Patch-level Routing in Mixture-of-Experts is Provably Sample-efficient for Convolutional Neural Networks

Mohammed Nowaz Rabbani Chowdhury [1] **Shuai Zhang** [1] **Meng Wang** [1] **Sijia Liu** [2 3] **Pin-Yu Chen** [4]

Abstract

In deep learning, mixture-of-experts (MoE) activates one or few experts (sub-networks) on a per-sample or per-token basis, resulting in significant computation reduction. The recently proposed patch-level routing in MoE (pMoE) divides each input into n patches (or tokens) and sends l patches ($l \ll n$) to each expert through prioritized routing. pMoE has demonstrated great empirical success in reducing training and inference costs while maintaining test accuracy. However, the theoretical explanation of pMoE and the general MoE remains elusive. Focusing on a supervised classification task using a mixture of two-layer convolutional neural networks (CNNs), we show for the first time that pMoE provably reduces the required number of training samples to achieve desirable generalization (referred to as the sample complexity) by a factor in the polynomial order of n/l, and outperforms its single-expert counterpart of the same or even larger capacity. The advantage results from the discriminative routing property, which is justified in both theory and practice that pMoE routers can filter label-irrelevant patches and route similar class-discriminative patches to the same expert. Our experimental results on MNIST, CIFAR-10, and CelebA support our theoretical findings on pMoE's generalization and show that pMoE can avoid learning spurious correlations.

[1]Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, NY, USA [2]Department of Computer Science and Engineering, Michigan State University, MI, USA [3]MIT-IBM Watson AI Lab, IBM Research, MA, USA [4]IBM Research, Yorktown Heights, NY, USA. Correspondence to: Mohammed Nowaz Rabbani Chowdhury <chowdm2@rpi.edu>, Meng Wang <wangm7@rpi.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

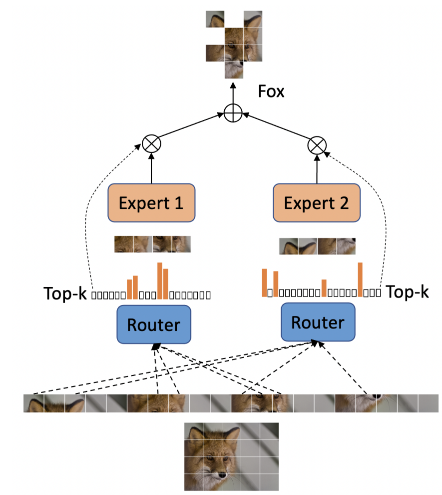

Figure 1. An illustration of pMoE. The image is divided into 20 patches while the router selects 4 of them for each expert.

1. Introduction

Deep learning has demonstrated exceptional empirical success in many applications at the cost of high computational and data requirements. To address this issue, mixture-of-experts (MoE) only activates partial regions of a neural network for each data point and significantly reduces the computational complexity of deep learning *without hurting the performance* in applications such as machine translation and natural image classification (Shazeer et al., 2017; Yang et al., 2019). A conventional MoE model contains multiple experts (subnetworks of the backbone architecture) and one learnable router that routes each input sample to a few but not all the experts (Ramachandran & Le, 2018). Position-wise MoE has been introduced in language models (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022), where the routing decisions are made on embeddings of different positions of the input separately rather than routing the entire text-input. Riquelme et al. (2021) extended it to vision models where the routing decisions are made on image patches. Zhou et al. (2022) further extended where the MoE layer has one router for each expert such that the router selects partial patches for the corresponding expert and discards the remaining patches. We termed this routing mode as *patch-level routing* and the MoE layer as *patch-level MoE* (pMoE) layer (see Figure 1 for an illustration of a pMoE). Notably, pMoE achieves the *same* test accuracy in vision tasks with 20% less training compute, and 50% less inference compute compared to its single-expert (i.e., one expert which is receiving all the patches of an input) counterpart of the same capacity (Riquelme et al., 2021).

Despite the empirical success of MoE, it remains elusive in theory, why can MoE maintain test accuracy while significantly reducing the amount of computation? To the best of our knowledge, only one recent work by Chen et al. (2022) shows theoretically that a conventional sample-wise MoE achieves higher test accuracy than convolutional neural networks (CNN) in a special setup of a binary classification task on data from linearly separable clusters. However, the sample-wise analyses by Chen et al. (2022) do not extend to patch-level MoE, which employ different routing strategies than conventional MoE, and their data model might not characterize some practical datasets. This paper addresses the following question theoretically:

How much computational resource does pMoE save from the single-expert counterpart while maintaining the same generalization guarantee?

In this paper, we consider a supervised binary classification task where each input sample consists of n equal-sized patches including *class-discriminative* patterns that determine the labels and *class-irrelevant* patterns that do not affect the labels. The neural network contains a pMoE layer[1] and multiple experts, each of which is a two-layer CNN[2] of the same architecture. The router sends l ($l \ll n$) patches to each expert. Although we consider a simplified neural network model to facilitate the formal analysis of pMoE, the insights are applicable to more general setups. Our major results include:

1. To the best of our knowledge, this paper provides **the first theoretical generalization analysis of pMoE**. Our analysis reveals that pMoE with two-layer CNNs as experts can achieve the same generalization performance as conventional CNN while reducing the sample complexity (the required number of training samples to learn a proper model) and model complexity. Specifically, we prove that as long as l is larger than a certain threshold, pMoE reduces the sample complexity and model complexity by a factor polynomial in n/l, indicating an improved generalization with a smaller l.

2. Characterization of the desired property of the pMoE router. We show that a desired pMoE router can dispatch the same class-discriminative patterns to the same expert and discard some class-irrelevant patterns. This discriminative property allows the experts to learn the class-discriminative patterns with reduced interference from irrelevant patterns, which in turn reduces the sample complexity and model complexity. We also prove theoretically that a separately trained pMoE router has the desired property and empirically verify this property on practical pMoE routers.

3. Experimental demonstration of reduced sample complexity by pMoE in deep CNN models. In addition to verifying our theoretical findings on synthetic data prepared from the MNIST dataset (LeCun et al., 2010), we demonstrate the sample efficiency of pMoE in learning some benchmark vision datasets (e.g., CIFAR-10 (Krizhevsky, 2009) and CelebA (Liu et al., 2015)) by replacing the last convolutional layer of a ten-layer wide residual network (WRN) (Zagoruyko & Komodakis, 2016) with a pMoE layer. These experiments not only verify our theoretical findings but also demonstrate the applicability of pMoE in reducing sample complexity in deep-CNN-based vision models, complementing the existing empirical success of pMoE with vision transformers.

2. Related Works

Mixture-of-Experts. MoE was first introduced in the 1990s with dense sample-wise routing, i.e. each input sample is routed to all the experts (Jacobs et al., 1991; Jordan & Jacobs, 1994; Chen et al., 1999; Tresp, 2000; Rasmussen & Ghahramani, 2001). Sparse sample-wise routing was later introduced (Bengio et al., 2013; Eigen et al., 2013), where each input sample activates few of the experts in an MoE layer both for joint training (Ramachandran & Le, 2018; Yang et al., 2019) and separate training of the router and experts (Collobert et al., 2001; 2003; Ahmed et al., 2016; Gross et al., 2017). Position/patch-wise MoE (i.e., pMoE) recently demonstrated success in large language and vision models (Shazeer et al., 2017; Lepikhin et al., 2020; Riquelme et al., 2021; Fedus et al., 2022). To solve the issue of load imbalance (Lewis et al., 2021), Zhou et al. (2022) introduces the *expert-choice routing* in pMoE, where each expert uses one router to select a fixed number of patches from the input. This paper analyzes the sparse patch-level MoE with expert-choice routing under both joint-training and separate-training setups.

Optimization and generalization analyses of neural networks (NN). Due to the significant nonconvexity of deep learning problem, the existing generalization analyses are

[1] In practice, pMoEs are usually placed in the last layers of deep models. Our analysis can be extended to this case as long as the input to the pMoE layer satisfies our data model (see Section 4.2).

[2] We consider CNN as expert due to its wide applications, especially in vision tasks. Moreover, the pMoE in (Riquelme et al., 2021; Zhou et al., 2022) uses two-layer Multi-Layer Perceptrons (MLPs) as experts in vision transformer (ViT), which operates on image patches. Hence, the MLPs in (Riquelme et al., 2021; Zhou et al., 2022) are effectively non-overlapping CNNs.

limited to linearized or shallow neural networks. The Neural-Tangent-Kernel (NTK) approach (Jacot et al., 2018; Lee et al., 2019; Du et al., 2019; Allen-Zhu et al., 2019b; Zou et al., 2020; Chizat et al., 2019; Ghorbani et al., 2021) considers strong over-parameterization and approximates the neural network by the first-order Taylor expansion. The NTK results are independent of the input data, and performance gaps in the representation power and generalization ability exist between the practical NN and the NTK results (Yehudai & Shamir, 2019; Ghorbani et al., 2019; 2020; Li et al., 2020; Malach et al., 2021). Nonlinear neural networks are analyzed recently through higher-order Taylor expansions (Allen-Zhu et al., 2019a; Bai & Lee, 2019; Arora et al., 2019; Ji & Telgarsky, 2019) or employing a model estimation approach from Gaussian input data (Zhong et al., 2017b;a; Zhang et al., 2020b;a; Fu et al., 2020; Li et al., 2022b), but these results are limited to two-layer networks with few papers on three-layer networks (Allen-Zhu et al., 2019a; Allen-Zhu & Li, 2019; 2020a; Li et al., 2022a).

The above works consider arbitrary input data or Gaussian input. To better characterize the practical generalization performance, some recent works analyze structured data models using approaches such as feature mapping (Li & Liang, 2018), where some of the initial model weights are close to data features, and feature learning (Daniely & Malach, 2020; Shalev-Shwartz et al., 2020; Shi et al., 2021; Allen-Zhu & Li, 2022; Li et al., 2023), where some weights gradually learn features during training. Among them, Allen-Zhu & Li (2020b); Brutzkus & Globerson (2021); Karp et al. (2021) analyze CNN on learning structured data composed of class-discriminative patterns that determine the labels and other label-irrelevant patterns. This paper extends the data models in Allen-Zhu & Li (2020b); Brutzkus & Globerson (2021); Karp et al. (2021) to a more general setup, and our analytical approach is a combination of feature learning in routers and feature mapping in experts for pMoE.

3. Problem Formulation

This paper considers the supervised binary classification[3] problem where given N i.i.d. training samples $\{(x_i, y_i)\}_{i=1}^N$ generated by an unknown distribution $\mathcal{D}$, the objective is to learn a neural network model that maps x to y for any (x, y) sampled from $\mathcal{D}$. Here, the input $x \in \mathbb{R}^{nd}$ has n disjoint patches, i.e., $x^\intercal = [x^{(1)\intercal}, x^{(2)\intercal}, ..., x^{(n)\intercal}]$, where $x^{(j)} \in \mathbb{R}^d$ denotes the j-th patch of x. $y \in \{+1, -1\}$ denotes the corresponding label.

[3]Our results can be extended to multiclass classification problems. See Section M in the Appendix for details.

3.1. Neural Network Models

We consider a pMoE architecture that includes k experts and the corresponding k routers. Each router selects l out of n ($l < n$) patches for each expert separately. Specifically, the router for each expert s ($s \in [k]$) contains a trainable gating kernel $w_s \in \mathbb{R}^d$. Given a sample x, the router computes a routing value $g_{j,s}(x) = \langle w_s, x^{(j)} \rangle$ for each patch j. Let $J_s(x)$ denote the index set of top-l values of $g_{j,s}$ among all the patches $j \in [n]$. Only patches with indices in $J_s(x)$ are routed to the expert s, multiplied by a gating value $G_{j,s}(x)$, which are selected differently in different pMoE models.

Each expert is a two-layer CNN with the same architecture. Let m denote the total number of neurons in all the experts. Then each expert contains (m/k) neurons. Let $w_{r,s} \in \mathbb{R}^d$ and $a_{r,s} \in \mathbb{R}$ denote the hidden layer and output layer weights for neuron r ($r \in [m/k]$) in expert s ($s \in [k]$), respectively. The activation function is the rectified linear unit (ReLU), where $\textbf{ReLU}(z) = \max(0, z)$.

Let $\theta = \{a_{r,s}, w_{r,s}, w_s, \forall s \in [k], \forall r \in [m/k]\}$ include all the trainable weights. The pMoE model denoted as f_M, is defined as follows:

$$f_M(\theta, x) = \sum_{s=1}^{k} \sum_{r=1}^{\frac{m}{k}} \frac{a_{r,s}}{l} \sum_{j \in J_s(w_s, x)} \textbf{ReLU}(\langle w_{r,s}, x^{(j)} \rangle) G_{j,s}(w_s, x) \tag{1}$$

An illustration of (1) is given in Figure 2.

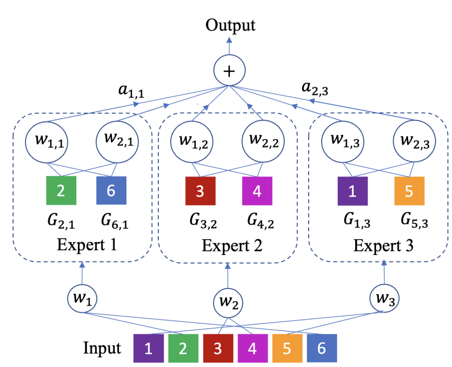

Figure 2. An illustration of the pMoE model in (1) with $k = 3, m = 6, n = 6$, and $l = 2$.

The learning problem solves the following empirical risk minimization problem with the logistic loss function,

$$\min_{\theta} : \quad L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + e^{-y_i f_M(\theta, x_i)}\right) \tag{2}$$

We consider two different training modes of pMoE, *Separate-training* and *Joint-training* of the routers and the experts. We also consider the conventional CNN architecture for comparison.

(I) **Separate-training pMoE**: Under the setup of the so-called *hard mixtures of experts* (Collobert et al., 2003; Ahmed et al., 2016; Gross et al., 2017), the router weights w_s are trained first and then fixed when training the weights of the experts. In this case, the gating values are set as

$$G_{j,s}(w_s, x) \equiv 1, \ \forall j, s, x \tag{3}$$

We select $k = 2$ in this case to simplify the analysis.

(II) **Joint-training pMoE**: The routers and the experts are learned jointly, see, e.g., (Lepikhin et al., 2020; Riquelme et al., 2021; Fedus et al., 2022). Here, the gating values are softmax functions with

$$G_{j,s}(w_s, x) = e^{g_{j,s}(x)} / (\sum_{i \in J_s(x)} g_{i,s}(x)) \tag{4}$$

(III) **CNN single-expert counterpart**: The conventional two-layer CNN with m neurons, denoted as f_C, satisfies,

$$f_C(\theta, x) = \sum_{r=1}^{m} a_r \left(\frac{1}{n} \sum_{j=1}^{n} \textbf{ReLU}(\langle w_r, x^{(j)} \rangle) \right) \tag{5}$$

Eq. (5) can be viewed as a special case of (1) when there is only one expert ($k = 1$), and all the patches are sent to the expert ($l = n$) with gating values $G_{j,s} \equiv 1$.

