $\langle w_s^{(T'')}, o_1 - q \rangle < 2\log l$ which implies from Lemma E.6 that, $\forall (x, y = +1) \sim \mathcal{D}$ s.t. $\exists j \in J_s^{(T'')}(x)$ with $x^{(j)} = q$, $G_1^{(s,T'')}(x)(1 - G_1^{(s,T'')}(x)) \geq \frac{1}{4l}$.

Therefore, $\langle w_s^{(T''+1)}, o_1 \rangle \geq \langle w_s^{(T'')}, o_1 \rangle + \Omega(\frac{mC_2}{l}\eta_r) - \tilde{O}(\frac{mC_2}{l\sqrt{B}}\eta_r)$.

Now, for any q, $\langle \frac{\partial l}{\partial w_s^{(T'')}}, q \rangle \geq -O(\frac{mC_2}{d}) - \tilde{O}(\frac{mC_2}{\sqrt{B}})$.

Therefore, for any q, $\langle w_s^{(T''+1)}, q \rangle \leq \langle w_s^{(T'')}, q \rangle + O(\frac{mC_2}{d}\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)$.

Therefore, for any q, $\langle w_s^{(T''+1)}, o_1 - q \rangle \geq \langle w_s^{(T'')}, o_1 - q \rangle + \Omega(\frac{mC_2}{l}\eta_r)$ as we are selecting $B = \tilde{\Omega}(l^2 d^2)$.

Now, from Lemma E.4, for any q s.t. $\langle w_s^{(T'')}, q\rangle < \langle w_s^{(T'')}, o_1\rangle$, $\langle w_s^{(T'')}, o_1 - q\rangle = \Omega(C_p)$, which implies $\langle w_s^{(T''+1)}, o_1 - q\rangle \geq \Omega(C_p) + \Omega(\frac{mC_2}{l}\eta_r)$.

Therefore, for any q s.t. $\langle w_s^{(T'')}, q\rangle < \langle w_s^{(T'')}, o_1\rangle$, $\langle w_s^{(T''+1)}, o_1\rangle > \langle w_s^{(T''+1)}, q\rangle$ and hence $p_1^{(s,T''+1)} \geq p_1^{(s,T'')} = p_1^{(s,0)}$.

Now, as $p_1^{(s,T'')} = p_1^{(s,0)}$, using the same procedure as in the proof of Lemma E.4, $\forall q$, $\delta_{1,q}^{(s,T''+1)} = \delta_{1,q}^{(s,T'')} + \Omega(\frac{m\eta_e}{l})$ and, $\delta_{2,q}^{(s,T''+1)} \leq \delta_{2,q}^{(s,T'')} = O(mC_2)$. Furthermore, for any two different task-irrelevant patterns, $|\delta_{q,q'}^{(s,T''+1)} - \delta_{q,q'}^{(s,T'')}| = O(\frac{m}{d}\eta_e)$.

Hence, by induction, $\forall t \geq T''$ such that, for any q with $\langle w_s^{(t)}, q\rangle < \langle w_s^{(t)}, o_1\rangle$ satisfies $\langle w_s^{(t)}, o_1 - q\rangle \leq 2\log l$, $p_1^{(s,t)} \geq p_1^{(s,0)}$ and hence,

for all q,

$$\delta_{1,q}^{(s,t+1)} - \delta_{1,q}^{(s,t)} = \Omega(\frac{m\eta_e}{l}) \tag{20}$$

$\delta_{2,q}^{(s,t)} = O(mC_2)$, $|\delta_{q,q'}^{(s,t)}| = O(mC_2)$ and hence for all q,

$$\langle w_s^{(t+1)}, o_1 - q\rangle \geq \langle w_s^{(T'')}, o_1 - q\rangle + \Omega(\frac{1}{l}mC_2\eta_r)(t - T'') + \Omega(\frac{1}{l^2}m\eta_e\eta_r)\frac{(t-T'')^2 - (t-T'')}{2} \tag{21}$$

Now, from Lemma E.4, for all q, $\langle w_s^{(T'')}, o_1 - q\rangle < \log l$. Let us assume that upto T iterations, for all q, $\langle w_s^{(T)}, o_1 - q\rangle \leq 2\log l$.

Now, if $\exists q'$ s.t. $\langle w_s^{(T)}, o_1 - q'\rangle \leq \log l$, then $\forall q \neq q'$,

$$\langle w_s^{(T)}, o_1 - q\rangle \leq \langle w_s^{(T)}, o_1 - q'\rangle + \langle w_s^{(T)}, q' - q\rangle \leq \langle w_s^{(T)}, q' - q\rangle + \log l \tag{22}$$

On the other hand, from inequality (21), by selecting $T = \Omega(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$ we get,
$\langle w_s^{(T+1)}, o_1 - q'\rangle > \log l$.

Now, $\forall q$, and $\forall t$, $\delta_{1,q}^{(s,t)} - \delta_{1,q}^{(s,t-1)} = O(m\eta_e)$ and $\delta_{2,q}^{(s,t-1)} - \delta_{2,q}^{(s,t)} = O(m\eta_e)$ and $|\delta_{q,q'}^{(s,t-1)} - \delta_{q,q'}^{(s,t)}| = O(\frac{m}{d}\eta_e)$.

Hence, $\langle w_s^{(t+1)}, q\rangle \geq \langle w_s^{(0)}, q\rangle - O(\frac{mC_2}{d}\eta_r)t - O(\frac{m}{d}\eta_e\eta_r)\frac{t^2 - t}{2}$.

On the other hand, $\langle w_s^{(t+1)}, q'\rangle \leq \langle w_s^{(0)}, q'\rangle + O(\frac{mC_2}{d}\eta_r)t$.

Therefore,

$$\langle w_s^{(t+1)}, q' - q\rangle \leq \langle w_s^{(0)}, q' - q\rangle + O(\frac{mC_2}{d}\eta_r)t + O(\frac{m}{d}\eta_e\eta_r)t^2 \tag{23}$$

Now, for the choice of $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$ and as $\langle w_s^{(0)}, q' - q\rangle \leq \frac{1}{2}\log l$, using inequality (23) we get,

$$\langle w_s^{(T)}, q' - q\rangle \leq \frac{1}{2}\log l + O(\frac{l^2}{d}\log l) \leq \log l$$

Therefore, from inequality (22), for the choice of $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$, $\forall q$, $\langle w_s^{(T)}, o_1 - q\rangle \leq 2\log l$ holds.

Now, as for any q, $\langle w_s^{(T+1)}, o_1 - q\rangle > \log l$, $p_1^{(s,T+1)} = 1$ and $\forall (x, y = +1) \sim \mathcal{D}$, $G_1^{(s,T+1)}(x) > \frac{1}{2}$.

Now, as for any t, $p_1^{(s,T'')} \geq p_1^{(s,0)}$, at least for $\Omega(1)$ fraction of $r \in [m]$ of s, $\langle w_r^{(s,t)}, o_1\rangle \geq \langle w_r^{s,0}, o_1\rangle + \Omega\left(\frac{\eta_e}{l}t\right)$.

Therefore, for $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$, $\langle w_r^{(s,T)}, o_1\rangle = \Omega(lC_2\sqrt{d\log l})$ for $\Omega(1)$ fraction of $r \in [m]$ of s.