Let $\tilde{\theta}$ denote the parameters of the learned model by solving (1). The predicted label for a test sample x by the learned model is sign$(f_M(\tilde{\theta}, x))$. The generalization accuracy, i.e., the fraction of correct predictions of all test samples equals $\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}} [y f_M(\theta, x) > 0]$. This paper studies both separate and joint training of pMoE and compares their performance with CNN, from the perspective of sample complexity to achieve a desirable generalization accuracy.

3.2. Training Algorithms

In the following algorithms, we fix the output layer weights $a_{r,s}$ and a_r at their initial values randomly sampled from the standard Gaussian distribution $\mathcal{N}(0, 1)$ and do not update them during the training. This is a typical simplification when analyzing NN, as used in (Li & Liang, 2018; Brutzkus et al., 2018; Allen-Zhu et al., 2019a; Arora et al., 2019).

(I) Separate-training pMoE: The routers are separately trained using N_r training samples ($N_r < N$), denoted by $\{(x_i, y_i)\}_{i=1}^{N_r}$ without loss of generality. The gating kernels w_1 and w_2 are obtained by solving the following minimization problem:

$$\min_{w_1, w_2} : \ l_r(w_1, w_2) = -\frac{1}{N_r} \sum_{i=1}^{N_r} y_i \langle w_1 - w_2, \sum_{j=1}^{n} x_i^{(j)} \rangle \tag{6}$$

To solve (6), we implement the mini-batch SGD with batch size B_r for $T_r = N_r/B_r$ iterations, starting from the random initialization as follows:

$$w_s^{(0)} \sim \mathcal{N}(0, \sigma_r^2 \mathbb{I}_{d\times d}), \forall s \in [2] \tag{7}$$

where, $\sigma_r = \Theta\big(1/(n^2 \log(\text{poly}(n))\sqrt{d})\big)$.

After learning the routers, we train the hidden-layer weights $w_{r,s}$ by solving (2) while fixing w_1 and w_2. We implement mini-batch SGD of batch size B for $T = N/B$ iterations starting from the initialization

$$w_{r,s}^{(0)} \sim \mathcal{N}(0, \frac{1}{m} \mathbb{I}_{d\times d}), \forall s \in [2], \forall r \in [m/2] \tag{8}$$

(II) Joint-training pMoE: w_s and $w_{r,s}$ in (1) are updated simultaneously by mini-batch SGD of batch size B for $T = N/B$ iterations starting from the initialization in (7) and (8).

(III) CNN: w_r in (5) are updated by mini-batch SGD of batch size B for $T = N/B$ iterations starting from the initialization in (8).

4. Theoretical Results

4.1. Key Findings At-a-glance

Before defining the data model assumptions and rationale in Section 4.2 and presenting the formal results in 4.3, we first summarize our key findings. We assume that the data patches are sampled from either *class-discriminative* patterns that determine the labels or a possibly infinite number of *class-irrelevant* patterns that have no impact on the label. The parameter δ (defined in (9)) is inversely related to the separation among patterns, i.e., δ decreases when (i) the separation among class-discriminative patterns increases, and/or (ii) the separation between class-discriminative and class-irrelevant patterns increases. The key findings are as follows.

(I). A properly trained patch-level router sends class-discriminative patches of one class to the same expert while dropping some class-irrelevant patches. We prove that separate-training pMoE routes class-discriminative patches of the class with label $y = +1$ (or the class with label $y = -1$) to the expert 1 (or the expert 2) respectively, and the class-irrelevant patterns that are sufficiently away from class-discriminative patterns are not routed to any expert (Lemma 4.1). This discriminative routing property is also verified empirically for joint-training pMoE (see section 5.1). Therefore, pMoE effectively reduces the interference by irrelevant patches when each expert learns the class-discriminative patterns. Moreover, we show empirically that pMoE can remove class-irrelevant patches that are spuriously correlated with class labels and thus can avoid learning from spuriously correlated features of the data.

(II). Both the sample complexity and the required number of hidden nodes of pMoE reduce by a polynomial factor of n/l over CNN. We prove that as long as l, the number of patches per expert, is greater than a threshold (that decreases as the separation between class-discriminative and class-irrelevant patterns increases), the sample complexity and the required number of neurons of learning pMoE are $\Omega(l^8)$ and $\Omega(l^{10})$ respectively. In contrast, the sample and model complexities of the CNN are $\Omega(n^8)$ and $\Omega(n^{10})$ respectively, indicating improved generalization by pMoE.

(III). Larger separation among class-discriminative and class-irrelevant patterns reduces the sample complexity and model complexity of pMoE. Both the sample complexity and the required number of neurons of pMoE is polynomial in δ, which decreases when the separation among patterns increases.

4.2. Data Model Assumptions and Rationale

The input x is comprised of one class-discriminative pattern and $n-1$ class-irrelevant patterns, and the label y is determined by the class-discriminative pattern only.

Distributions of class-discriminative patterns: The unit vectors o_1 and $o_2 \in \mathbb{R}^d$ denote the *class-discriminative* patterns that determine the labels. The separation between o_1 and o_2 is measured as $\delta_d := \langle o_1, o_2 \rangle \in (-1, 1)$. o_1 and o_2 are equally distributed in the samples, and each sample has exactly one of them. If x contains o_1 (or o_2), then y is $+1$ (or -1).

Distributions of class-irrelevant patterns. *Class-irrelevant* patterns are unit vectors in $\mathbb{R}^d$ belonging to p disjoint pattern sets $S_1, S_2, ..., S_p$, and these patterns distribute equally for both classes. δ_r measures the separation between class-discriminative patterns and class-irrelevant patterns, where $|\langle o_i, q \rangle| \leq \delta_r$, $\forall i \in [2]$, $\forall q \in S_j$, $j = 1, ..., p$. Each S_j belongs to a ball with a diameter of $\Theta(\sqrt{(1-\delta_r^2)/dp^2})$. Note that NO separation among class-irrelevant patterns themselves is required.

The rationale of our data model. The data distribution $\mathcal{D}$ captures the locality of the label-defining features in image data. It is motivated by and extended from the data distributions in recent theoretical frameworks (Yu et al., 2019; Brutzkus & Globerson, 2021; Karp et al., 2021; Chen et al., 2022). Specifically, Yu et al. (2019) and Brutzkus & Globerson (2021) require orthogonal patterns, i.e., δ_r and δ_d are both 0, and there are only a fixed number of non-discriminative patterns. Karp et al. (2021) and Chen et al. (2022) assume that $\delta_d = -1$ and a possibly infinite number of patterns drawn from zero-mean Gaussian distribution. In our model, δ_d takes any value in $(-1, 1)$, and the class-irrelevant patterns can be drawn from p pattern sets that contain an infinite number of patterns that are not necessarily Gaussian or orthogonal.

Define

$$\delta = 1/(1 - \max(\delta_d^2, \delta_r^2)) \tag{9}$$

δ decreases if (1) o_1 and o_2 are more separated from each other, and (2) Both o_1 and o_2 are more separated from any set S_i, $i \in [p]$. We also define an integer l^* ($l^* \leq n$) that measures *the maximum number of class-irrelevant patterns per sample that are sufficiently closer to o_1 than o_2, and vice versa.* Specifically, a class-irrelevant pattern q is called δ'-closer ($\delta' > 0$) to o_1 than o_2, if $\langle o_1 - o_2, q \rangle > \delta'$ holds. Similarly, q is δ'-closer to o_2 than o_1 if $\langle o_2 - o_1, q \rangle > \delta'$. Then, let $l^* - 1$ be the maximum number of class-irrelevant patches that are either δ'-closer to o_1 than o_2 or vice versa with $\delta' = \Theta(1 - \delta_d)$ in any x sampled from $\mathcal{D}$. l^* depends on $\mathcal{D}$ and δ_d. When $\mathcal{D}$ is fixed, a smaller δ_d corresponds to a larger separation between o_1 and o_2 and leads to a small l^*. In contrast to linearly separable data in (Yu et al., 2019; Brutzkus et al., 2018; Chen et al., 2022), our data model is **NOT** linearly separable as long as $l^* = \Omega(1)$ (see section K in Appendix for the proof).

4.3. Main Theoretical Results

4.3.1. Generalization Guarantee of Separate-training pMoE

Lemma 4.1 shows that as long as the number of patches per expert, l, is greater than l^*, then the separately learned routers by solving (6) always send o_1 to expert 1 and o_2 to expert 2. Based on this discriminative property of the learned routers, Theorem 4.2 then quantifies the sample complexity and network size of separate-training pMoE to achieve a desired generalization error ϵ. Theorem 4.3 quantifies the sample and model complexities of CNN for comparison.

Lemma 4.1 (Discriminative Property of Separately Trained Routers). *For every $l \geq l^*$, w.h.p. over the random initialization defined in (7), after doing mini-batch SGD with batch-size $B_r = \Omega\left(n^2/(1-\delta_d)^2\right)$ and learning rate $\eta_r = \Theta(1/n)$, for $T_r = \Omega\left(1/(1-\delta_d)\right)$ iterations, the returned w_1 and w_2 satisfy*

$$\underset{j \in [n]}{arg}\ (x^{(j)} = o_1) \in J_1(w_1, x), \quad \forall (x, y = +1) \sim \mathcal{D}$$

$$\underset{j \in [n]}{arg}\ (x^{(j)} = o_2) \in J_2(w_2, x), \quad \forall (x, y = -1) \sim \mathcal{D}$$

i.e., the learned routers always send o_1 to expert 1 and o_2 to expert 2.

The main idea in proving Lemma 4.1 is to show that the gradient in each iteration has a large component along the directions of o_1 and o_2. Then after enough iterations, the inner product of w_1 and o_1 (similarly, w_2 and o_2) is sufficiently large. The intuition of requiring $l \geq l^*$ is that because there are at most $l^* - 1$ class-irrelevant patches sufficiently closer to o_1 than o_2 (or vice versa), then sending $l \geq l^*$ patches to one expert will ensure that one of them is o_1 (or o_2). Note that the batch size B_r and the number of iterations T_r depend on δ_d, the separation between o_1 and o_2, but are independent of the separation between class-discriminative and class-irrelevant patterns.

We then show that the separate-training pMoE reduces both the sample complexity and the required model size (Theorem 4.2) compared to the CNN (Theorem 4.3).

Theorem 4.2 (Generalization guarantee of separate-training pMoE)**.** *For every $\epsilon > 0$ and $l \geq l^*$, for every $m \geq M_S = \Omega\left(l^{10}p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_S = \Omega(l^8 p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \Omega\left(l^4p^6\delta^3/\epsilon^8\right)$ and the learning rate $\eta = O\big(1/(m poly(l, p, \delta, 1/\epsilon, \log m))\big)$ for $T = O\left(l^4p^6\delta^3/\epsilon^8\right)$ iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[y f_M(\theta^{(T)}, x) > 0\right] \geq 1 - \epsilon$$

Theorem 4.2 implies that to achieve generalization error ϵ by a separate-training pMoE, we need $N_S = \Omega(l^8p^{12}\delta^6/\epsilon^{16})$ training samples and $M_S = \Omega\left(l^{10}p^{12}\delta^6/\epsilon^{16}\right)$ hidden nodes. Therefore, both N_S and M_S increase polynomially with the number of patches l sent to each expert. Moreover, both N_S and M_S are polynomial in δ defined in (9), indicating an improved generalization performance with stronger separation among patterns.

The proof of Theorem 4.2 is inspired by Li & Liang (2018), which analyzes the generalization performance of fully-connected neural networks (FCN) on structured data, but we have new technical contributions in analyzing pMoE models. In addition to analyzing the pMoE routers (Lemma 4.1), which do not appear in the FCN analysis, our analyses also significantly relax the separation requirement on the data, compared with that by Li & Liang (2018). For example, Li & Liang (2018) requires the separation between the two classes, measured by the smallest ℓ_2-norm distance of two points in different classes, being $\Omega(n)$ to obtain a sample complexity bound of poly(n) for the binary classification task. In contrast, the separation between the two classes in our data model is $\min\{\sqrt{2(1-\delta_d)}, 2\sqrt{1-\delta_r}\}$, much less than $\Omega(n)$ required by Li & Liang (2018).

Theorem 4.3 (Generalization guarantee of CNN)**.** *For every $\epsilon > 0$, for every $m \geq M_C = \Omega\left(n^{10}p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_C = \Omega(n^8p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \Omega\left(n^4p^6\delta^3/\epsilon^8\right)$ and the learning rate $\eta = O\big(1/(m poly(n, p, \delta, 1/\epsilon, \log m))\big)$ for $T = O\left(n^4p^6\delta^3/\epsilon^8\right)$ iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[y f_C(\theta^{(T)}, x) > 0\right] \geq 1 - \epsilon$$

Theorem 4.3 implies that to achieve a generalization error ϵ using CNN in (5), we need $N_C = \Omega(n^8p^{12}\delta^6/\epsilon^{16})$ training samples and $M_C = \Omega\left(n^{10}p^{12}\delta^6/\epsilon^{16}\right)$ neurons.

Sample-complexity gap between single CNN and mixture of CNNs. From Theorem 4.2 and Theorem 4.3, the sample-complexity ratio of the CNN to the separate-training pMoE is $N_C/N_S = \Theta\big((n/l)^8\big)$. Similarly, the required number of neurons is reduced by a factor of $M_C/M_S = \Theta\big((n/l)^{10}\big)$ in separate-training pMoE[4].

4.3.2. Generalization Guarantee of Joint-training pMoE with Proper Routers

Theorem 4.5 characterizes the generalization performance of joint-training pMoE assuming the routers are properly trained in the sense that after some SGD iterations, for each class at least one of the k experts receives all class-discriminative patches of that class with the largest gating-value (see Assumption 4.4).

Assumption 4.4. There exists an integer $T' < T$ such that for all $t \geq T'$, it holds that:

There exists an expert $s \in [k]$ s.t. $\forall (x, y = +1) \sim \mathcal{D}$,

$$j_{o_1} \in J_s(w_s^{(t)}, x), \text{ and } G_{j_{o_1},s}^{(t)}(x) \geq G_{j,s}^{(t)}(x)$$

and an expert $s \in [k]$ s.t. $\forall (x, y = -1) \sim \mathcal{D}$,

$$j_{o_2} \in J_s(w_s^{(t)}, x), \text{ and } G_{j_{o_2},s}^{(t)}(x) \geq G_{j,s}^{(t)}(x)$$

where j_{o_1} (j_{o_2}) denotes the index of the class-discriminative pattern o_1 (o_2), $G_{j,s}^{(t)}(x)$ is the gating output of patch $j \in J_s(w_s^{(t)}, x)$ of sample x for expert s at the iteration t, and $w_s^{(t)}$ is the gating kernel for expert s at iteration t.

Assumption 4.4 is required in proving Theorem 4.5 because of the difficulty of tracking the dynamics of the routers in joint-training pMoE. Assumption 4.4 is verified on empirical experiments in Section 5.1, while its theoretical proof is left for future work.

Theorem 4.5 (Generalization guarantee of joint-training pMoE)**.** *Suppose Assumption 4.4 hold. Then for every $\epsilon > 0$, for every $m \geq M_J = \Omega\left(k^3n^2l^6p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_J = \Omega(k^4l^6p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \Omega\left(k^2l^4p^6\delta^3/\epsilon^8\right)$ and the learning rate*

[4]The bounds for the sample complexity and model size in Theorem 4.2 and Theorem 4.3 are sufficient but not necessary. Thus, rigorously speaking, one can not compare sufficient conditions only. In our analysis, however, the bounds for MoE and CNN are derived with exactly the same technique with the only difference to handle the routers. Therefore, it is fair to compare these two bounds to show the advantage of pMoE.

Table 1. Computational complexity of pMoE and CNN.

	pMoE			CNN
Complexity to achieve ϵ error (Complx/Iter $\times$ T)	Separate-training	Joint-training		
	$O(Bml^5d/\epsilon^8)$	$O(Bmk^2l^3d/\epsilon^8)$		$O(Bmn^5d/\epsilon^8)$
Complexity per Iteration (Complx/Iter)		Router	Expert	
	$O(Bmld)$	$O(Bknd)$ (Forward pass) $O(Bkl^2d)$ (Backward pass)	$O(Bmld)$	$O(Bmnd)$
Iteration required to converge with ϵ error (T)	$O(l^4/\epsilon^8)$	$O(k^2l^2/\epsilon^8)$		$O(n^4/\epsilon^8)$

$\eta = O\big(1/(mpoly(l,p,\delta,1/\epsilon,\log m))\big)$ *for* $T = O\left(k^2l^2p^6\delta^3/\epsilon^8\right)$ *iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[yf_M(\theta^{(T)},x)>0\right]\geq 1-\epsilon$$

Theorem 4.5 indicates that, with proper routers, joint-training pMoE needs $N_J=\Omega(k^4l^6p^{12}\delta^6/\epsilon^{16})$ training samples and $M_J=\Omega\left(k^3n^2l^6p^{12}\delta^6/\epsilon^{16}\right)$ neurons to achieve ϵ generalization error. Compared with CNN in Theorem 4.3, joint-training pMoE reduces the sample complexity and model size by a factor of $\Theta(n^8/k^4l^6)$ and $\Theta(n^{10}/k^3l^6)$, respectively. With more experts (a larger k), it is easier to satisfy Assumption 4.4 to learn proper routers but requires larger sample and model complexities. When the number of samples is fixed, the expression of N_J also indicates that ϵ sales as $k^{1/4}l^{3/8}$, corresponding to an improved generalization when k and l decrease.

We provide the end-to-end computational complexity comparison between the analyzed pMoE models and general CNN model in Table 1 (see section N in Appendix for details). The results in Table 1 indicates that the computational complexity in joint-training pMoE is reduced by a factor of $O(n^5/k^2l^3)$ compared with CNN. Similarly, the reduction of computational complexity of separate-training pMoE is $O(n^5/l^5)$.

5. Experimental Results

5.1. pMoE of Two-layer CNN

Dataset: We verify our theoretical findings about the model in (1) on synthetic data prepared from MNIST (LeCun et al., 2010) data set. Each sample contains $n=16$ patches with patch size $d=28\times 28$. Each patch is drawn from the MNIST dataset. See Figure 3 as an example. We treat the digits "**1**" and "**0**" as the class-discriminative patterns o_1 and o_2, respectively. Each of the digits from "**2**" to "**9**" represents a class-irrelevant pattern set.

Setup: We compare separate-training pMoE, joint-training pMoE, and CNN with similar model sizes. The separate-training pMoE contains *two* experts with 20 hidden nodes in each expert. The joint-training pMoE has eight experts with five hidden nodes per expert. The CNN has 40 hidden nodes. All are trained using SGD with $\eta=0.2$ until zero training error. pMoE converges much faster than CNN, which takes 150 epochs. Before training the experts in the separate-training pMoE, we train the router for 100 epochs. The models are evaluated on 1000 test samples.

Generalization performance: Figure 4 compares the test accuracy of the three models, where $l=2$ and $l=6$ for separate-training and joint-training pMoE, respectively. The error bars show the mean plus/minus one standard deviation

Figure 3. Sample image of the synthetic data from MNIST. Class label is "1".

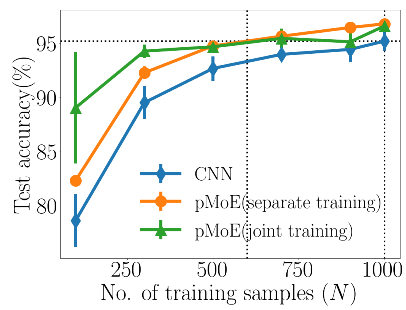

Figure 4. Generalization performance of pMoE and CNN with a similar model size

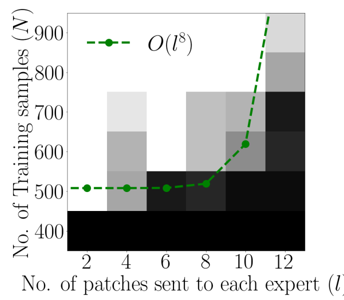

Figure 5. Phase transition of sample complexity with l in separate-training pMoE

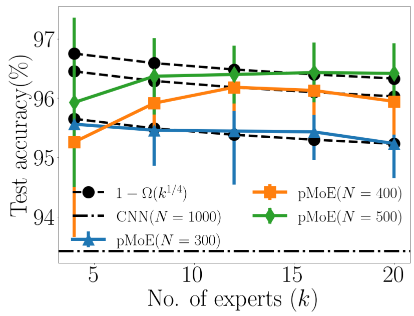

Figure 6. Change of test accuracy in joint-training pMoE with k for fixed sample sizes

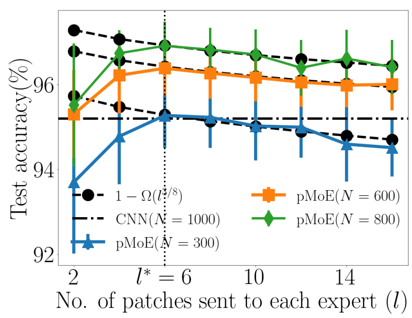

Figure 7. Change of test accuracy in joint-training pMoE with l for fixed sample sizes

of five independent experiments. pMoE outperforms CNN with the same number of training samples. pMoE only requires 60% of the training samples needed by CNN to achieve 95% test accuracy.

Figure 5 shows the sample complexity of separate-training pMoE with respect to l. Each block represents 20 independent trials. A white block indicates all success, and a black block indicates all failure. The sample complexity is polynomial in l, verifying Theorem 4.2. Figure 7 and 6 show the test accuracy of joint-training pMoE with a fixed sample size when l and k change, respectively. When l is greater than l^*, which is 6 in Figure 7, the test accuracy matches our predicted order. Similarly, the dependence on k also matches our prediction, when k is large enough to make Assumption 4.4 hold.

Router performance: Figure 8 verifies the discriminative property of separately trained routers (Lemma 4.1) by showing the percentage of testing data that have class-discriminative patterns (o_1 and o_2) in top l patches of the separately trained router. With very few training samples (such as 300), one can already learn a proper router that has discriminative patterns in top-4 patches for 95% of data. Figure 9 verifies the discriminative property of jointly trained routers (Assumption 4.4). With only 300 training samples, the jointly trained router dispatches o_1 with the largest gating value to a particular expert for 95% of class-1 data and similarly for o_2 in 92% of class-2 data.

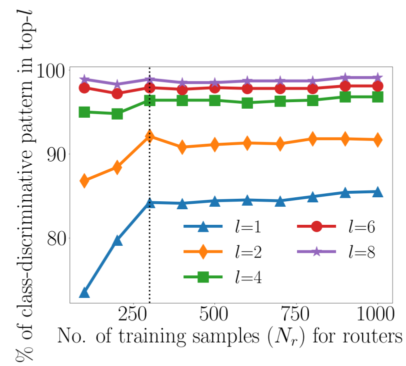

Figure 8. Percentage of properly routed discriminative patterns by a separately trained router.

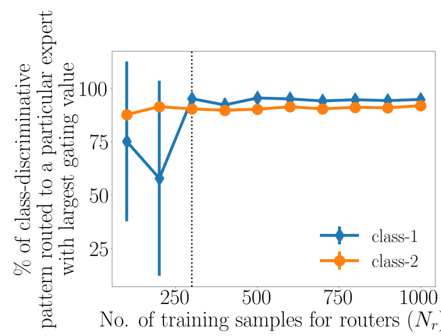

Figure 9. Percentage of properly routed discriminative patterns by a jointly trained router. $l = 6$.

5.2. pMoE of Wide Residual Networks (WRNs)

Neural network model: We employ the 10-layer WRN (Zagoruyko & Komodakis, 2016) with a widening factor of 10 as the expert. We construct a patch-level MoE counterpart of WRN, referred to as WRN-pMoE, by replacing the last convolutional layer of WRN with an pMoE layer of an equal number of trainable parameters (see Figure 18 in Appendix for an illustration). WRN-pMoE is trained with the joint-training method[5]. All the results are averaged over five independent experiments.

Datasets: We consider both CelebA (Liu et al., 2015) and CIFAR-10 datasets. The experiments on CIFAR-10 are deferred to the Appendix (see section A). We down-sample the images of CelebA to 64×64. The last convolutional layer of WRN receives a $(16\times16\times640)$ dimensional feature map. The feature map is divided into 16 patches with size $4 \times 4 \times 640$ in WRN-pMoE. $k = 8$ and $l = 2$ for the pMoE layer.

[5]Code is available at `https://github.com/nowazrabbani/pMoE_CNN`

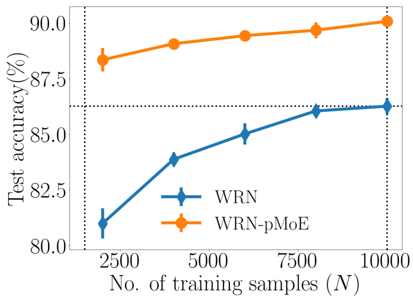

Figure 10. Classification accuracy of WRN-pMoE and WRN on "smiling" in CelebA

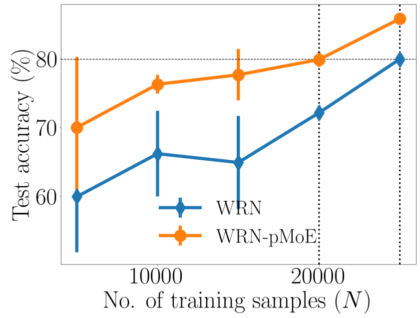

Figure 11. Classification accuracy of WRN-pMoE and WRN on "smiling" when spuriously correlated with "black hair" in CelebA

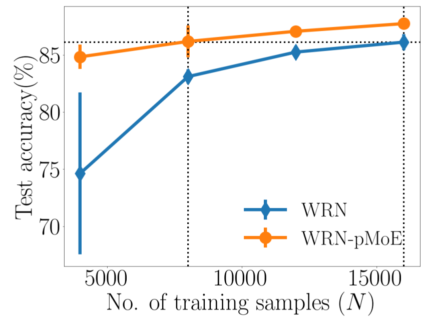

Figure 12. Classification accuracy of WRN-pMoE and WRN on multiclass classification in CelebA

Table 2. Comparison of training compute of WRN and WRN-pMoE.

No. of training samples	Convergence time (sec)		Training FLOPs ($\times 10^{15}$)	
	WRN	WRN-pMoE	WRN	WRN-pMoE
4000	260	**156**	6	**3.5**
8000	324	**192**	7.5	**4.4**
12000	468	**280**	11	**6.4**
16000	630	**368**	15	**8.5**

Performance Comparison: Figure 10 shows the test accuracy of the binary classification problem on the attribute "smiling." WRN-pMoE requires less than *one-fifth* of the training samples needed by WRN to achieve 86% accuracy. Figure 11 shows the performance when the training data contain spurious correlations with the hair color as a spurious attribute. Specifically, 95% of the training images with the attribute "smiling" also have the attribute "black hair," while 95% of the training images with the attribute "not-smiling" have the attribute "blond hair." The models may learn the hair-color attribute rather than "smiling" due to spurious correlation and, thus, the test accuracies are lower in Figure 11 than those in Figure 10. Nevertheless, WRN-pMoE outperforms WRN and reduces the sample complexity to achieve the same accuracy.

Figure 12 shows the test accuracy of multiclass classification (four classes with class attributes: "Not smiling, Eyeglass," "Smiling, Eyeglass," "Smiling, No eyeglass," and "Not smiling, No eyeglass") in CelebA. The results are consistent with the binary classification results. Furthermore, Table 2 empirically verifies the computational efficiency of WRN-pMoE over WRN on multiclass classification in CelebA[6]. Even with same number of training samples, WRN-pMoE is still more computationally efficient than WRN, because WRN-pMoE requires fewer iterations to converge and has a lower per-iteration cost.

[6]An NVIDIA RTX 4500 GPU was used to run the experiments, training FLOPs are calculated as Training FLOPs = Training time (second) × Number of GPUs × peak FLOP/second × GPU utilization rate

6. Conclusion

MoE reduces computational costs significantly without hurting the generalization performance in various empirical studies, but the theoretical explanation is mostly elusive. This paper provides the first theoretical analysis of patch-level MoE and proves its savings in sample complexity and model size quantitatively compared with the single-expert counterpart. Although centered on a classification task using a mixture of two-layer CNNs, our theoretical insights are verified empirically on deep architectures and multiple datasets. Future works include analyzing other MoE architectures such as MoE in Vision Transformer (ViT) and connecting MoE with other sparsification methods to further reduce the computation.

Acknowledgements

This work was supported by AFOSR FA9550-20-1-0122, NSF 1932196 and the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons). We thank Yihua Zhang at Michigan State University for the help in experiments with CelebA dataset. We thank all anonymous reviewers.

References

Ahmed, K., Baig, M. H., and Torresani, L. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, pp. 516–532. Springer, 2016.