Now, $\left|\left|w_s^{(T+1)}\right|\right| = \left(\langle w_s^{(T+1)}, o_1\rangle^2 + \langle w_s^{(T+1)}, o_2\rangle^2 + \sum_{i=1}^{d-2}\langle w_s^{(T+1)}, q_i\rangle^2\right)^{\frac{1}{2}} \geq \langle w_s^{(T+1)}, o_1\rangle > 2\log l$ where, q_i denotes the i-th task-irrelevant pattern. Now, as $\left|\left|w_s^{(0)}\right|\right| \leq \frac{1}{2}\log l$, $\Delta_s^{(T)} > \frac{3}{2}\log l$.

Similarly, using Lemma E.5, for $s \in S_2$ such that, $p_2^{(s,0)} = \Omega(1)$, we can proof the statements (v), (vi), (vii) and (viii). □

E. Lemmas Used to Prove the Lemma 4.1

Lemma E.1. *Let, $S \subset \mathcal{D}$ such that $p := \mathbb{P}\left[(x, y) \sim D : (x, y) \in S\right]$. Then, w.h.p. over any randomly sampled batch $\mathcal{B}_t$ of size B at the iteration t, $\left|\left|\mathcal{B}_t \cap S\right| - Bp\right| = \tilde{O}\left(\sqrt{B}\right)$.*

Proof. Let us define a random variable X associated with any sample $(x, y) \sim \mathcal{D}$ such that,
$X := \begin{cases} 1 & \text{if } (x, y) \in S \\ 0 & \text{if } (x, y) \notin S \end{cases}$

Therefore, $X \sim \text{Ber}(p)$.

Now, for any randomly sampled batch $\mathcal{B}_t := \{(x_1, y_1), (x_2, y_2), ..., (x_B, y_B)\}$ of size B, we can denote the B i.i.d. random variables following the same distribution as X by $X_1, X_2, ..., X_B$ corresponding to the B samples of the batch, respectively.

Therefore, $\left|\mathcal{B}_t \cap S\right| = \sum_{i=1}^{B} X_i$.

Now, $\mathbb{E}\left[\left|\mathcal{B}_t \cap S\right|\right] = \sum_{i=1}^{B}\mathbb{E}\left[X_i\right] = Bp$.

Therefore, using the Hoeffding's inequality, $\mathbb{P}\left[\left|\left|\mathcal{B}_t \cap S\right| - Bp\right| = \tilde{O}\left(\sqrt{B}\right)\right] \geq 1 - \frac{1}{\text{poly}(d)}$ which completes the proof. □

Lemma E.2. *For any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, w.h.p. over a randomly sampled batch of size B we can ensure that,*

(i) $\left|\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle\right| \leq O\left(\frac{mC_2}{d}\right) + \tilde{O}\left(\frac{mC_2}{\sqrt{B}}\right)$

(ii) $\left|\langle \frac{\partial l}{\partial w_s^{(0)}}, o_1\rangle\right| \leq O\left(mC_2\right) + \tilde{O}\left(\frac{mC_2}{\sqrt{B}}\right)$

(iii) $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1\rangle \leq -\frac{1}{2l} + \tilde{O}\left(\frac{1}{l\sqrt{B}}\right)$ *and* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1\rangle \geq -\frac{1}{2} - \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$

(iv) $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2\rangle \geq 0$ *and* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2\rangle \leq \frac{1}{2} + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$

(v) $\left|\langle \frac{\partial l}{\partial w_r^{(s,0)}}, q\rangle\right| \leq O\left(\frac{1}{d}\right) + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$

Proof. For any task-irrelevant pattern q,

$|\delta_{1,q}^{(s,0)}| = |\sigma_1^{(s,0)} - \sigma_q^{(s,0)}| = |\sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,0)}, o_1\rangle) - \sum_{r=1}^m \text{ReLU}(\langle w_r^{(s,0)}, q\rangle)| = O(mC_2)$

Similarly, $|\delta_{2,q}^{(s,0)}| = O(mC_2)$ and for any two different task-irrelevant patterns $q, q' \in \mathcal{P}$, $\left|\delta_{q,q'}^{(s,t)}\right| = O(mC_2)$.

We denote $\mathcal{B}_0$ as the randomly sampled batch before the first update of SGD.

(i)

$$\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle = \frac{1}{B}\sum_{x\in\mathcal{B}_0} \langle \frac{\partial l(x,y)}{\partial w_s^{(0)}}, q\rangle$$

Let us define the set $S_q := \{(x,y) \sim \mathcal{D} : \exists j \in J_s^{(0)} \text{ s.t. } x^{(j)} = q\}$. Then, $p_q^{(s,0)} := \mathbb{P}\left[(x,y) \sim \mathcal{D} : (x,y) \in S_q\right]$

Here, $p_q^{(s,0)} = O(\frac{1}{d})$

Therefore, $\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle = \frac{1}{B}\sum_{x\in\mathcal{B}_0\cap S_q} \langle \frac{\partial l(x,y)}{\partial w_s^{(0)}}, q\rangle + \frac{1}{B}\sum_{x\in\mathcal{B}_0\cap S_q'} \langle \frac{\partial l(x,y)}{\partial w_s^{(0)}}, q\rangle$

Now, from equation (14), for any $(x,y) \in \mathcal{D}\backslash S_q$, $\langle \frac{\partial l(x,y)}{\partial w_s^{(0)}}, q\rangle = 0$

Therefore, $\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle = \frac{1}{B}\sum_{x\in\mathcal{B}_0\cap S_q} \langle \frac{\partial l(x,y)}{\partial w_s^{(0)}}, q\rangle$

Now, as $s \in S_1$, $a^{(s)} = +1$

Furthermore, for any (x,y), $l_q^{(s,0)}(x)G_q^{(s,0)}(x) \leq 1$

Therefore, as $|\delta_{1,q}^{(s,0)}| = O(mC_2)$, $|\delta_{2,q}^{(s,0)}| = O(mC_2)$ and $|\delta_{q,q'}^{(s,0)}| = O(mC_2)$, from equation (14), $\left|\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle\right| \leq \frac{|\mathcal{B}_0 \cap S_q|}{B} O(mC_2)$

Now, from Lemma E.1, *w.h.p.* $\frac{|\mathcal{B}_0 \cap S_q|}{B} \leq p_q^{(s,0)} + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$ which implies,

$\left|\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle\right| \leq O\left(\frac{mC_2}{d}\right) + \tilde{O}\left(\frac{mC_2}{\sqrt{B}}\right)$

(ii) Using equation (15) and the fact that $p_1^{(s,0)} = \Omega(1)$, by following the same procedure as in the proof of statement (i), we can complete the proof.

(iii) Using equation (18) and the fact that $p_1^{(s,0)} = \Omega(1)$, by following the same procedure as in the proof of the statement (i) we can complete the proof.

(iv) As $p_2^{(s,0)} \geq 0$, using equation (19) and the same procedure as in the proof of the statement (i) we can show that $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2 \rangle \geq 0$. Similarly, as $G_2^{(s,0)}(x) \leq 1$ we can show that $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2 \rangle \leq \frac{1}{2} + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$.

(v) Using equation (17) and the fact that for any $(x, y) \sim \mathcal{D}$, $l_q^{(s,t)}(x) G_q^{(s,t)}(x) \leq 1$ and by following the same procedure as in the proof of the statement (i) we can complete the proof. □

Lemma E.3. *For any expert $s \in S_2$ such that $p_2^{(s,0)} = \Omega(1)$, w.h.p. over a randomly sampled batch of size B we can ensure that,*

(i) $\left| \langle \frac{\partial l}{\partial w_s^{(0)}}, q \rangle \right| \leq O\left(\frac{mC_2}{d}\right) + \tilde{O}\left(\frac{mC_2}{\sqrt{B}}\right)$

(ii) $\left| \langle \frac{\partial l}{\partial w_s^{(0)}}, o_2 \rangle \right| \leq O\left(mC_2\right) + \tilde{O}\left(\frac{mC_2}{\sqrt{B}}\right)$

(iii) $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1 \rangle \leq \frac{1}{2} + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$ *and* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1 \rangle \geq 0$

(iv) $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2 \rangle \geq -\frac{1}{2} - \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$ *and* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2 \rangle \leq -\frac{1}{2l} + \tilde{O}\left(\frac{1}{l\sqrt{B}}\right)$

(v) $\left| \langle \frac{\partial l}{\partial w_r^{(s,0)}}, q \rangle \right| \leq O\left(\frac{1}{d}\right) + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$

Proof. Using the Lemma E.1 and following the same procedure as in Lemma E.2, we can complete the proof. □

Lemma E.4. *For any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, by selecting $\eta_r = O\left(\frac{\eta_e C_p}{ml^2 C_2^2}\right)$ and $B = \tilde{\Omega}\left(l^2 d^2\right)$, we can ensure that after $T'' = O\left(\frac{lC_2}{\eta_e}\right)$ iterations,*

(i) for any task-irrelevant pattern q, $\delta_{1,q}^{(s,T'')} = \Omega\left(mC_2\right)$, $|\delta_{2,q}^{(s,T'')}| = O(mC_2)$ and, for any different two task-irrelevant pattern q and q' $|\delta_{q,q'}^{(s,T'')}| = O(mC_2)$

(ii) for any task-irrelevant pattern q such that $\langle w_s^{(0)}, q \rangle < \langle w_s^{(0)}, o_1 \rangle$, $\langle w_s^{(T'')}, o_1 \rangle - \langle w_s^{(T'')}, q \rangle = \Omega(C_p)$ and $\langle w_s^{(T'')}, o_1 \rangle - \langle w_s^{(T'')}, q \rangle < \log l$

(iii) for any task-irrelevant pattern q such that $\langle w_s^{(0)}, q \rangle > \langle w_s^{(0)}, o_1 \rangle$, $\langle w_s^{(T'')}, q \rangle - \langle w_s^{(T'')}, o_1 \rangle = \Omega(C_p)$ and $\langle w_s^{(T'')}, q \rangle - \langle w_s^{(T'')}, o_1 \rangle < \log l$

(iii) $p_1^{(s,T'')} = p_1^{(s,0)}$

Proof. From the statement (i) of the Lemma E.2, *w.h.p.* over a randomly sampled batch, $\left| \langle \frac{\partial l}{\partial w_s^{(0)}}, q \rangle \right| \leq O(\frac{mC_2}{d}) + \tilde{O}(\frac{mC_2}{\sqrt{B}})$

Therefore, $\langle w_s^{(1)}, q\rangle = \langle w_s^{(0)}, q\rangle - \eta_r \langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle \leq \langle w_s^{(0)}, q\rangle + O(\frac{mC_2}{d}\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)$

On the other hand, from the statement (ii) of the Lemma E.2, *w.h.p.* over a randomly sampled batch, $\left|\langle \frac{\partial l}{\partial w_s^{(0)}}, o_1\rangle\right| \leq O(mC_2) + \tilde{O}(\frac{mC_2}{\sqrt{B}})$

Therefore, $\langle w_s^{(1)}, o_1\rangle = \langle w_s^{(0)}, o_1\rangle - \eta_r \langle \frac{\partial l}{\partial w_s^{(0)}}, o_1\rangle \geq \langle w_s^{(0)}, o_1\rangle - O(mC_2\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)$

Therefore, if $\langle w_s^{(0)}, q\rangle < \langle w_s^{(0)}, o_1\rangle$,

$$\begin{aligned}\langle w_s^{(1)}, o_1\rangle - \langle w_s^{(1)}, q\rangle &\geq \langle w_s^{(0)}, o_1\rangle - \langle w_s^{(0)}, q\rangle - O(mC_2\eta_r) - O(\frac{mC_2}{d}\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)\\ &\geq C_p - O(mC_2\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)\end{aligned}$$

and if $\langle w_s^{(0)}, q\rangle > \langle w_s^{(0)}, o_1\rangle$

$$\begin{aligned}\langle w_s^{(1)}, q\rangle - \langle w_s^{(1)}, o_1\rangle &\leq \langle w_s^{(0)}, q\rangle - \langle w_s^{(0)}, o_1\rangle + O(mC_2\eta_r) + O(\frac{mC_2}{d}\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)\\ &\leq \frac{1}{\sqrt{2}}\log l + O(mC_2\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)\end{aligned}$$

Now, by selecting $\eta_r = O(\frac{C_p}{mC_2})$ and $B = \tilde{\Omega}(d^2)$, for $\langle w_s^{(0)}, q\rangle < \langle w_s^{(0)}, o_1\rangle$ we get , $\langle w_s^{(1)}, o_1\rangle - \langle w_s^{(1)}, q\rangle = \Omega(C_p)$ which ensures that $p_1^{(s,1)} \geq p_1^{(s,0)}$. Similarly for $\langle w_s^{(0)}, q\rangle > \langle w_s^{(0)}, o_1\rangle$ we get, $\langle w_s^{(1)}, q\rangle - \langle w_s^{(1)}, o_1\rangle < \log l$ as, $C_p = \Theta\left(\frac{1}{d}\right)$.

Now, for any $r \in [m]$ such that $\langle w_r^{(s,0)}, o_1\rangle \geq 0$, from the statement (iii) of the Lemma E.2, *w.h.p.* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1\rangle \leq -(\frac{1}{2l} - \tilde{O}(\frac{1}{l\sqrt{B}}))$ which implies $\langle w_r^{(s,1)}, o_1\rangle = \langle w_r^{(s,0)}, o_1\rangle - \eta_e \langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1\rangle \geq \langle w_r^{(s,0)}, o_1\rangle + \frac{\eta_e}{2l} - \tilde{O}(\frac{\eta_e}{l\sqrt{B}})$ and hence $\sigma_1^{(s,1)} \geq \sigma_1^{(s,0)} + \Omega(\frac{m\eta_e}{l}) - \tilde{O}(\frac{m\eta_e}{l\sqrt{B}})$.

Again, for any $r \in [m]$ such that $\langle w_r^{(s,0)}, o_2\rangle \geq 0$, from the statement (iv) of the Lemma E.2, *w.h.p.* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2\rangle \geq 0$ which implies
$\langle w_r^{(s,1)}, o_2\rangle = \langle w_r^{(s,0)}, o_2\rangle - \eta_e \langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2\rangle \leq \langle w_r^{(s,0)}, o_2\rangle$ and hence $\sigma_2^{(s,1)} \leq \sigma_2^{(s,0)}$.