Allen-Zhu, Z. and Li, Y. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

Allen-Zhu, Z. and Li, Y. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020a.

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020b.

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., 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.

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.

Bai, Y. and Lee, J. D. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. In *International Conference on Learning Representations*, 2019.

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

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, K., Xu, L., and Chi, H. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252, 1999.

Chen, Z., Deng, Y., Wu, Y., Gu, Q., and Li, Y. Towards understanding mixture of experts in deep learning. *arXiv preprint arXiv:2208.02813*, 2022.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Collobert, R., Bengio, S., and Bengio, Y. A parallel mixture of SVMs for very large scale problems. *Advances in Neural Information Processing Systems*, 14, 2001.

Collobert, R., Bengio, Y., and Bengio, S. Scaling large learning problems with hard parallel mixtures. *International Journal of pattern recognition and artificial intelligence*, 17(03):349–365, 2003.

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

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.

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

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.

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.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Limitations of lazy training of two-layers neural network. *Advances in Neural Information Processing Systems*, 32, 2019.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems*, 33: 14820–14830, 2020.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 2021.

Gross, S., Ranzato, M., and Szlam, A. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6865–6873, 2017.

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

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

Ji, Z. and Telgarsky, M. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *International Conference on Learning Representations*, 2019.

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2): 181–214, 1994.

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.

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

LeCun, Y., Cortes, C., and Burges, C. MNIST handwritten digit database. AT&T labs [online]. available http. *yann. lecun. com/exdb/mnist*, 2010.

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, 2019.

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, 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, 2022a.

Li, H., Zhang, S., and Wang, M. Learning and generalization of one-hidden-layer neural networks, going beyond standard gaussian data. In *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, pp. 37–42. IEEE, 2022b.

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

Li, Y., Ma, T., and Zhang, H. R. Learning over-parametrized two-layer neural networks beyond NTK. In *Conference on learning theory*, pp. 2613–2682. PMLR, 2020.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Malach, E., Kamath, P., Abbe, E., and Srebro, N. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pp. 7379–7389. PMLR, 2021.

Ramachandran, P. and Le, Q. V. Diversity and depth in per-example routing models. In *International Conference on Learning Representations*, 2018.

Rasmussen, C. and Ghahramani, Z. Infinite mixtures of gaussian process experts. *Advances in neural information processing systems*, 14, 2001.

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.

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.

Tresp, V. Mixtures of gaussian processes. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/file/9fdb62f932adf55af2c0e09e55861964-Paper.pdf.

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.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019.

Yehudai, G. and Shamir, O. On the power and limitations of random features for understanding neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Yu, B., Zhang, J., and Zhu, Z. On the learning dynamics of two-layer nonlinear convolutional neural networks. *arXiv preprint arXiv:1905.10157*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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.

Zhong, K., Song, Z., and Dhillon, I. S. Learning non-overlapping convolutional neural networks with multiple kernels. *arXiv preprint arXiv:1711.03440*, 2017a.

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, 2017b.

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.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109(3):467–492, 2020.

A. Experiments on CIFAR-10 Datasets

We also compare WRN and WRN-pMoE on CIFAR-10-based datasets. To better reflect local features, in addition to the original CIFAR-10, we adopt techniques of Karp et al. (2021) to generate two datasets based on CIFAR-10:

1. CIFAR-10 with IMAGENET noise. Each CIFAR-10 image is down-sampled to size 16×16 and placed at a random location of a background image chosen from ImageNet Plants synset. Figure 13(c) shows an example image of this dataset.

2. CIFAR-VEHICLES. Each vehicle image of CIFAR-10 is down-sampled to size 16×16 and placed in one quadrant of an image randomly where the other quadrants are randomly filled with down-sampled animal images in CIFAR-10. See Figure 13(b) for a sample image.

The last convolutional layer of WRN receives a $(8 \times 8 \times 640)$ dimensional feature map. In WRN-pMoE we divide this feature map into 64 patches with size $(1 \times 1 \times 640)$. The MoE layer of WRN-pMoE contains $k = 4$ experts with each expert receiving $l = 16$ patches.

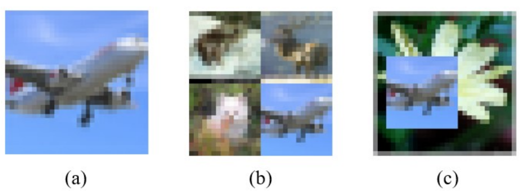

Figure 13. Example images from (a) CIFAR-10, (b) CIFAR-VEHICLES, and (c) CIFAR-10, IMAGENET noise datasets

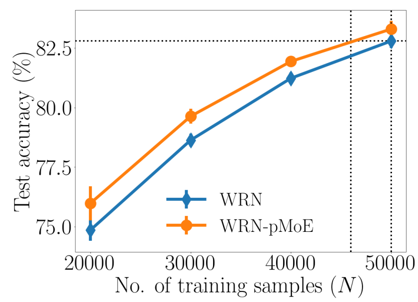

Figure 14. Ten-classification accuracy of WRN and WRN-pMoE on CIFAR-10

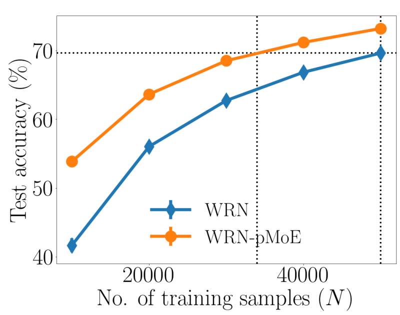

Figure 15. Ten-classification accuracy of WRN and WRN-pMoE on CIFAR-10, IMAGENET noise

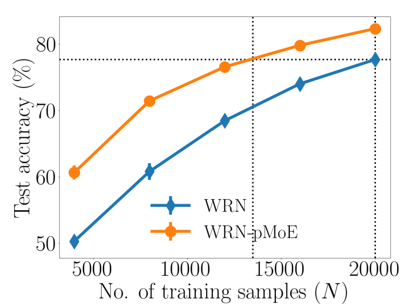

Figure 16. Four-classification accuracy of WRN and WRN-pMoE on CIFAR-VEHICLES

Figures 14, 15, and 16 compare the test accuracy of WRN and WRN-pMoE for the ten-classification problem on CIFAR10 and CIFAR-10 with IMAGENET noise, and the four-classification problem in CIFAR-VEHICLES, respectively. WRN-pMoE outperforms WRN in all these datasets, indicating reduced sample complexity using the pMoE layer. The performance gap is more significant in the other two datasets than the original CIFAR-10 dataset. That is because these constructed datasets contain local features, and the pMoE layer has a clear advantage in learning local features effectively.

B. Preliminaries

The loss function for SGD at iteration t with minibatch $\mathcal{B}_t$:

$$\mathcal{L}(\theta^{(t)}) := \frac{1}{B} \sum_{(x,y) \in \mathcal{B}_t} \log\left(1 + e^{-y f_M(\theta^{(t)}, x)}\right) \tag{10}$$

For the router-training in separate-training pMoE, the loss function of SGD at iteration t with minibatch $\mathcal{B}_t^r$:

$$\ell_r(w_1^{(t)}, w_2^{(t)}) := -\frac{1}{B_r} \sum_{(x,y)\in\mathcal{B}_t^r} y\langle w_1^{(t)} - w_2^{(t)}, \sum_{j=1}^{n} x^{(j)}\rangle \tag{11}$$

Notations:

1. Generally $\tilde{O}(.)$ and $\tilde{\Omega}(.)$ hides factor $\log(\text{poly}(m, n, p, \delta, \frac{1}{\epsilon}))$. At Lemma E.3 and D.4, $\tilde{\Omega}(.)$ hides factor $\log(\text{poly}(n))$.
2. Generally with high probability (abbreviated as w.h.p.) implies with probability $1 - \dfrac{1}{\text{poly}(m, n, p, \delta, \frac{1}{\epsilon})}$, where poly$(.)$ implies a sufficiently large polynomial. At Lemma E.2, E.3 and D.4 "w.h.p." implies $1 - \dfrac{1}{\text{poly}(n)}$.
3. We denote, $\sigma = \frac{1}{\sqrt{m}}$ such that the expert initialization, $w_{r,s}^{(0)} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{d\times d}), \forall s \in [k], \forall r \in [m/k]$.

The training algorithms for separate-training and joint-training pMoE are given in Algorithm 1 and Algorithm 2, respectively:

Algorithm 1 Two-phase SGD for separate-training pMoE

Input : Training data $\{(x_i, y_i)\}_{i=1}^N$, learning rates η_r and η, number of iterations T_r and T, batch-sizes B_r and B

Step-1: Initialize $w_s^{(0)}, w_{r,s}^{(0)}, a_{r,s}, \forall s \in \{1, 2\}, r \in [m/k]$ according to (7) and (8)

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

$$w_s^{(t+1)} = w_s^{(t)} - \eta_r \frac{\partial \ell_r(w_1^{(t)}, w_2^{(t)})}{\partial w_s^{(t)}}, \forall s \in \{1, 2\}$$

Step-3: for $t = 0, 1, ..., T - 1$ do:

$$w_{r,s}^{(t+1)} = w_{r,s}^{(t)} - \eta \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \quad \forall r \in [m/k], s \in \{1, 2\}$$

Algorithm 2 SGD for joint-training pMoE

Input : Training data $\{(x_i, y_i)\}_{i=1}^N$, learning rate η, number of iteration T, batch-size B

Step-1: Initialize $w_s^{(0)}, w_{r,s}^{(0)}, a_{r,s}, \forall s \in [k], r \in [m/k]$ according to (7) and (8)

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

$$w_s^{(t+1)} = w_s^{(t)} - \eta \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_s^{(t)}}, \forall s \in [k]$$

$$w_{r,s}^{(t+1)} = w_{r,s}^{(t)} - \eta \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \quad \forall r \in [m/k], s \in [k]$$

C. Proof Sketch

The proof of generalization guarantee for pMoE (i.e., Theorem 4.2 and 4.5) can be outlined as follows (the proof for single CNN follows a simpler version of the outline provided below):

Step 1. (**Feature learning in the router**) For separate-training pMoE, we first show that the batch-gradient of the router loss (i.e., $\ell_r(w_1^{(t)}, w_2^{(t)})$) w.r.t. the gating kernels (i.e., $w_1^{(t)}$ and $w_2^{(t)}$) has large component (of size $\frac{1-\delta_d}{2} - \Omega\left(\frac{n}{\sqrt{B_r}}\right)$) along the class-discriminative pattern o_1 and o_2 respectively. Then, by selecting $B_r = \Omega\left(\frac{n^2}{(1-\delta_d)^2}\right)$ (which provides us $\Omega(1)$ loss reduction per step) and training for $\Omega\left(\frac{1}{1-\delta_d}\right)$ iterations, we can show that w_1 and w_2 is sufficiently aligned with o_1 and o_2

respectively to guarantee the selection of these class-discriminative patterns in TOP-l patches when $l \geq l^*$ (see Lemma D.4 for exact statement).

Step 2. (**Coupling the experts to pseudo experts**) When the experts of pMoE are sufficiently overparameterized, w.h.p. the experts can be coupled to a smooth pseudo network[7] of experts as for every sample drawn from the distribution $\mathcal{D}$ and every $\tau > 0$, the activation pattern for $1 - \Omega\left(\frac{\tau l}{\sigma}\right)$ (for separate-training pMoE) or $1 - \Omega\left(\frac{\tau n}{\sigma}\right)$ (for joint-training pMoE) fraction of hidden nodes in each expert does not change from the initialization for $O(\frac{\tau}{\eta})$ iterations (see Lemma G.1 or H.1 for exact statement). This indicates that with $\tau = O\left(\frac{\sigma}{l}\right)$ (for separate-training pMoE) or $\tau = O\left(\frac{\sigma}{n}\right)$ (for joint-training pMoE), $\eta = \Omega\left(\frac{1}{ml}\right)$ (for separate-training pMoE) or $\eta = \Omega\left(\frac{1}{mn}\right)$ (for joint-training pMoE) and $\sigma = O\left(\frac{1}{\sqrt{m}}\right)$ we can couple $\Omega(1)$ fraction of hidden nodes of each expert to the corresponding pseudo experts for $O(\sqrt{m})$ iterations.

Step 3.(**Large error implies large gradient**) We can now analyze the pseudo network of experts corresponding to the separate-training pMoE to show that, at any iteration t, the magnitude of the expected gradient for any expert $s \in \{1, 2\}$ of the pseudo network is $\Omega\left(\frac{v_s^{(t)}}{l}\right)$ where $v_s^{(t)}$ characterizes the class-conditional expected error over samples with $y = +1$ and $y = -1$ for $s = 1$ and $s = 2$, respectively (see Lemma G.3 for exact statement). Similarly, for joint-training pMoE we show that the magnitude of the expected gradient is $\Omega\left(\frac{v_s^{(t)}}{l}\right)$, but this time $v_s^{(t)}$ characterizes the maximum of the class-conditional expected-errors over the samples for which the expert "s" receiving class-discriminative patterns from the router (see Lemma H.3 for exact statement).

Step 4. (**Convergence**) Now let us define $v^{(t)} = \sqrt{\sum_{s\in[k]} v_s^2(t)}$. For separate-training pMoE, by selecting the batch size $B_t = \Omega(\frac{l^4}{(v^{(t)})^4})$ at iteration t, $\eta = \Omega(\frac{(v^{(t)})^2}{ml^2})$ and $\tau = O(\frac{\sigma(v^{(t)})^2}{l^3})$, we can couple the empirical batch gradient of each expert of the true network for that batch to the expected gradient of the corresponding expert of the pseudo network. Because the pseudo network is smooth, we can show that SGD minimizes the expected loss of the true network by $\Omega(\frac{\eta m(v^{(t)})^2}{l^2})$ at each iteration for $t = O(\frac{\sigma(v^{(t)})^2}{\eta l^3})$ iterations (see Lemma G.4 for the exact statement). Similarly, for joint-training pMoE, by selecting $B_t = \Omega(\frac{k^2}{(v^{(t)})^4})$ and $\eta = \Omega(\frac{(v^{(t)})^2 l^3}{mk^2})$ we can show that SGD minimizes the expected loss of the true network by $\Omega(\frac{\eta m(v^{(t)})^2}{l^2})$ for $t = O(\frac{\sigma(v^{(t)})^2 l^2}{\eta nk})$ (see Lemma H.4 for exact statement). As the loss of the true network is $O(1)$ at initialization, eventually the network will converge.