On the other hand, for any $r \in [m]$ and any q such that $\langle w_r^{(s,0)}, q\rangle \geq 0$, from the statement (v) of the Lemma E.2, *w.h.p.* $\left|\langle \frac{\partial l}{\partial w_r^{(s,0)}}, q\rangle\right| \leq O\left(\frac{1}{d}\right) + \tilde{O}(\frac{1}{\sqrt{B}})$.

Therefore, for any q, $\langle w_r^{(s,1)}, q\rangle = \langle w_r^{(s,0)}, q\rangle - \eta_e \langle \frac{\partial l}{\partial w_r^{(s,0)}}, q\rangle \geq \langle w_r^{(s,0)}, q\rangle - O(\frac{\eta_e}{d}) - \tilde{O}(\frac{\eta_e}{\sqrt{B}})$ and $\langle w_r^{(s,1)}, q\rangle \leq \langle w_r^{(s,0)}, q\rangle + O(\eta_e\frac{1}{d}) + \tilde{O}(\frac{\eta_e}{\sqrt{B}})$ which implies $\sigma_q^{(s,1)} \geq 0$ and

$\sigma_q^{(s,1)} \leq \sigma_q^{(s,0)} + O(\frac{m}{d}\eta_e) + \tilde{O}(\frac{m\eta_e}{\sqrt{B}})$.

Therefore, for any task-irrelevant pattern q, $\delta_{1,q}^{(s,1)} \geq \delta_{1,q}^{(s,0)} + \Omega(\frac{m\eta_e}{l}) - \tilde{O}(\frac{m\eta_e}{\sqrt{B}})$ and for any two different task-irrelevant pattern, $|\delta_{q,q'}^{(s,1)} - \delta_{q,q'}^{(s,0)}| \leq O(\frac{m}{d}\eta_e) + \tilde{O}(\frac{m\eta_e}{\sqrt{B}})$.

Now, selecting $B = \tilde{\Omega}(l^2 d^2)$ we get, $\delta_{1,q}^{(s,1)} - \delta_{1,q}^{(s,0)} = \Omega(\frac{m\eta_e}{l})$ and $|\delta_{q,q'}^{(s,1)} - \delta_{q,q'}^{(s,0)}| \leq O(\frac{m}{d}\eta_e)$.

In that case, $\langle \frac{\partial l}{\partial w_s^{(1)}}, q\rangle \geq -O(\frac{mC_2}{d}) - O(\frac{m}{d^2}\eta_e) - \tilde{O}(\frac{mC_2}{\sqrt{B}})$ and $\langle \frac{\partial l}{\partial w_s^{(1)}}, o_1\rangle \leq O(mC_2) + \tilde{O}(\frac{mC_2}{\sqrt{B}})$ as $p_1^{(s,1)} \geq p_1^{(s,0)}$.

Therefore, for any T'' such that $\forall 0 \leq t \leq T''$, $p_1^{(s,t)} \geq p_1^{(s,0)}$, by induction we can show that,
for any q, $\langle \frac{\partial l}{\partial w_s^{(t)}}, q\rangle \geq -O(\frac{mC_2}{d}) - \tilde{O}(\frac{mC_2}{\sqrt{B}})$ and $\langle \frac{\partial l}{\partial w_s^{(t)}}, o_1\rangle \leq O(mC_2) + \tilde{O}(\frac{mC_2}{\sqrt{B}})$.

Therefore, for $\langle w_s^{(0)}, q\rangle < \langle w_s^{(0)}, o_1\rangle$, $\langle w_s^{(T'')}, o_1\rangle - \langle w_s^{(T'')}, q\rangle \geq C_p - O(mC_2\eta_r T'') - O(\frac{m}{d^2}\eta_e\eta_r T'') - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r T'')$ and for $\langle w_s^{(0)}, q\rangle > \langle w_s^{(0)}, o_1\rangle$, $\langle w_s^{(T'')}, q\rangle - \langle w_s^{(T'')}, o_1\rangle \leq \frac{1}{2}\log l + O(mC_2\eta_r T'') + O(\frac{m}{d^2}\eta_e\eta_r T'') + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r T'')$.

On the other hand, assuming for all $0 \leq t \leq T''$, $p_1^{(s,t)} \geq p_1^{(s,0)}$, for any q we get, $\delta_{1,q}^{(s,T'')} \geq \delta_{1,q}^{(s,0)} + \Omega(\frac{m\eta_e}{l}T'') - \tilde{O}(\frac{m\eta_e}{\sqrt{B}}T'')$, which implies we need $T'' = O(\frac{lC_2}{\eta_e})$ iterations to achieve $\delta_{(1,q)}^{(s,T'')} = \Omega(mC_2)$. In that case, for any different two task-irrelevant pattern q and q', $|\delta_{q,q'}^{(s,T'')}| = O(mC_2)$ and $|\delta_{2,q}^{(s,T'')}| = O(mC_2)$.

Now, to ensure that for all $0 \leq t \leq T''$, $p_1^{(s,t)} \geq p_1^{(s,0)}$, we need $\eta_r = O(\frac{C_p}{mC_2}\frac{1}{T''}) = O(\frac{C_p\eta_e}{mlC_2^2})$ such that, for any q s.t. $\langle w_s^{(0)}, q\rangle < \langle w_s^{(0)}, o_1\rangle$, $\langle w_s^{(t)}, o_1\rangle - \langle w_s^{(t)}, q\rangle \geq \Omega(C_p)$ for all $0 \leq t \leq T''$.

In that case, for any q s.t. $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$, $\langle w_s^{(T'')}, q\rangle - \langle w_s^{(T'')}, o_1\rangle < \log l$.

Now, $\langle w_s^{(1)}, o_1\rangle \leq \langle w_s^{(0)}, o_1\rangle + O(mC_2\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)$ and, $\langle w_s^{(1)}, q\rangle \geq \langle w_s^{(0)}, q\rangle - O(\frac{mC_2}{d}\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)$.

Therefore, if $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$,

$$\begin{aligned}\langle w_s^{(1)}, q\rangle - \langle w_s^{(1)}, o_1\rangle &\geq \langle w_s^{(0)}, q\rangle - \langle w_s^{(0)}, o_1\rangle - O(mC_2\eta_r) - O(\frac{mC_2}{d}\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r) \\ &\geq C_p - O(mC_2\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r)\end{aligned}$$

Therefore, for any q s.t. $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$, by the selection of η_r, $\langle w_s^{(1)}, q\rangle - \langle w_s^{(1)}, o_1\rangle = \Omega(C_p)$ which implies $p_1^{(s,1)} = p_1^{(s,0)}$.

Now, for any $r \in [m]$ such that $\langle w_r^{(s,0)}, o_1\rangle \geq 0$, from the statement (iii) of the Lemma E.2, *w.h.p.* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_1\rangle \geq -O(1) - \tilde{O}(\frac{1}{\sqrt{B}})$ which implies $\langle w_r^{(s,1)}, o_1\rangle \leq \langle w_r^{(s,0)}, o_1\rangle + O(\eta_e) + \tilde{O}(\frac{\eta_e}{\sqrt{B}})$ and hence

$\sigma_1^{(s,1)} \leq \sigma_1^{(s,0)} + O(m\eta_e) + \tilde{O}(\frac{m\eta_e}{\sqrt{B}})$.