Step 5. (**Generalization**) We show that to ensure at most ϵ generalization error after any iteration t, we need $\max\{v_1^{(t)}, v_2^{(t)}\} < \epsilon^2$ where $v_1^{(t)}$ and $v_2^{(t)}$ correspond to the class-conditional expected error of the class with $y = +1$ and $y = -1$, respectively. Now as we show that the router in the separate-training pMoE dispatch class-discriminative patches of all the samples labeled as $y = +1$ to the expert indexed by $s = 1$ and class-discriminative patches of all the samples labeled as $y = -1$ to the expert indexed by $s = 2$ from the beginning of expert-training, $v^{(t)} < \epsilon^2$ ensures $\max\{v_1^{(t)}, v_2^{(t)}\} < \epsilon^2$. On the other hand, for the joint-training pMoE, as we assume that the router ensures the dispatchment of all the class-discriminative patches of a class to a particular expert before the convergence of the model and the gating value of the patch is the largest among all the patches sent to that particular expert, $v^{(t)} < \frac{\epsilon^2}{l}$ implies $\max\{v_1^{(t)}, v_2^{(t)}\} < \epsilon^2$. Hence for separate-training pMoE, by setting $v^{(t)} \geq \epsilon^2$ we show that with $B = \Omega(l^4/\epsilon^8)$ and $\eta = \Omega(1/m\text{poly}(l, 1/\epsilon))$ for $T = O(l^4/\epsilon^8)$ iterations, we can guarantee that the generalization error is less than ϵ (see Theorem F.3 for exact statement). Similarly, for joint-training pMoE, by setting $v^{(t)} \geq \frac{\epsilon^2}{l}$ and setting $B = \Omega(k^2 l^4/\epsilon^8)$ and $\eta = \Omega(1/(m\text{poly}(l, 1/\epsilon)$ for $T = O(k^2 l^2/\epsilon^8)$ iterations, we can guarantee that the generalization error is less than ϵ (see Theorem F.5 for exact statement).

D. Proof of the Lemma 4.1

Definition D.1. (δ'-closer class-irrelevant patterns) For any $\delta' > 0$, a class-irrelevant pattern q is δ'-closer to o_1 than o_2, if $\langle o_1, q\rangle - \langle o_2, q\rangle > \delta'$ for any $\delta' > 0$. Similarly, a class-irrelevant pattern q is δ'-closer to o_2 than o_1 if $\langle o_2, q\rangle - \langle o_1, q\rangle > \delta'$.

Definition D.2. (Set of δ'-closer class-irrelevant patterns, $\mathcal{S}_c(\delta')$) For any $\delta' > 0$, define the set of δ'-closer class-irrelevant patterns, denoted as $\mathcal{S}_c(\delta') \subset \bigcup_{i=1}^{p} S_j$ such that: $\forall q \in \mathcal{S}_c(\delta'), |\langle o_1 - o_2, q\rangle| > \delta'$.

[7]The pseudo network is defined as the network which activation pattern does not change from the initialization i.e., the sign of the pre-activation output of hidden nodes does not change from the sign at initialization; see (Li & Liang, 2018) for details.

Definition D.3. (Threshold, l^*) Define the threshold l^* such that:
$\forall(x,y)\sim\mathcal{D}, \left|\{j\in[n]: x^{(j)}\neq o_1 \text{ and } x^{(j)}\in S_c\left(\frac{1-\delta_d}{2}\right)\}\right|\leq l^*-1$

Lemma D.4. *(Full version of **Lemma 4.1**) For every $l\geq l^*$, w.h.p. over the random initialization defined in (7), after completing the Step-2 of Algorithm-1 with batch-size $B_r=\tilde{\Omega}\left(\frac{n^2}{(1-\delta_d)^2}\right)$ and learning rate $\eta_r=\Theta\left(\frac{1}{n}\right)$ for $T_r=\Omega\left(\frac{1}{1-\delta_d}\right)$ iterations, the returned $w_1^{(T_r)}$ and $w_2^{(T_r)}$ satisfy*

$$\underset{j\in[n]}{\arg}\,(x^{(j)}=o_1)\in J_1(w_1^{(T_r)},x),\quad \forall(x,y=+1)\sim\mathcal{D}$$

$$\underset{j\in[n]}{\arg}\,(x^{(j)}=o_2)\in J_2(w_2^{(T_r)},x),\quad \forall(x,y=-1)\sim\mathcal{D}$$

Proof. The proof follows directly from the Definition D.3 and the Lemma E.3. □

E. Lemmas Used to Prove the Lemma 4.1

We denote,
$\nabla_{w_s^{(t)}}\mathbb{E}[\ell_r(w_1,w_2)]:=\mathbb{E}_{\mathcal{D}}\left[\frac{\partial\ell_r(w_1^{(t)},w_2^{(t)})}{\partial w_s^{(t)}}\right]$ where $w_s^{(t)}\in\{w_1^{(t)},w_2^{(t)}\}$ for all $t\in[T_r]$.

Lemma E.1. *At any iteration $t\leq T_r$ of the Step-2 of Algorithm 1,*

$$\nabla_{w_1^{(t)}}\mathbb{E}[l(f(x),y)]=-\frac{1}{2}(o_1-o_2),\text{ and }\nabla_{w_2^{(t)}}\mathbb{E}[l(f(x),y)]=-\frac{1}{2}(o_2-o_1)$$

Proof. As, $\ell_r(w_1^{(t)},w_2^{(t)})=-\frac{1}{B_r}\sum_{(x,y)\in\mathcal{B}_t^r}y\langle w_1^{(t)}-w_2^{(t)},\sum_{j=1}^n x^{(j)}\rangle$,
$\nabla_{w_1^{(t)}}\mathbb{E}[l_r(w_1,w_2)]=-\mathbb{E}_{\mathcal{D}}\left[y\sum_{j=1}^n x^{(j)}\right]$ and $\nabla_{w_2^{(t)}}\mathbb{E}[l_r(w_1,w_2)]=\mathbb{E}_{\mathcal{D}}\left[y\sum_{j=1}^n x^{(j)}\right]$
Therefore,

$$\begin{aligned}
\nabla_{w_1^{(t)}}\mathbb{E}[l_r(w_1,w_2)] &= -\frac{1}{2}\mathbb{E}_{\mathcal{D}|y=+1}\left[\sum_{j=1}^n x^{(j)}|y=+1\right]+\frac{1}{2}\mathbb{E}_{\mathcal{D}|y=-1}\left[\sum_{j=1}^n x^{(j)}|y=-1\right]\\
&= -\frac{1}{2}\mathbb{E}_{\mathcal{D}|y=+1}\left[\sum_{j\in[n]/\underset{j}{\arg}\,x^{(j)}=o_1} x^{(j)}|y=+1\right]+\frac{1}{2}\mathbb{E}_{\mathcal{D}|y=-1}\left[\sum_{j\in[n]/\underset{j}{\arg}\,x^{(j)}=o_2} x^{(j)}|y=-1\right]\\
&\quad -\frac{1}{2}(o_1-o_2)\\
&= -\frac{1}{2}(o_1-o_2)
\end{aligned}$$

where the last equality comes from the fact that class-irrelevant patterns are distributed identically in both classes. Using similar line of arguments we can show that, $\nabla_{w_2^{(t)}}\mathbb{E}[l(f(x),y)]=-\frac{1}{2}(o_2-o_1)$. □

Lemma E.2. *With probability $1-\frac{1}{poly(n)}$ (i.e., w.h.p.) over the random initialization of the gating kernels defined in (7), $\left|\left|w_s^{(0)}\right|\right|\leq\frac{1}{n^2}$; $\forall s\in\{1,2\}$*

Proof. Let us denote the i-th element of the vector $w_s^{(0)}$ as $w_{s_i}^{(0)}$ where $i \in [d]$.
Then according to the random initialization of $w_s^{(0)}$ and using a Gaussian tail-bound (i.e., for $X \sim \mathcal{N}(0,\sigma^2) : Pr[|X| \geq t] \leq 2e^{-t^2/2\sigma^2}$): $\mathbb{P}\left[\left|w_{s_i}^{(0)}\right| \geq \frac{1}{n^2\sqrt{d}}\right] \leq \frac{1}{poly(n)}$.
Let us denote the event $\mathcal{E} : \forall i \in [d], \left|w_{s_i}^{(0)}\right| \leq \frac{1}{n^2\sqrt{d}}$. Therefore, $\mathbb{P}[\mathcal{E}] \geq 1 - \frac{1}{poly(n)}$.
Now, conditioned on the event $\mathcal{E}$, $\left|\left|w_s^{(0)}\right|\right| \leq \frac{1}{n^2}$.
Therefore, $\mathbb{P}\left[\left|\left|w_s^{(0)}\right|\right| \leq \frac{1}{n^2}\right] \leq \mathbb{P}\left[\left|\left|w_s^{(0)}\right|\right| \geq \frac{1}{n^2}|\mathcal{E}\right]\mathbb{P}[\mathcal{E}] = 1 - \frac{1}{poly(n)}$ □

Lemma E.3. *W.h.p. over the random initialization of the gating-kernels defined in (7) and randomly selected batch of batch-size* $B_r = \tilde{\Omega}\left(\frac{n^2}{(1-\delta_d)^2}\right)$ *at each iteration, after* $T_r = \Omega\left(\frac{1}{1-\delta_d}\right)$ *iterations of Step-2 of Algorithm 1 with learning rate* $\eta_r = \Theta\left(\frac{1}{n}\right)$, $\forall (x,y) \sim \mathcal{D}, j \in [n] : x^{(j)} \notin \mathcal{S}_c(\frac{1-\delta_d}{2})$, $\langle w_1^{(T_r)}, o_1\rangle > \langle w_1^{(T_r)}, x^{(j)}\rangle$ *and* $\langle w_2^{(T_r)}, o_2\rangle > \langle w_2^{(T_r)}, x^{(j)}\rangle$.

Proof. Let, at t-th iteration of Step-2 of Algorithm 1, $\tilde{\nabla}_{w_s}^{(t)} = \dfrac{\partial \ell_r(w_1^{(t)}, w_2^{(t)})}{\partial w_s^{(t)}}$ for all $s \in \{1,2\}$
Also let us denote, $\nabla_{w_s^{(t)}}\mathbb{E}\left[\ell_r(w_1^{(t)}, w_2^{(t)})\right] = \nabla_{w_s}^{(t)}$ for all $s \in \{1,2\}$

Therefore, after T_r-th iteration of SGD and using Lemma E.1,

$$\begin{aligned} w_1^{(T_r)} &= w_1^{(0)} - \eta_r \sum_{t=0}^{T_r-1} \tilde{\nabla}_{w_1}^{(t)} \\ &= w_1^{(0)} + \frac{\eta_r T_r}{2}(o_1 - o_2) - \eta_r \sum_{t=0}^{T_r-1}\left(\tilde{\nabla}_{w_1}^{(t)} - \nabla_{w_1}^{(t)}\right) \end{aligned}$$

Similarly, $w_2^{(T_r)} = w_2^{(0)} + \frac{\eta_r T_r}{2}(o_2 - o_1) - \eta_r \sum_{t=0}^{T_r-1}\left(\tilde{\nabla}_{w_2}^{(t)} - \nabla_{w_2}^{(t)}\right)$.

Now, $||\tilde{\nabla}_{w_s}^{(t)}|| = O(n)$. Hence, w.h.p. over a randomly sampled batch of size B_r, using Hoeffding's concentration,

$||\tilde{\nabla}_{w_s}^{(t)} - \nabla_{w_s}^{(t)}|| = \tilde{O}\left(\frac{n}{\sqrt{B_r}}\right); \forall s \in \{1,2\}$.

Now,

$$\begin{aligned} \langle w_1^{(T_r)}, o_1\rangle &= \langle w_1^{(0)}, o_1\rangle + \frac{\eta_r T_r}{2}(1 - \langle o_1, o_2\rangle) - \eta_r \sum_{t=0}^{T_r-1}\langle \tilde{\nabla}_{w_1}^{(t)} - \nabla_{w_1}^{(t)}, o_1\rangle \\ &\geq \frac{\eta_r T_r}{2}(1-\delta_d) - \eta_r T_r \tilde{O}\left(\frac{n}{\sqrt{B_r}}\right) - \left|\left|w_1^{(0)}\right|\right| \end{aligned}$$

On the other hand, $\forall (x,y) \sim \mathcal{D}, \forall j \in [n] : x^{(j)} \notin \mathcal{S}_c\left(\frac{1-\delta_d}{2}\right)$,

$$\begin{aligned} \langle w_1^{(T_r)}, x^{(j)}\rangle &= \langle w_1^{(0)}, x^{(j)}\rangle + \frac{\eta_r T_r}{2}\left(\langle o_1, x^{(j)}\rangle - \langle o_2, x^{(j)}\rangle\right) - \eta_r \sum_{t=0}^{T_r-1}\langle \tilde{\nabla}_{w_1}^{(t)} - \nabla_{w_1}, x^{(j)}\rangle \\ &\leq \frac{\eta_r T_r}{4}(1-\delta_d) + \eta_r T_r \tilde{O}\left(\frac{n}{\sqrt{B_r}}\right) + \left|\left|w_1^{(0)}\right|\right| \end{aligned}$$

From Lemma E.2, w.h.p. over the random initialization: $\left|\left|w_1^{(0)}\right|\right| \leq \dfrac{1}{n^2}$.

Therefore, selecting $B_r = \tilde{\Omega}\left(\frac{n^2}{(1-\delta_d)^2}\right)$ and $\eta_r = \Theta\left(\frac{1}{n}\right)$, we need $T_r = \Omega\left(\frac{1}{1-\delta_d}\right)$ iterations to achieve $\langle w_1^{(T_r)}, o_1\rangle > \langle w_1^{(T_r)}, x^{(j)}\rangle, \forall j \in [n] : x^{(j)} \in \mathcal{S}_c\left(\frac{1-\delta_d}{2}\right)$

Similar line of arguments can be made to show with batch size $B_r = \tilde{\Omega}\left(\frac{n^2}{(1-\delta_d)^2}\right)$ and learning rate $\eta_r = \Theta\left(\frac{1}{n}\right)$, after $T_r = \Omega\left(\frac{1}{1-\delta_d}\right)$ iterations, $\langle w_2^{(T_r)}, o_2\rangle \geq \langle w_2^{(T_r)}, x^{(j)}\rangle, \forall j \in [n] : x^{(j)} \in \mathcal{S}_c\left(\frac{1-\delta_d}{2}\right)$.