Therefore, for any q, $\delta_{1,q}^{(s,1)} - \delta_{1,q}^{(s,0)} \leq O(m\eta_e)$ and as from the statement (v) of the Lemma E.2, *w.h.p.* $\langle \frac{\partial l}{\partial w_r^{(s,0)}}, o_2 \rangle \leq \frac{1}{2} + \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$, $\delta_{q,2}^{(s,1)} - \delta_{q,2}^{(s,0)} \leq O(m\eta_e)$.

Now, as $p_1^{(s,1)} = p_1^{(s,0)}$,

$$\langle w_s^{(2)}, o_1 \rangle \leq \langle w_s^{(1)}, o_1 \rangle + O(mC_2\eta_r) + O(m\eta_e\eta_r) + \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r) + \tilde{O}(\frac{m\eta_e}{\sqrt{B}}\eta_r)$$

and,

$$\langle w_s^{(2)}, q \rangle \geq \langle w_s^{(1)}, q \rangle - O(\frac{mC_2}{d}\eta_r) - O(\frac{m}{d}\eta_e\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r) - \tilde{O}(\frac{m\eta_e}{d\sqrt{B}}\eta_r)$$

Therefore, for any q s.t. $\langle w_s^{(0)}, o_1 \rangle < \langle w_s^{(0)}, q \rangle$,

$$\begin{aligned}
&\langle w_s^{(2)}, q \rangle - \langle w_s^{(2)}, o_1 \rangle \\
&\geq \langle w_s^{(1)}, q \rangle - \langle w_s^{(1)}, o_1 \rangle - O(mC_2\eta_r) - O(m\eta_e\eta_r) - \tilde{O}(\frac{mC_2}{\sqrt{B}}\eta_r) - \tilde{O}(\frac{m\eta_e}{\sqrt{B}}\eta_r) \\
&= C_p - O(C_p\frac{2}{T}) - O(C_p\frac{l}{T''^2}) = \Omega(C_p)
\end{aligned}$$

which implies $p_1^{(s,2)} = p_1^{(s,0)}$.

Therefore, by induction, for any $t \leq T''$, and for any q s.t. $\langle w_s^{(0)}, o_1 \rangle < \langle w_s^{(0)}, q \rangle$, by selecting $\eta_r = O(\frac{\eta_e C_p}{ml^2C_2^2})$,

$$\langle w_s^{(t)}, q \rangle - \langle w_s^{(t)}, o_1 \rangle \geq C_p - O(C_p\frac{t}{T''}) - O(C_p\frac{t^2}{T''^2})$$

which implies, for any q s.t. $\langle w_s^{(0)}, o_1 \rangle < \langle w_s^{(0)}, q \rangle$, $\langle w_s^{(T'')}, o_1 \rangle < \langle w_s^{(T'')}, q \rangle = \Omega(C_p)$ and hence $p_1^{(s,T'')} = p_1^{(s,0)}$.

Similarly, for any $t \leq T''$, and for any q s.t. $\langle w_s^{(0)}, q \rangle < \langle w_s^{(0)}, o_1 \rangle$,

$$\begin{aligned}
\langle w_s^{(t)}, o_1 \rangle - \langle w_s^{(t)}, q \rangle &\leq \langle w_s^{(1)}, o_1 \rangle - \langle w_s^{(1)}, q \rangle + O(C_p\frac{t}{T''}) + O(C_p\frac{l(t-1)}{T''^2}) \\
&\leq \frac{1}{2}\log l + O(C_p\frac{t}{T''}) + O(C_p\frac{t^2}{T''^2})
\end{aligned}$$

which implies, for any q s.t. $\langle w_s^{(0)}, q \rangle < \langle w_s^{(0)}, o_1 \rangle$, $\langle w_s^{(T'')}, o_1 \rangle - \langle w_s^{(T'')}, q \rangle < \log l$.

□

Lemma E.5. *For any expert $s \in S_2$ such that $p_2^{(s,0)} = \Omega(1)$, for η_r and B as in Lemma E.4, we can ensure that after T'' iterations where T'' is as in Lemma E.4,*

(i) for any task-irrelevant pattern q, $\delta_{2,q}^{(s,T'')} = \Omega(mC_2)$, $|\delta_{1,q}^{(s,T'')}| = O(mC_2)$ and, for any different two task-irrelevant pattern q and q' $|\delta_{q,q'}^{(s,T'')}| = O(mC_2)$

(ii) for any task-irrelevant pattern q such that $\langle w_s^{(0)}, q \rangle < \langle w_s^{(0)}, o_2 \rangle$, $\langle w_s^{(T'')}, o_2 \rangle - \langle w_s^{(T'')}, q \rangle = \Omega(C_p)$, and $\langle w_s^{(T'')}, o_2 \rangle - \langle w_s^{(T'')}, q \rangle < \log l$

(iii) for any task-irrelevant pattern q such that $\langle w_s^{(0)}, q\rangle > \langle w_s^{(0)}, o_2\rangle$, $\langle w_s^{(T'')}, q\rangle - \langle w_s^{(T'')}, o_2\rangle = \Omega(C_p)$, and $\langle w_s^{(T'')}, q\rangle - \langle w_s^{(T'')}, o_2\rangle < \log l$
(iv) $p_2^{(s,T'')} = p_2^{(s,0)}$

Proof. Using Lemma E.3 and following the same procedure as in Lemma E.4 we can complete the proof. □

Lemma E.6. *For any t and q with $\langle w_s^{(t)}, q\rangle < \langle w_s^{(t)}, o_1\rangle$ such that $\langle w_s^{(t)}, o_1\rangle - \langle w_s^{(t)}, q\rangle \leq 2\log l$, $G_1^{(s,t)}(x)(1-G_1^{(s,t)}(x)) \geq \frac{1}{4l}$, where $(x, y=+1) \sim \mathcal{D}$ s.t. $\exists j \in J_s^{(t)}(x)$ with $x^{(j)} = q$. Similarly, for any t and q with $\langle w_s^{(t)}, q\rangle < \langle w_s^{(t)}, o_2\rangle$ such that $\langle w_s^{(t)}, o_2\rangle - \langle w_s^{(t)}, q\rangle \leq 2\log l$, $G_2^{(s,t)}(x)(1-G_2^{(s,t)}(x)) \geq \frac{1}{4l}$, where $(x, y=-1) \sim \mathcal{D}$ s.t. $\exists j \in J_s^{(t)}(x)$ with $x^{(j)} = q$*

Proof. For any t and q with $\langle w_s^{(t)}, q\rangle < \langle w_s^{(t)}, o_1\rangle$, as $\langle w_s^{(t)}, o_1\rangle - \langle w_s^{(t)}, q\rangle \leq \log l$, for any $(x, y) \sim \mathcal{D}$ s.t. $\exists j \in J_s^{(t)}(x)$ with $x^{(j)} = q$, $G_1^{(s,t)}(x)(1-G_1^{(s,t)}(x)) \geq \min\{\frac{\frac{(l-1)}{l^2}}{(1+\frac{(l-1)}{l^2})^2}, \frac{(l-1)}{l^2}\} = \frac{\frac{(l-1)}{l^2}}{(1+\frac{(l-1)}{l^2})^2}$.

Now, $\frac{\frac{(l-1)}{l^2}}{(1+\frac{(l-1)}{l^2})^2} = \frac{l^2(l-1)}{(l^2+l-1)^2}$.

Now, let there exists a constant $C > 0$ such that $\frac{l^2(l-1)}{(l^2+l-1)^2} \geq \frac{C}{l} \Leftrightarrow l^4(1-C) - l^3(1+2C) + Cl^2 + 2Cl - C \geq 0$.

Now, $Cl^2+2Cl-C > 0$ as $l \geq 2$. Therefore, $l^3(1+2C) \leq l^4(1-C)$ satisfies $l^4(1-C)-l^3(1+2C)+Cl^2+2Cl-C \geq 0$.

Now, $l^3(1+2C) \leq l^4(1-C) \Leftrightarrow C \leq \frac{l-1}{l+2}$. Now, $\frac{l-1}{l+2} \geq \frac{1}{4}$ as $l \geq 2$. Hence, picking $C = \frac{1}{4}$ satisfies that $\frac{l^2(l-1)}{(l^2+l-1)^2} \geq \frac{1}{4l}$ which implies $G_1^{(s,t)}(x)(1-G_1^{(s,t)}(x)) \geq \frac{1}{4l}$.

Similarly, we can show that for any t and q with $\langle w_s^{(t)}, q\rangle < \langle w_s^{(t)}, o_2\rangle$ such that $\langle w_s^{(t)}, o_2\rangle - \langle w_s^{(t)}, q\rangle \leq 2\log l$, $G_2^{(s,t)}(x)(1-G_2^{(s,t)}(x)) \geq \frac{1}{4l}$, where $(x, y=-1) \sim \mathcal{D}$ s.t. $\exists j \in J_s^{(t)}(x)$ with $x^{(j)} = q$. □

F. Proof of Lemma 4.2

Lemma F.1 (Full version of the **Lemma 4.2**). *Suppose the expert learning rate η_e such that, the router learning rate $\eta_r = O\left(\frac{\eta_e C_p}{mdl^2C_2^2}\right)$, the batch-size $B = \tilde{\Omega}(l^2d^2)$, and the number of iterations $T = \Omega\left(\frac{l^2C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}}\right)$. Then,*

For any expert $s \in S_1$ such that $p_1^{(s,0)} = O\left(\frac{1}{d}\right)$, we have

(i) $p_1^{(s,T)} = O(1/d)$,
(ii) $\Delta_s^{(T)} = O(\log^2 l/\sqrt{d}) + O(l^4\log^2 l/d^2)$.

and, for any expert $s \in S_2$ such that $p_2^{(s,0)} = O\left(\frac{1}{d}\right)$, we have

(iii) $p_2^{(s,T)} = O(1/d)$,

(iv) $\Delta_s^{(T)} = O(\log^2 l/\sqrt{d}) + O(l^4 \log^2 l/d^2)$.

Proof. For any expert $s \in S_1$ such that $p_1^{(s,0)} = O\left(\frac{1}{d}\right)$, for any q, $\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle \leq O(\frac{mC_2}{d}) + \tilde{O}(\frac{mC_2}{\sqrt{B}})$.

Therefore, with $B = \tilde{\Omega}(l^2 d^2)$, $\langle \frac{\partial l}{\partial w_s^{(0)}}, q\rangle \leq O(\frac{mC_2}{d})$.

Hence, $\langle w_s^{(1)}, q\rangle \geq \langle w_s^{(0)}, q\rangle - O(\frac{mC_2}{d}\eta_r)$.

Similarly, $\langle w_s^{(1)}, o_1\rangle \leq \langle w_s^{(0)}, o_1\rangle + O(\frac{mC_2}{d}\eta_r)$.

Therefore, for any q s.t. $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$, $\langle w_s^{(1)}, q - o_1\rangle \geq \langle w_s^{(0)}, q - o_1\rangle - O(\frac{mC_2}{d}\eta_r)$.

Now, with $\eta_r = O(\frac{\eta_e C_p}{ml^2 C_2^2})$ and $\langle w_s^{(0)}, q - o_1\rangle \geq C_p$, we get $\langle w_s^{(1)}, q - o_1\rangle = \Omega(C_p)$.

Therefore, $p_1^{(s,1)} \leq p_1^{(s,0)}$.

Now, for $B = \tilde{\Omega}(l^2 d^2)$, w.h.p. for any q, as $p_1^{(s,0)} = O(\frac{1}{d})$,
$\delta_{1,q}^{(s,1)} - \delta_{1,q}^{(s,0)} \leq O(\frac{m\eta_e}{d})$, $\delta_{2,q}^{(s,0)} - \delta_{2,q}^{(s,1)} \leq O(m\eta_e)$ and for any two different task-irrelevant patterns q and q', $|\delta_{q,q'}^{(s,0)} - \delta_{q,q'}^{(s,1)}| = O(\frac{m\eta_e}{d})$.

Therefore, for any q s.t. $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$,
$\langle w_s^{(2)}, q - o_1\rangle \geq \langle w_s^{(1)}, q - o_1\rangle - O(\frac{mC_2}{d}\eta_r) - O(\frac{m\eta_e}{d}\eta_r) = \Omega(C_p)$ which implies $p_1^{(s,2)} \leq p_1^{(s,0)}$.

Therefore,
by induction, for any t' s.t. $\forall 0 \leq t \leq t'$, $p_1^{(s,t)} \leq p_1^{(s,0)} = O(\frac{1}{d})$, for any q s.t. $\langle w_s^{(0)}, o_1\rangle < \langle w_s^{(0)}, q\rangle$,

$$\langle w_s^{(t'+1)}, q - o_1\rangle \geq \langle w_s^{(0)}, q - o_1\rangle - O(\frac{mC_2}{d}\eta_r)t' - O(\frac{m\eta_e}{d^2}\eta_r)t'^2 \tag{24}$$

Now, given, $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$.

Therefore, using inequality (24), $\langle w_s^{(T+1)}, q - o_1\rangle = \Omega(C_p)$ which implies that,
for any $t \leq T+1$, $p_1^{(s,t)} \leq p_1^{(s,0)} = O(\frac{1}{d})$.

Now, as $p_1^{(s,T)} \leq p_1^{(s,0)} = O(\frac{1}{d})$, $\forall q$,

$$\langle w_s^{(T+1)}, q\rangle \leq \langle w_s^{(0)}, q\rangle + O(\frac{mC_2}{d}\eta_r)T + O(\frac{m\eta_e}{d^2}\eta_r)T^2 \tag{25}$$

as $\forall t \leq T$, $\delta_{q,2}^{(s,t-1)} - \delta_{q,2}^{(s,t)} \leq 0$, $\delta_{q,1}^{(s,t)} - \delta_{q,1}^{(s,t-1)} \leq O(\frac{m\eta_e}{d})$, $|\delta_{q,q'}^{(s,t-1)} - \delta_{q,q'}^{(s,t)}| = O(\frac{m\eta_e}{d})$, $\left|\delta_{2,q}^{(s,0)}\right| = O(mC_2)$ and,

$\left|\delta_{q,q'}^{(s,0)}\right| = O(mC_2)$.

Therefore, as given $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$, using inequality (25),
$\langle w_s^{(T+1)}, q\rangle - \langle w_s^{(s,0)}, q\rangle \leq O(\frac{1}{d^{3/2}}\log l)$.

Therefore, $\langle w_s^{(T+1)}, q\rangle^2 \leq \langle w_s^{(0)}, q\rangle^2 + O(\frac{1}{d^{3/2}}\log^2 l)$.

Similarly, we can show that, $\langle w_s^{(T+1)}, o_1\rangle^2 \leq \langle w_s^{(0)}, o_1\rangle^2 + O(\frac{1}{d^{3/2}}\log^2 l)$.