□

F. Proofs of the Theorem 4.2, 4.3 and 4.5

Definition F.1. At any iteration t of the minibatch SGD,

1. Define the **value function**, $v^{(t)}(\theta^{(t)}, x, y) := \frac{1}{1+e^{yf_M(\theta^{(t)},x)}}$. It is easy to show that for any $(x, y) \sim \mathcal{D}$, $0 \leq v^{(t)}(\theta^{(t)}, x, y) \leq 1$. The function captures the prediction error, i.e., a larger $v^{(t)}$ indicates a larger prediction error.
2. Define, the **class-conditional expected value function**, $v_1^{(t)} := \mathbb{E}_{\mathcal{D}|y=+1}[v^{(t)}(\theta^{(t)}, x, y)|y = +1]$ and $v_2^{(t)} := \mathbb{E}_{\mathcal{D}|y=-1}[v^{(t)}(\theta^{(t)}, x, y)|y = -1]$. Here, $v_1^{(t)}$ captures the expected error for the class with label $y = +1$ and $v_2^{(t)}$ captures the expected error for the class with label $y = -1$.

Definition F.2. At any iteration t of the minibatch SGD,

1. For any sample $(x, y) \sim \mathcal{D}$, we define the **reduction of loss** at the t-th iteration of SGD as,
$$\Delta L(\theta^{(t)}, \theta^{(t+1)}, x, y) := \mathcal{L}(\theta^{(t)}, x, y) - \mathcal{L}(\theta^{(t+1)}, x, y)$$
where, $\mathcal{L}(\theta^{(t)}, x, y) := \log(1 + e^{-yf_M(\theta^{(t)},x)})$ is the **single-sample loss function**.
2. Define the **expected reduction of loss** at the t-th iteration of SGD as,
$$\Delta L(\theta^{(t)}, \theta^{(t+1)}) := \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(\theta^{(t)}, x, y) - \mathcal{L}(\theta^{(t+1)}, x, y)\right]$$

Theorem F.3. *(Full version of **Theorem 4.2**) For every $\epsilon > 0$ and $l \geq l^*$, for every $m \geq M_S = \tilde{\Omega}\left(l^{10}p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_S = \tilde{\Omega}(l^8p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \tilde{\Omega}\left(l^4p^6\delta^3/\epsilon^8\right)$ and the learning rate $\eta = \tilde{O}\big(1/m\text{poly}(l, p, \delta, 1/\epsilon, \log m)\big)$ for $T = \tilde{O}\left(l^4p^6\delta^3/\epsilon^8\right)$ iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[yf_M(\theta^{(T)}, x) > 0\right] \geq 1 - \epsilon$$

Proof. First we will show that for any $\epsilon < \frac{1}{2}$, if $\mathbb{P}_{(x,y)\sim\mathcal{D}}\left[yf(\theta^{(t)}, x) > 0\right] \leq 1 - \epsilon$, then $\max\{v_1^{(t)}, v_2^{(t)}\} \geq \epsilon^2$.

Now for any $(x, y) \sim \mathcal{D}$ and $\epsilon < \frac{1}{2}$, if $v^{(t)}(\theta^{(t)}, x, y) \leq \epsilon$, $yf_M(\theta^{(t)}, x, y) > 0$ i.e., the prediction is correct.

Now if $v_1^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y)\middle|y = +1\right] \leq \epsilon^2$, then using Markov's inequality $\mathbb{P}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \leq \epsilon\right] \geq 1 - \epsilon$ which implies for any $\epsilon < \frac{1}{2}$, $\mathbb{P}_{\mathcal{D}|y=+1}\left[yf(\theta^{(t)}, x) > 0\right] \geq 1 - \epsilon$.

Similarly, if $v_2^{(t)} = \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\middle|y = -1\right] \leq \epsilon^2$, for any $\epsilon < \frac{1}{2}$, $\mathbb{P}_{\mathcal{D}|y=-1}\left[yf(\theta^{(t)}, x) > 0\right] \geq 1 - \epsilon$.

Therefore, for any $\epsilon < \frac{1}{2}$, if $\mathbb{P}_{(x,y)\sim\mathcal{D}}\left[yf(\theta^{(t)}, x) > 0\right] \leq 1 - \epsilon$, then $\max\{v_1^{(t)}, v_2^{(t)}\} \geq \epsilon^2$.

Now, if $v^{(t)} := \sqrt{\sum_{s\in\{1,2\}}(v_s^{(t)})^2} \leq \epsilon^2$ then $\max\{v_1^{(t)}, v_2^{(t)}\} \leq \epsilon^2$, which implies after a proper number of iterations if $v^{(t)} \leq \epsilon^2$ then $\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[yf_M(\theta^{(T)}, x) > 0\right] \geq 1-\epsilon$.

Let, $v^{(t)} \geq \epsilon^2$. Then by using Lemma G.4 for every $l \geq l^*$, with $\eta = \tilde{O}\left(\frac{\epsilon^4}{ml^2p^3\delta^{3/2}}\right)$ and $B = \tilde{\Omega}\left(\frac{l^4p^6\delta^3}{\epsilon^8}\right)$, at least for $t = \tilde{O}\left(\frac{\sigma\epsilon^4}{\eta l^3p^3\delta^{3/2}}\right)$ we have,

$$\Delta L(\theta^{(t)}, \theta^{(t+1)}) = \tilde{\Omega}\left(\frac{\eta m\epsilon^4}{l^2p^3\delta^{3/2}}\right) \tag{12}$$

Now, as $w_{r,s}^{(0)} \sim \mathcal{N}(0,\sigma^2)$ with $\sigma = \frac{1}{\sqrt{m}}$, $\left\langle w_{r,s}^{(0)}, x^{(j)}\right\rangle \sim \mathcal{N}(0,\sigma^2)$ $\forall j \in J_s(w_s^{(0)}, x)$ and $\forall (x,y)\sim\mathcal{D}$. Therefore, w.h.p. $\left|f_M(\theta^{(0)}, x)\right| = \tilde{O}(1)$ which implies $\mathcal{L}(\theta^{(0)}, x, y) = \tilde{O}(1)$. Now as $\mathcal{L}(\theta^{(t)}, x, y) > 0$, (12) can happen at most $\tilde{O}\left(\frac{l^2p^3\delta^{3/2}}{\eta m\epsilon^4}\right)$ iterations. Now as $\eta m = \tilde{O}\left(\frac{\epsilon^4}{l^2p^3\delta^{3/2}}\right)$, we need $T = \tilde{O}\left(\frac{l^4p^6\delta^3}{\epsilon^8}\right)$ iterations to ensure that $v^{(t)} \leq \epsilon^2$.

On the other hand, to ensure (12) hold for T iterations, we need,

$$\frac{\sigma\epsilon^4}{\eta l^3p^3\delta^{3/2}} = \tilde{\Omega}\left(\frac{l^2p^3\delta^{3/2}}{\eta m\epsilon^4}\right)$$

which implies we need $m = \tilde{\Omega}\left(\frac{l^{10}p^{12}\delta^6}{\epsilon^{16}}\right)$. □

Now, for any $(x, y=+1)\sim\mathcal{D}$ and $(x, y=-1)\sim\mathcal{D}$, let us denote the index of the class-discriminative patterns i.e., o_1 and o_2 as j_{o_1} and j_{o_2}, respectively.

Definition F.4. At any iteration t of **minibatch SGD of the joint-training pMoE** (i.e., Step-2 of Algorithm 2),

1. For any $(x, y=+1)\sim\mathcal{D}$ and the expert $s\in[k]$, define the **event that** o_1 **in Top-**l as, $\mathcal{E}_{1,s}^{(t)} : j_{o_1} \in J_s(w_s^{(t)}, x)$. Similarly, for any $(x, y=-1)\sim\mathcal{D}$ define the **event that** o_2 **in Top-**l as, $\mathcal{E}_{2,s}^{(t)} : j_{o_2} \in J_s(w_s^{(t)}, x)$.

2. For any expert $s\in[k]$, define the **probability of the event that** o_1 **in Top-**l as, $p_{1,s}^{(t)} := \mathbb{P}_{\mathcal{D}|y=+1}\left[\mathcal{E}_{1,s}^{(t)}|y=+1\right]$ and the **probability of the event that** o_2 **in Top-**l as, $p_{2,s}^{(t)} := \mathbb{P}_{\mathcal{D}|y=-1}\left[\mathcal{E}_{2,s}^{(t)}|y=-1\right]$

3. For any expert $s\in[k]$ define, $v_{1,s}^{(t)} := \mathbb{E}_{\mathcal{D}|y=+1,\mathcal{E}_{1,s}^{(t)}}\left[p_{1,s}^{(t)}G_{j_{o_1},s}^{(t)}(x)v^{(t)}(\theta^{(t)}, x, y)\big|y=+1, \mathcal{E}_{1,s}^{(t)}\right]$ and $v_{2,s}^{(t)} := \mathbb{E}_{\mathcal{D}|y=-1,\mathcal{E}_{2,s}^{(t)}}\left[p_{2,s}^{(t)}G_{j_{o_2},s}^{(t)}(x)v^{(t)}(\theta^{(t)}, x, y)\big|y=-1, \mathcal{E}_{2,s}^{(t)}\right]$ where $G_{j_{o_1},s}^{(t)}(x)$ and $G_{j_{o_2},s}^{(t)}(x)$ denote the gating value for the class-discriminative patterns o_1 and o_2 conditioned on $\mathcal{E}_{1,s}^{(t)}$ and $\mathcal{E}_{2,s}^{(t)}$, respectively.

Theorem F.5. *(Full version of the **Theorem 4.5**) Suppose Assumption 4.4 hold. Then for every $\epsilon > 0$, for every $m \geq M_J = \tilde{\Omega}\left(k^3n^2l^6p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_J = \tilde{\Omega}(k^4l^6p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \tilde{\Omega}\left(k^2l^4p^6\delta^3/\epsilon^8\right)$ and the learning rate $\eta = \tilde{O}\big(1/mpoly(l,p,\delta,1/\epsilon,\log m)\big)$ for $T = \tilde{O}\left(k^2l^2p^6\delta^3/\epsilon^8\right)$ iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[yf_M(\theta^{(T)}, x) > 0\right] \geq 1-\epsilon$$

Proof. From the argument of the proof of Theorem F.3, we know that for any $\epsilon < \frac{1}{2}$, if $\mathbb{P}_{(x,y)\sim\mathcal{D}}\left[yf(\theta^{(t)}, x) > 0\right] \leq 1-\epsilon$, then $\max\{v_1^{(t)}, v_2^{(t)}\} \geq \epsilon^2$ where $v_1^{(t)} := \mathbb{E}_{\mathcal{D}|y=+1}[v^{(t)}(\theta^{(t)}, x, y)|y=+1]$ and $v_2^{(t)} := \mathbb{E}_{\mathcal{D}|y=-1}[v^{(t)}(\theta^{(t)}, x, y)|y=-1]$

Now, we will consider the case when $t \geq T'$ where T' is defined in Assumption 4.4.

Now, if the expert $s_1 \in [k]$ satisfies Assumption 4.4 for $y = +1$, then $p_{1,s_1}^{(t)} = 1$ and $G_{j_{o_1},s_1}^{(t)}(x) \geq \frac{1}{l}$ for any $(x, y = +1) \sim \mathcal{D}$. Therefore, $v_{1,s_1}^{(t)} \geq \frac{v_1^{(t)}}{l}$.

Similarly, if the expert $s_2 \in [k]$ satisfies Assumption 4.4 for $y = -1$, then $v_{2,s_2}^{(t)} \geq \frac{v_2^{(t)}}{l}$.

Now for any expert $s \in [k]$, let us define $v_s^{(t)} := \max\{v_{1,s}^{(t)}, v_{2,s}^{(t)}\}$

Now, if $v^{(t)} := \sqrt{\sum_{s \in [k]} (v_s^{(t)})^2} \leq \frac{\epsilon^2}{l}$, then $v_{s_1}^{(t)} \leq \frac{\epsilon^2}{l}$ and $v_{s_2}^{(t)} \leq \frac{\epsilon^2}{l}$.

This implies, $\max\{v_{1,s_1}^{(t)}, v_{2,s_1}^{(t)}\} \leq \frac{\epsilon^2}{l}$ and $\max\{v_{1,s_2}^{(t)}, v_{2,s_2}^{(t)}\} \leq \frac{\epsilon^2}{l}$.

Therefore, $v_{1,s_1}^{(t)} \leq \frac{\epsilon^2}{l}$ and $v_{2,s_2}^{(t)} \leq \frac{\epsilon^2}{l}$ which implies $v_1^{(t)} \leq \epsilon^2$ and $v_2^{(t)} \leq \epsilon^2$.

In that case, $\max\{v_1^{(t)}, v_2^{(t)}\} \leq \epsilon^2$.

Therefore, by taking $v^{(t)} \geq \frac{\epsilon^2}{l}$, using the results of Lemma H.4 and following same procedure as in Theorem F.3 we can complete the proof. ☐

Theorem F.6. *(Full version of the **Theorem 4.3**) For every $\epsilon > 0$, for every $m \geq M_C = \tilde{\Omega}\left(n^{10}p^{12}\delta^6/\epsilon^{16}\right)$ with at least $N_C = \tilde{\Omega}(n^8 p^{12}\delta^6/\epsilon^{16})$ training samples, after performing minibatch SGD with the batch size $B = \tilde{\Omega}\left(n^4p^6\delta^3/\epsilon^8\right)$ and the learning rate $\eta = \tilde{O}\big(1/mpoly(n, p, \delta, 1/\epsilon, \log m)\big)$ for $T = \tilde{O}\left(n^4p^6\delta^3/\epsilon^8\right)$ iterations, it holds w.h.p. that*

$$\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}\left[yf_C(\theta^{(T)}, x) > 0\right] \geq 1 - \epsilon$$

Proof. From the argument of the proof of Theorem F.3, we know that for any $\epsilon < \frac{1}{2}$, if $\mathbb{P}_{(x,y)\sim\mathcal{D}}\left[yf(\theta^{(t)}, x) > 0\right] \leq 1 - \epsilon$, then $v^{(t)} := \max\{v_1^{(t)}, v_2^{(t)}\} \geq \epsilon^2$ where $v_1^{(t)} := \mathbb{E}_{\mathcal{D}|y=+1}[v^{(t)}(\theta^{(t)}, x, y)|y = +1]$ and $v_2^{(t)} := \mathbb{E}_{\mathcal{D}|y=-1}[v^{(t)}(\theta^{(t)}, x, y)|y = -1]$.