Again as $\delta_{2,q}^{(t-1)} - \delta_{2,q}^{(t)} \leq O(\frac{m\eta_e}{d})$,

$$\langle w_s^{(T+1)}, o_2\rangle \leq \langle w_s^{(0)}, o_2\rangle + O(mC_2\eta_r)T + O(\frac{m\eta_e}{d}\eta_r)T^2 \tag{26}$$

Therefore, with $T = O(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}})$, using inequality (26),
$\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(s,0)}, o_2\rangle \leq O(\frac{1}{d^{1/2}}\log l) + O(\frac{l^2}{d}\log l)$.

Therefore, $\langle w_s^{(T+1)}, o_2\rangle^2 \leq \langle w_s^{(0)}, o_2\rangle^2 + O(\frac{l^4}{d^2}\log^2 l)$.

Therefore,

$$\left|\left|w_s^{(T+1)}\right|\right|^2 - \left|\left|w_s^{(0)}\right|\right|^2 \leq O(\frac{1}{d^{1/2}}\log^2 l) + O(\frac{l^4}{d^2}\log^2 l)$$

Therefore, $\Delta_s^{(T)} \leq O(\frac{1}{d^{1/2}}\log^2 l) + O(\frac{l^4}{d^2}\log^2 l)$.

Similarly, for any expert $s \in S_2$ such that $p_1^{(s,0)} = O\left(\frac{1}{d}\right)$, we can complete the proof for the statements (iii) and (iv). □

G. Proof of Theorem 4.3

Proof. From Lemma D.1 (i.e., Lemma 4.1) and F.1 (i.e., Lemma 4.2), after pruning the model with expert-pruning-ratio $\rho \leq \gamma$, there exist $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$ and exist $s \in S_2$ such that $p_2^{(s,0)} = \Omega(1)$.

Again, from Lemma D.1, for any expert $s \in S_1$ such that $p_1^{(s,0)} = \Omega(1)$, $\langle w_r^{(s,T+1)}, o_1\rangle = \Omega(lC_2\sqrt{d}\sqrt{\log l})$

Similarly, for any $s \in S_2$ s.t. $p_2^{(s,0)} = \Omega(1)$, $\langle w_r^{(s,T+1)}, o_2\rangle = \Omega(lC_2\sqrt{d}\sqrt{\log l})$.

Now, $\forall s \in [k]$, $\forall q$ and $\forall r \in [m]$ s.t. $\langle w_r^{(s,0)}, q\rangle \geq 0$, $\forall t$, $\langle \frac{\partial l}{\partial w_r^{(s,t)}}, q\rangle \geq -O(\frac{1}{d}) - \tilde{O}(\frac{1}{\sqrt{B}})$.

Therefore,

$$\langle w_r^{(s,T+1)}, q\rangle \leq \langle w_r^{(s,0)}, q\rangle + O(\frac{1}{d}\eta_e T) = C_2 + O(\frac{l^2}{\sqrt{d}}\sqrt{\log l})C_2$$

Again, $\forall s \in S_2$, $\forall t$ and, $\forall r \in [m]$, $\langle \frac{\partial l}{\partial w_r^{(s,t)}}, o_1 \rangle \geq 0$ which implies $\langle w_r^{(s,T+1)}, o_1 \rangle \leq \langle w_r^{(s,0)}, o_1 \rangle \leq C_2$.

Similarly, $\forall s \in S_1$, $\forall t$ and, $\forall r \in [m]$, $\langle w_r^{(s,T+1)}, o_2 \rangle \leq C_2$.

Now, after T iterations of SGD and pruning with $\rho \leq \gamma$, $\forall (x, y = +1) \sim \mathcal{D}$,

$$
\begin{aligned}
f^{(T+1,\rho)}(x) &= \sum_{s \in S_{k'}} f_s^{(T+1)}(x) \\
&= \sum_{s \in S_{k'}} a^{(s)} \sum_{j \in J_s^{(T+1)}(x)} G_j^{(s,T+1)} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, x^{(j)} \rangle) \\
&= \sum_{s \in S_1 \cap S_{k'}} \sum_{j \in J_s^{(T+1)}(x)} G_j^{(s,T+1)} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, x^{(j)} \rangle) \\
&\quad - \sum_{s \in S_2 \cap S_k} \sum_{j \in J_s^{(T+1)}(x)} G_j^{(s,T'+1)} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, x^{(j)} \rangle) \\
&\geq \sum_{s \in S_1 \cap \{s : p_1^{(s,0)} = \Omega(1)\} \cap S_{k'}} \sum_{j \in J_s^{(T+1)}(x)} G_j^{(s,T+1)} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, x^{(j)} \rangle) \\
&\quad - \sum_{s \in S_2 \cap S_{k'}} \sum_{j \in J_s^{(T+1)}(x)} G_j^{(s,T+1)} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, x^{(j)} \rangle) \\
&\overset{(I)}{\geq} \sum_{s \in S_1 \cap \{s : p_1^{(s,0)} = \Omega(1)\} \cap S_{k'}} G_1^{(s,T+1)}(x) \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, o_1 \rangle) \\
&\quad - \sum_{s \in S_2 \cap S_{k'}} \sum_{r=1}^{m} \text{ReLU}(\langle w_r^{(s,T+1)}, q^{(s)} \rangle) \\
&\overset{(II)}{\geq} \Omega(mlC_2\sqrt{d}\sqrt{\log l}) - O(kmC_2) - O(km\frac{1}{\sqrt{d}} l^2 \sqrt{\log l} C_2)
\end{aligned}
$$

Here, (I) comes from the statement (i) of Lemma D.1, $q^{(s)}$ in (I) denotes the task-irrelevant pattern routed to expert $s \in S_2 \cap S_{k'}$ and, (II) comes from the statement (ii) of Lemma D.1 and the fact that $\exists s \in S_1 \cap S_{k'}$ s.t. $p_1^{(s,0)} = \Omega(1)$.

Therefore, for $k = O(\sqrt{d})$, $\forall (x, y = +1) \sim \mathcal{D}$, $f^{(T+1,\rho)}(x) > 1$.

Similarly, using the statements (vi),(vii) and (viii) of Lemma D.1, and as $\exists s \in S_2 \cap S_{k'}$ s.t. $p_2^{(s,0)} = \Omega(1)$, we can show that, $\forall (x, y = -1) \sim \mathcal{D}$, $f^{(T+1,\rho)}(x) < -1$.

Therefore, $\forall (x, y) \sim \mathcal{D}$, $y f^{(T+1,\rho)}(x) > 1$ which implies that,
$\hat{l}(f^{(T+1,\rho)}(x), y) = 0$ and, $\mathbb{P}\left[\forall (x, y) \sim \mathcal{D} : y f^{(T+1,\rho)}(x) > 0\right] = 1$.