Therefore, taking $v^{(t)} \geq \epsilon^2$, using the results of Lemma I.3 and following similar procedure as in Theorem F.3 we can complete the proof. ☐

G. Lemmas Used to Prove the Theorem 4.2

For any iteration t of the Step-3 of Algorithm 1, recall the loss function for a single-sample generated by the distribution $\mathcal{D}$, $\mathcal{L}(\theta^{(t)}, x, y) := \log(1 + e^{-yf_M(\theta^{(t)}, x)})$. The gradient of the loss for a single sample with respect to the hidden nodes of the experts:

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = -ya_{r,s}v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l}\sum_{j \in J_s(w_s^{(t)}, x)} x^{(j)} 1_{\langle w_{r,s}^{(t)}, x^{(j)}\rangle \geq 0}\right) \tag{13}$$

We define the corresponding ***pseudo-gradient*** as:

$$\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = -ya_{r,s}v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l}\sum_{j \in J_s(w_s^{(t)}, x)} x^{(j)} 1_{\langle w_{r,s}^{(0)}, x^{(j)}\rangle \geq 0}\right) \tag{14}$$

Therefore, the expected pseudo-gradient:

$$\begin{aligned}\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} &= \mathbb{E}_{\mathcal{D}}\left[\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}}\right] \\ &= -\frac{a_{r,s}}{2}\Bigg(\mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)\Big|y=+1\right] \\ &\qquad -\mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(0)},P_jx\rangle\geq 0}\right)\Big|y=-1\right]\Bigg) \\ &= -\frac{a_{r,s}}{2}P_{r,s}^{(t)}\end{aligned}$$

Here,

$$\begin{aligned}P_{r,s}^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}&\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)\Big|y=+1\right] \\ &- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)\Big|y=-1\right]\end{aligned}$$

Lemma G.1. *W.h.p. over the random initialization of the hidden nodes of the experts defined in 8, for every $(x,y)\sim\mathcal{D}$ and for every $\tau>0$, for every $t=\tilde{O}\left(\frac{\tau}{\eta}\right)$ of the Step-3 of Algorithm 1, we have that for at least $\left(1-\frac{2e\tau l}{\sigma}\right)$ fraction of $r\in[m/2]$ of the expert $s\in\{1,2\}$:*

$$\frac{\partial\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}} = \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}} \text{ and } |\langle w_{r,s}^{(t)},x^{(j)}\rangle|\geq\tau, \forall j\in J_s(w_s^{(t)},x)$$

Proof. Recall the gradient of the loss for single-sample $(x,y)\sim\mathcal{D}$ w.r.t. the hidden node $r\in[m/2]$ of the expert $s\in\{1,2\}$:

$$\frac{\partial\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}} = -ya_{r,s}v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(t)},x^{(j)}\rangle\geq 0}\right)$$

and the corresponding pseudo-gradient:

$$\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}} = -ya_{r,s}v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)$$

Now, $a_{r,s}\sim\mathcal{N}(0,1)$. Hence, using the concentration bound of Gaussian random variable (i.e., for $X\sim\mathcal{N}(0,\sigma^2)$: $Pr[|X|\leq t]\geq 1-2e^{-t^2/2\sigma^2}$) and as $\tilde{O}(.)$ hides factor $\log\left(poly(m,n,p,\delta,\frac{1}{\epsilon})\right)$ we get:

$$\mathbb{P}\left[|a_{r,s}|=\tilde{O}(1)\right]\geq 1-\frac{1}{poly(m,n,p,\delta,\frac{1}{\epsilon})} \text{ (i.e., w.h.p.)}$$

Now as $v^{(t)}(\theta^{(t)},x,y)\leq 1$ and $||x^{(j)}||=1$, w.h.p. $\left|\left|\frac{\partial\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}}\right|\right|=\tilde{O}(1)$ so as the mini-batch gradient, $\left|\left|\frac{\partial\mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right|=\tilde{O}(1)$.

Now, from the update rule of the Step-3 of Algorithm 1, $w_{r,s}^{(t)} - w_{r,s}^{(t+1)} = \eta \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}$

Therefore, using the property of Telescoping series, $w_{r,s}^{(0)} - w_{r,s}^{(t)} = \eta \sum_{i=1}^{t} \frac{\partial \mathcal{L}(\theta^{(i-1)})}{\partial w_{r,s}^{(t)}}$

Therefore, $\left|\left|w_{r,s}^{(t)} - w_{r,s}^{(0)}\right|\right| = \Upsilon \eta t$ where we denote $\tilde{O}(1)$ by Υ

Now, for every $\tau > 0$, consider the set $\mathcal{H}_s := \left\{ r \in [m/2] : \forall j \in J_s(w_s^{(t)}, x), |\langle w_{r,s}^{(0)}, x^{(j)} \rangle| \geq 2\tau \right\}$

Now, for every $t \leq \dfrac{\tau}{\Upsilon \eta}$, $|\langle w_{r,s}^{(t)} - w_{r,s}^{(0)}, x^{(j)} \rangle| \leq \tau$

Which implies for every $r \in \mathcal{H}_s$, $t \leq \dfrac{\tau}{\Upsilon \eta}$ and $j \in J_s(w_s^{(t)}, x)$, $|\langle w_{r,s}^{(t)}, x^{(j)} \rangle| \geq \tau$

Therefore, for every $r \in \mathcal{H}_s$, $t = \tilde{O}(\frac{\tau}{\eta})$ and $j \in J_s(w_s^{(t)}, x)$, $1_{\langle w_{r,s}^{(t)}, x^{(j)} \rangle \geq 0} = 1_{\langle w_{r,s}^{(0)}, x^{(j)} \rangle \geq 0}$ and hence, $\dfrac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = \dfrac{\widetilde{\partial} \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}$

Now, we will find the lower bound of $|\mathcal{H}_s|$:

As, $w_{r,s}^{(0)} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{d \times d}), \forall j \in J_s(w_s^{(t)}, x), \langle w_{r,s}^{(0)}, x^{(j)} \rangle \sim \mathcal{N}(0, \sigma^2)$

Hence, $\mathbb{P}[|\langle w_{r,s}^{(0)}, x^{(j)} \rangle| \leq 2\tau] \leq \dfrac{2e\tau}{\sigma}$

Now as $|J_s(w_s^{(t)}, x)| = l$, $\mathbb{P}[\forall j \in J_s(w_s^{(t)}, x), |\langle w_{r,s}^{(0)}, x^{(j)} \rangle| \geq 2\tau] \geq 1 - \dfrac{2e\tau l}{\sigma}$

Therefore, $|\mathcal{H}_s| \geq \left(1 - \dfrac{2e\tau l}{\sigma}\right) \frac{m}{2}$

□

Using the following two lemmas we show that when $v_1^{(t)}$ is large, the expected pseudo-gradient of the loss function w.r.t. the hidden nodes of the *expert 1* is large. Similar thing happens for *expert 2* when $v_2^{(t)}$ is large. We prove the first of these two lemmas for a fixed set $\{v^{(t)}(\theta^{(t)}, x, y) : (x, y) \sim \mathcal{D}\}$ which does not depend on the random initialization of the hidden nodes of the experts (i.e., on $\{w_{r,s}^{(0)}\}$). In the second of these two lemmas we remove the dependency on fixed set by means of a sampling trick introduced in (Li & Liang, 2018) to take a union bound over an epsilon-net on the set $\{v^{(t)}(\theta^{(t)}, x, y) : (x, y) \sim \mathcal{D}\}$.

Lemma G.2. *For any possible fixed set $\{v^{(t)}(\theta^{(t)}, x, y) : (x, y) \sim \mathcal{D}\}$ (that does not depend on $w_{r,s}^{(0)}$) such that $v_s^{(t)} = v_1^{(t)}$ for $s = 1$ and $v_s^{(t)} = v_2^{(t)}$ for $s = 2$ we have for every $l \geq l^*$:*

$$\mathbb{P}\left[||P_{r,s}^{(t)}|| = \widetilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}} \right) \right] = \Omega\left(\frac{1}{p\sqrt{\delta}} \right)$$

Proof. WLOG, let's assume $s = 1$. Now,

$$P_{r,1}^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} x^{(j)} 1_{\langle w_{r,1}^{(0)}, x^{(j)} \rangle \geq 0} \right) \Big| y = +1 \right]$$
$$- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} x^{(j)} 1_{\langle w_{r,1}^{(0)}, x^{(j)} \rangle \geq 0} \right) \Big| y = -1 \right]$$

Then,

$$
\begin{aligned}
h(w_{r,1}^{(0)}) &:= \langle P_{r,1}, w_{r,1}^{(0)} \rangle \\
&= \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\langle w_{r,1}^{(0)}, x^{(j)} \rangle \right) \right) \Big| y = +1 \right] \\
&\quad - \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\langle w_{r,1}^{(0)}, x^{(j)} \rangle \right) \right) \Big| y = -1 \right]
\end{aligned}
$$

Now, let us decompose $w_{r,1}^{(0)} = \alpha o_1 + \beta$, where $\beta \perp o_1$

Then,

$$
\begin{aligned}
h(w_{r,1}^{(0)}) &= \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\alpha \langle o_1, x^{(j)} \rangle + \langle \beta, x^{(j)} \rangle \right) \right) \Big| y = +1 \right] \\
&\quad - \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\alpha \langle o_1, x^{(j)} \rangle + \langle \beta, x^{(j)} \rangle \right) \right) \Big| y = -1 \right] \\
&= \phi(\alpha) - l(\alpha)
\end{aligned}
$$

Where,

$$
\phi(\alpha) := \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\alpha \langle o_1, x^{(j)} \rangle + \langle \beta, x^{(j)} \rangle \right) \right) \Big| y = +1 \right]
$$

and,

$$
l(\alpha) := \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(w_1^{(t)}, x)} \textbf{ReLU}\left(\alpha \langle o_1, x^{(j)} \rangle + \langle \beta, x^{(j)} \rangle \right) \right) \Big| y = -1 \right]
$$

Note that, $\phi(\alpha)$ and $l(\alpha)$ both are convex functions.

Now for $l \geq l^*$, using Lemma D.4, we can express $\phi(\alpha)$ as follows:

$$
\begin{aligned}
\phi(\alpha) &= \frac{v_1^{(t)}}{l} \textbf{ReLU}(\alpha) \\
&\quad + \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_1(x) / \underset{j \in J_1(x)}{\arg} x^{(j)} = o_1} \textbf{ReLU}\left(\alpha \langle o_1, x^{(j)} \rangle + \langle \beta, x^{(j)} \rangle \right) \right) \Big| y = +1 \right]
\end{aligned}
$$

Now, for any class-irrelevant pattern set S_i where $i \in [p]$, let us define $q_i^* \in S_i$ such that $q_i^* = \frac{\mathbb{E}_{S_i}[q]}{||\mathbb{E}_{S_i}[q]||}$. Also, let us define the set, $\mathcal{H} := \{q_i^* : i \in [p]\} \cup \{o_2\}$

Now let us define the event $\mathcal{E}_\tau : (i)\ |\alpha| \leq \tau;\ (ii)\ \forall q' \in \mathcal{H} : |\langle \beta, q' \rangle| \geq 4\tau$

Now, as $\alpha \sim \mathcal{N}(0, \sigma^2)$, for every $q' \in \mathcal{H}$, $\langle \beta, q' \rangle \sim \mathcal{N}\left(0, \left(1 - \langle o_1, q' \rangle^2\right)\sigma^2\right)$

Now, $1 - \langle o_1, q' \rangle^2 \geq \frac{1}{\delta}$. Hence, $\mathbb{P}[\exists q' \in \mathcal{H} : |\langle \beta, q' \rangle| \leq 4\tau] \leq \dfrac{4e\tau p\sqrt{\delta}}{\sigma}$

Therefore, $\mathbb{P}[\forall q' \in \mathcal{H} : |\langle \beta, q' \rangle| \geq 4\tau] \geq 1 - \dfrac{4e\tau p\sqrt{\delta}}{\sigma}$.

Picking, $\tau \leq \dfrac{\sigma}{8ep\sqrt{\delta}}$ gives, $\mathbb{P}[\forall q' \in \mathcal{H} : |\langle \beta, q' \rangle| \geq 4\tau] \geq \dfrac{1}{2}$.

On the other hand, $\mathbb{P}[|\alpha| \leq \tau] = \Omega\left(\dfrac{\tau}{\sigma}\right)$. Therefore, $\mathbb{P}[\mathcal{E}_\tau] = \Omega\left(\dfrac{\tau}{\sigma}\right)$

Now, $\forall i \in [p]$ s.t. $q \in S_i, \mathbb{E}[|\langle w_{r,1}^{(0)}, q - q_i^* \rangle|] \leq \mathbb{E}_{\mathcal{N}(0,\sigma^2 \mathbb{I}_{d\times d})}[||w_{r,1}^{(0)}||]\mathbb{E}_{S_i}[||q - q_i^*||] \leq \tau$, where the last inequality comes from the bound of the diameter of the pattern sets and the fact that for any $X \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{d\times d}), \mathbb{E}[||X||] \leq 4\sigma\sqrt{d}$.

Therefore, using Markov's inequality $\forall i \in [p]$ s.t. $q \in S_i, \mathbb{P}[|\langle w_{r,1}^{(0)}, q - q_i^* \rangle| \leq 2\tau] \geq \frac{1}{2}$

Now,
$\forall i \in [p]$, s.t. $q \in S_i, \textbf{ReLU}\left(\alpha\langle o_1, q \rangle + \langle \beta, q \rangle\right) = \textbf{ReLU}\left(\alpha\langle o_1, q_i^* \rangle + \langle \beta, q_i^* \rangle + \langle w_{r,1}^{(0)}, q - q_i^* \rangle\right)$

Now, conditioned on the event $\mathcal{E}_\tau$, for a fixed β and α is the only random variable,
$\forall i \in [p]$ s.t. $q \in S_i, \textbf{ReLU}\left(\alpha\langle o_1, q \rangle + \langle \beta, q \rangle\right) = \left(\alpha\langle o_1, q \rangle + \langle \beta, q \rangle\right) 1_{\langle \beta, q_i^* \rangle \geq 0}$ which is a linear function of $\alpha \in [-\tau, \tau]$ with probability at least $\frac{1}{2}$ and, $\textbf{ReLU}\left(\alpha\langle o_1, o_2 \rangle + \langle \beta, o_2 \rangle\right) = \left(\alpha\langle o_1, o_2 \rangle + \langle \beta, o_2 \rangle\right) 1_{\langle \beta, o_2 \rangle \geq 0}$ which is a linear function of $\alpha \in [-\tau, \tau]$ with probability 1.

Now, let us define $\{\partial l(\alpha)\}$ and $\{\partial \phi(\alpha)\}$ as the set of sub-gradient at the point α for $l(\alpha)$ and $\phi(\alpha)$ respectively such that $\partial_{\max} l(\alpha) = \max\{\partial l(\alpha)\}$, $\partial_{\max}\phi(\alpha) = \max\{\partial\phi(\alpha)\}$, $\partial_{\min} l(\alpha) = \min\{\partial l(\alpha)\}$ and $\partial_{\min}\phi(\alpha) = \min\{\partial\phi(\alpha)\}$.

Then, using the above argument, conditioned on the event $\mathcal{E}_\tau$, $\partial_{\max} l(\tau) - \partial_{\min} l(-\tau) = 0$.