□

H. Proof of Lemma 4.4

Lemma H.1 (Full version of the **Lemma 4.4**). *Suppose the expert learning rate η_e such that, the router learning rate $\eta_r = O\left(\frac{\eta_e C_p}{mdl^2 C_2^2}\right)$, the batch-size $B = \tilde{\Omega}(l^2 d^2)$, and the number of iterations $T = \Omega\left(\frac{l^2 C_2}{\eta_e}\sqrt{\frac{\log l}{C_p}}\right)$. For any expert*

$s \in S_1$ such that $\Delta_s^{(T)} > \frac{3}{2}\log l$, we have

(i) $p_1^{(s,T)} = 1$,
(ii) for every $(x, +1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_1$,

Moreover, after pruning experts with the ratio $\rho \geq \gamma$, if $s \in S_{k'}$, and the number of post-pruning fine-tuning steps satisfies $T' = \Omega(kC_2 l^2 \sqrt{\log l}/\eta_e)$, then we have,

(iii) $\langle w_r^{(s,T,T')}, o_1\rangle = \Omega(kl^2 C_2 \sqrt{\log l})$ for a constant fraction $r \in [m]$ of s

Similarly, for $s \in S_2$ such that $\Delta_s^{(T)} > \frac{3}{2}\log l$, we have

(iv) $p_2^{(s,T)} = 1$,
(v) for every $(x, -1) \sim \mathcal{D}$, $G_j^{(s,T)}(x) > 1/2$, if $x^{(j)} = o_2$,

Moreover, after pruning experts with the ratio $\rho \geq \gamma$, if $s \in S_{k'}$, and the number of post-pruning fine-tuning steps satisfies $T' = \Omega(kC_2 l^2 \sqrt{\log l}/\eta_e)$, then we have,

(vi) $\langle w_r^{(s,T,T')}, o_2\rangle = \Omega(kl^2 C_2 \sqrt{\log l})$ for a constant fraction $r \in [m]$ of $s \in [k]$

Proof. For any $s \in S_1$ such that $\Delta_s^{(T)} > \frac{3}{2}\log l$,

$$
\begin{aligned}
&\left|\left|w_s^{(T)}\right|\right| - \left|\left|w_s^{(0)}\right|\right| > \frac{3}{2}\log l \\
&\Rightarrow \left|\left|w_s^{(T)}\right|\right|^2 > \frac{9}{4}\log^2 l + \left|\left|w_s^{(0)}\right|\right|^2 + 3\log l \left|\left|w_s^{(0)}\right|\right| \\
&\Rightarrow \langle w_s^{(T+1)}, o_1\rangle^2 + \langle w_s^{(T+1)}, o_2\rangle^2 + \sum_{i=1}^{d-2}\langle w_s^{(T+1)}, q_i\rangle^2 > \frac{9}{4}\log^2 l + \left|\left|w_s^{(0)}\right|\right|^2 + 3\log l \left|\left|w_s^{(0)}\right|\right| \\
&\Rightarrow \langle w_s^{(T+1)}, o_1\rangle^2 + \left(\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(0)}, o_2\rangle\right)^2 + 2\left(\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(0)}, o_2\rangle\right)\langle w_s^{(0)}, o_2\rangle \\
&\quad + \langle w_s^{(0)}, o_2\rangle^2 + \sum_{i=1}^{d-2}\langle w_s^{(0)}, q_i\rangle^2 + \sum_{i=1}^{d-2}\left(\langle w_s^{(T+1)}, q_i\rangle - \langle w_s^{(0)}, q_i\rangle\right)^2 \\
&\quad + 2\sum_{i=1}^{d-2}\left(\langle w_s^{(T+1)}, q_i\rangle - \langle w_s^{(0)}, q_i\rangle\right)\langle w_s^{(0)}, q_i\rangle > \frac{9}{4}\log^2 l + \left|\left|w_s^{(0)}\right|\right|^2 + 3\log l \left|\left|w_s^{(0)}\right|\right| \\
&\Rightarrow \langle w_s^{(T+1)}, o_1\rangle^2 - \langle w_s^{(0)}, o_1\rangle^2 + \left(\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(0)}, o_2\rangle\right)^2 \\
&\quad + 2\left(\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(0)}, o_2\rangle\right)\langle w_s^{(0)}, o_2\rangle + \sum_{i=1}^{d-2}\left(\langle w_s^{(T+1)}, q_i\rangle - \langle w_s^{(0)}, q_i\rangle\right)^2 \\
&\quad + 2\sum_{i=1}^{d-2}\left(\langle w_s^{(T+1)}, q_i\rangle - \langle w_s^{(0)}, q_i\rangle\right)\langle w_s^{(0)}, q_i\rangle > \frac{9}{4}\log^2 l
\end{aligned}
$$

Now, $\forall q$, $\langle w_s^{(T+1)}, q\rangle - \langle w_s^{(0)}, q\rangle \leq O(\frac{1}{d^{3/2}}\log l)$ and, $\langle w_s^{(0)}, q\rangle < \frac{1}{2}\log l$.

Also, $\langle w_s^{(T+1)}, o_2\rangle - \langle w_s^{(0)}, o_2\rangle \leq O(\frac{1}{d^{1/2}}\log l)$ and, $\langle w_s^{(0)}, o_2\rangle < \frac{1}{2}\log l$.

Therefore,

$$\langle w_s^{(T+1)}, o_1\rangle^2 > \frac{9}{4}\log^2 l$$

$$\Rightarrow \langle w_s^{(T+1)}, o_1 \rangle > \frac{3}{2}\log l$$

On the other hand, $\forall q$, $\langle w_s^{(T+1)}, q \rangle \leq \frac{1}{2}\log l$.

Therefore, $\forall q$, $\langle w_s^{(T+1)}, o_1 - q \rangle > \log l$.

Hence, $p_1^{(s,T+1)} = 1$ and, $\forall (x, +1) \sim \mathcal{D}$, $G_1^{(s,T+1)}(x) > \frac{1}{2}$.

Now, as $\langle w_s^{(T+1)}, o_1 \rangle > \frac{3}{2}\log l$,

$$\langle w_s^{(T+1)}, o_1 \rangle - \langle w_s^{(0)}, o_1 \rangle > \log l \Rightarrow -\sum_{t=0}^{T} \langle \frac{\partial l}{\partial w_s^{(t)}}, o_1 \rangle \eta_r > \log l \Rightarrow \sum_{t=0}^{T} \delta_{1,q}^{(s,t)} > \frac{2}{\eta_r}\log l \Rightarrow \sum_{t=0}^{T} \sigma_1^{(s,t)} > \frac{2}{\eta_r}\log l$$

Now, as $\sigma_1^{(s,0)} = O(mC_2)$, for the selection of η_r and T, there is contradiction if there is only $o(1)$ fraction of $r \in [m]$ of s with $\langle w_r^{(s,0)}, o_1 \rangle \geq 0$. Therefore, there is $\Omega(1)$ fraction of $r \in [m]$ of s such that $\langle w_r^{(s,0)}, o_1 \rangle \geq 0$.

Therefore, we need $T' = O(kC_2 l^2 \sqrt{\log l}/\eta_e)$ iterations to show, $\langle w_r^{(s,T,T')}, o_1 \rangle = \Omega(kl^2 C_2 \sqrt{\log l})$ for $\Omega(1)$ fraction of $r \in [m]$ of s.

Similarly, we can complete the proof of the statements (iv), (v) and (vi).

□

I. Proof of Theorem 4.5

Proof. For the pruning ratio of $\rho = 1 - O(\frac{1}{k})$, $|S_1 \cap S_{k'}| = O(1)$ and $|S_2 \cap S_{k'}| = O(1)$.

Now, using Lemma 4.4 and following the same procedure as in the proof of Theorem 4.3 we can complete the proof. □