On the other hand, $\partial_{\max}\phi(\tau/2) - \partial_{\min}\phi(-\tau/2) = \dfrac{v_1^{(t)}}{l}$.

Now using Lemma J.1, conditioned on the event $\mathcal{E}_\tau$, $\underset{\alpha \sim U(-\tau,\tau)}{\mathbb{P}}\left[|\phi(\alpha) - l(\alpha)| \geq \dfrac{v_1^{(t)}\tau}{512l}\right] \geq \dfrac{1}{64}$.

Now, for $\tau \leq \dfrac{\sigma}{8ep\sqrt{\delta}}$, conditioned on $\mathcal{E}_\tau$, the density $p(\alpha) \in \left[\dfrac{1}{e\tau}, \dfrac{e}{\tau}\right]$, which implies that,

$$\mathbb{P}\left[h(w_{r,1}^{(0)}) \geq \frac{v_1^{(t)}\tau}{128l}\right] \geq \mathbb{P}\left[h(w_{r,1}^{(0)}) \geq \frac{v_1^{(t)}\tau}{128l} | \mathcal{E}_\tau\right] \mathbb{P}\left[\mathcal{E}_\tau\right] = \Omega\left(\frac{\tau}{\sigma}\right) \tag{15}$$

Now, as $v_1^{(t)}$ does not depends on $w_{r,1}^{(0)}$, $\langle P_{r,1}^{(t)}, w_{r,1}^{(0)} \rangle \sim \mathcal{N}(0, \sigma^2 ||P_{r,1}^{(t)}||^2)$.

Now, using a concentration bound of Gaussian RV (i.e., $\mathbb{P}[X \geq \sigma x] \leq e^{-x^2/2}$),

$$\mathbb{P}[\langle P_{r,1}^{(t)}, w_{r,1}^{(0)} \rangle \geq (\sigma||P_{r,1}^{(t)}||)10c] \leq e^{-50c^2};\text{ here } c > 10. \tag{16}$$

Now, taking $c = 100\sqrt{\log \dfrac{p\sqrt{\delta}}{\sigma}}$ in (16) we get,

$$\mathbb{P}[\langle P_{r,1}^{(t)}, w_{r,1}^{(0)} \rangle = \tilde{\Omega}(\sigma||P_{r,1}^{(t)}||)] = o(1) \tag{17}$$

On the other hand, picking $\tau = \Theta(\frac{\sigma}{p\sqrt{\delta}})$ and plugging in at (15) gives,

$$\mathbb{P}\left[\langle P_{r,1}^{(t)}, w_{r,1}^{(0)}\rangle = \Omega\left(\sigma\frac{v_1^{(t)}}{lp\sqrt{\delta}}\right)\right] = \Omega\left(\frac{1}{p\sqrt{\delta}}\right) \tag{18}$$

Comparing (17) and (18) we get, $\mathbb{P}\left[||P_{r,1}^{(t)}|| = \tilde{\Omega}\left(\frac{v_1^{(t)}}{lp\sqrt{\delta}}\right)\right] = \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$

□

Lemma G.3. *Let $v_s^{(t)} = v_1^{(t)}$ for $s = 1$ and $v_s^{(t)} = v_2^{(t)}$ for $s = 2$. Then, for every $v_s^{(t)} > 0$, for $m = \tilde{\Omega}\left(\frac{l^2p^3\delta^{3/2}}{(v_s^{(t)})^2}\right)$, for every possible set $\{v^{(t)}(\theta^{(t)}, x, y) : (x, y) \sim \mathcal{D}\}$ (that depends on $w_{r,s}^{(0)}$), there exist at least $\Omega\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/2]$ of the expert $s \in \{1, 2\}$ such that for every $l \geq l^*$,*

$$\left|\left|\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Proof. Let us pick S samples to form $\mathbf{S} = \{(x_i, y_i)\}_{i=1}^S$ with $S/2$ many samples from $y = +1$ and $S/2$ many samples from $y = -1$. Let us denote the subset of samples with $y = +1$ as $\mathbf{S}_1$ and the subset of samples with $y = -1$ as $\mathbf{S}_2$. Therefore, $|\mathbf{S}_1| = |\mathbf{S}_2| = S/2$. Let us denote the corresponding value function of i-th sample of $\mathbf{S}$ as $v^{(t)}(\theta^{(t)}, x_i, y_i)$. Since, each $v^{(t)}(\theta^{(t)}, x_i, y_i) \in [0, 1]$ using Hoeffding's inequality we know that w.h.p. :

$$\left|v_s^{(t)} - \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_s} v^{(t)}(\theta^{(t)}, x_i, y_i)\right| = \tilde{O}\left(\frac{1}{\sqrt{S}}\right)$$

This implies that, as long as $S = \tilde{\Omega}\left(\frac{1}{(v_s^{(t)})^2}\right)$, we will have that,

$$\frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_s} v^{(t)}(\theta^{(t)}, x_i, y_i) \in \left[\frac{1}{2}v_s^{(t)}, \frac{3}{2}v_s^{(t)}\right]$$

Now, the average pseudo-gradient over the set $\mathbf{S}$,

$$\begin{aligned}\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x_i, y_i)}{\partial w_{r,s}^{(t)}} &= \frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} - ya_{r,s}v^{(t)}(\theta^{(t)}, x_i, y_i)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x_i)} x_i^{(j)}1_{\langle w_{r,s}^{(0)}, x_i^{(j)}\rangle\geq 0}\right)\\ &= -\frac{a_{r,s}}{2}P_{r,s}^{(t)}(\mathbf{S})\end{aligned}$$

where,

$$\begin{aligned}P_{r,s}^{(t)}(\mathbf{S}) = \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_1} v^{(t)}(\theta^{(t)}, x_i, y_i)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x_i)} x_i^{(j)}1_{\langle w_{r,s}^{(0)}, x_i^{(j)}\rangle\geq 0}\right)\\ - \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_2} v^{(t)}(\theta^{(t)}, x_i, y_i)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x_i)} x_i^{(j)}1_{\langle w_{r,s}^{(0)}, x_i^{(j)}\rangle\geq 0}\right)\end{aligned}$$

Now as $a_{r,s} \sim \mathcal{N}(0,1)$, $\mathbb{P}\left[\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}}\right|\right| = \left|\left|\frac{a_{r,s}}{2}P_{r,s}^{(t)}(\mathbf{S})\right|\right| \geq \frac{1}{2}\left|\left|P_{r,s}^{(t)}(\mathbf{S})\right|\right|\right] \geq \frac{1}{e}$

Now for a fixed set $\{v^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$ as long as $S = \tilde{\Omega}\left(\frac{1}{v_s^2(t)}\right)$, for every $l \geq l^*$ using Lemma G.2,

$$\mathbb{P}\left[||P_{r,s}^{(t)}(\mathbf{S})|| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)\right] = \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$$

Hence, for a fixed set $\{v^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$, the probability that there are less than $O\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/2]$ such that $\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}}\right|\right|$ is $\tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$ is no more than p_{fix} where, $p_{\text{fix}} \leq \exp\left(-\Omega\left(\frac{m}{p\sqrt{\delta}}\right)\right)$.

Moreover, for every $\bar{\varepsilon} > 0$, for two different $\{v^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$, $\{v'^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$ such that $\forall(x_i,y_i) \in \mathbf{S}$, $|v^{(t)}(\theta^{(t)},x_i,y_i) - v'^{(t)}(\theta^{(t)},x_i,y_i)| \leq \bar{\varepsilon}$, since w.h.p. $|a_{r,s}| = \tilde{O}(1)$,

$$\begin{aligned}&\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} - ya_{r,s}(v^{(t)}(\theta^{(t)},x_i,y_i) - v'^{(t)}(\theta^{(t)},x_i,y_i))\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x_i)} x_i^{(j)}1_{\langle w_{r,s}^{(0)},x_i^{(j)}\rangle\geq 0}\right)\right|\right| \\ &= \tilde{O}(\bar{\varepsilon})\end{aligned}$$

which implies that we can take $\bar{\varepsilon}$-net with $\bar{\varepsilon} = \tilde{\Theta}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$.

Thus, the probability that there exists $\{v^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$ such that there are no more than $O\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/2]$ with $\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$ is no more than, $p \leq p_{\text{fix}}\left(\frac{v_s^{(t)}}{\bar{\varepsilon}}\right)^S \leq \exp\left(-\Omega\left(\frac{m}{p\sqrt{\delta}}\right) + S\log\left(\frac{v_s^{(t)}}{\bar{\varepsilon}}\right)\right)$.

Hence, for $m = \widetilde{\Omega}\left(Sp\sqrt{\delta}\right)$ with $S = \tilde{\Omega}\left(\frac{1}{v_s^2(t)}\right)$, w.h.p. for every possible choice of $\{v^{(t)}(\theta^{(t)},x_i,y_i) : (x_i,y_i) \in \mathbf{S}\}$, there are at least $\Omega\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/2]$ such that,

$$\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now, we consider the difference between the sample gradient and the expected gradient. Since, $\left|\left|\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}}\right|\right| = \widetilde{O}(1)$, by using the Hoeffding's inequality, we know that for every $r \in [m/2]$:

$$\left|\left|\frac{1}{S}\sum_{(x_i,y_i)\in S}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)},x_i,y_i)}{\partial w_{r,s}^{(t)}} - \frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{O}\left(\frac{1}{\sqrt{S}}\right)$$

This implies that as long as $S = \tilde{\Omega}\left(\left(\frac{lp\sqrt{\delta}}{v_s^{(t)}}\right)^2\right)$ and hence for $m = \tilde{\Omega}\left(\frac{l^2p^3\delta^{3/2}}{(v_s^{(t)})^2}\right)$, such $r \in [m/2]$ also have:

$$\left\|\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

□

Lemma G.4. *Let us define* $v^{(t)} := \sqrt{\sum_{s\in\{1,2\}}(v_s^{(t)})^2}$ *where* $v_s^{(t)} = v_1^{(t)}$ *for* $s = 1$ *and* $v_s^{(t)} = v_2^{(t)}$ *for* $s = 2$*;* $\gamma := \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$*. Then, by selecting learning rate* $\eta = \tilde{O}\left(\frac{\gamma^3(v^{(t)})^2}{ml^2}\right)$ *and batch size* $B = \tilde{\Omega}\left(\frac{l^4}{\gamma^6(v^{(t)})^4}\right)$*, at each iteration* t *of the Step-3 of Algorithm 1 such that* $t = \tilde{O}\left(\frac{\sigma\gamma^3(v^{(t)})^2}{\eta l^3}\right)$*, w.h.p. we can ensure that for every* $l \geq l^*$*,*

$$\Delta L(\theta^{(t)}, \theta^{(t+1)}) \geq \frac{\eta m\gamma^3}{l^2}\tilde{\Omega}\left((v^{(t)})^2\right)$$

Proof. For every $l \geq l^*$, from Lemma G.3, for at least γ fraction of $r \in [m/2]$ of expert s:

$$\left\|\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now w.h.p., $\left\|\frac{\tilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}\right\| = \widetilde{O}(1)$. Therefore, w.h.p. over a randomly sampled batch from $\mathcal{D}$ at iteration t denoted as $\mathcal{B}_t$ of size B:

$$\left\|\frac{1}{B}\sum_{(x,y)\in\mathcal{B}_t}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} - \frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\| = \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$$

This implies, by selecting batch-size of $B = \Omega\left(\frac{l^2p^2\delta}{(v_s^{(t)})^2}\right)$, for these γ fraction of $r \in [m/2]$ of expert s we can ensure that:

$$\left\|\frac{1}{B}\sum_{(x_i,y_i)\in\mathcal{B}_t}\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now using Lemma G.1, for a fixed $(x, y) \in \mathcal{B}_t$, by selecting $\tau = \frac{\sigma\gamma}{4elB}$ we have $\left(1 - \frac{\gamma}{2B}\right)$ fraction of $r \in [m/2]$ of the expert s:

$$\frac{\partial\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}$$

Therefore, at least $(1 - \gamma/2)$ fraction of $r \in [m/2]$ of the expert s:

$$\frac{\partial\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} \qquad \forall(x, y) \in \mathcal{B}_t$$

Recall our definition of loss-function for SGD at iteration t with mini-batch $\mathcal{B}_t$, $\mathcal{L}(\theta^{(t)}) = \frac{1}{B}\sum_{(x,y)\in\mathcal{B}_t} \log\left(1+e^{-yf_M(\theta^{(t)},x)}\right) = \frac{1}{B}\sum_{(x,y)\in\mathcal{B}_t} \mathcal{L}(\theta^{(t)},x,y)$ and the corresponding batch-gradient at iteration t, $\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} = \frac{1}{B}\sum_{(x,y)\in\mathcal{B}_t} \frac{\partial \mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}}$. Therefore, there are at least $\gamma/2$ fraction of $r \in [m/2]$ of the expert s:

$$\left|\left|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now for any $(x',y') \sim \mathcal{D}$, according to Lemma G.1, w.h.p. there are at least $1-\frac{2e\tau l}{\sigma}$ fraction of $r \in [m/2]$ of the expert s such that $\forall j \in J_s(w_s^{(t)},x'), |\langle w_{r,s}^{(t)}, x'^{(j)}\rangle| \geq \tau$. Let us denote the set of these r's of s as $\mathcal{S}_{r,s}$. Therefore, on the set $\bigcup_{s\in\{1,2\}} \mathcal{S}_{r,s}$, the loss function $\mathcal{L}(\theta^{(t)},x',y')$ is $\widetilde{O}(1)$ -smooth and $\widetilde{O}(1)$ -Lipschitz smooth.

On the other hand, the update rule of SGD at the iteration t is, $\theta^{(t+1)} = \theta^{(t)} - \eta\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}$

Therefore, using Lemma J.2,

$$\begin{aligned}
&\Delta L(\theta^{(t)},\theta^{(t+1)},x',y') := \mathcal{L}(\theta^{(t)},x',y') - \mathcal{L}(\theta^{(t+1)},x',y')\\
&\geq \eta \sum_{r\in \bigcup\limits_{s\in[2]}\mathcal{S}_{r,s}} \left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \mathcal{L}(\theta^{(t)},x',y')}{\partial w_{r,s}^{(t)}}\right\rangle - \sum_{s\in[2], r\in[m/2]\setminus \bigcup\limits_{s\in[2]}\mathcal{S}_{r,s}} \widetilde{O}(\eta) - \widetilde{O}\left(\eta^2 m^2\right)\\
&\geq \eta \sum_{r\in[m/2],s\in[2]} \left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \mathcal{L}(\theta^{(t)},x',y')}{\partial w_{r,s}^{(t)}}\right\rangle - \widetilde{O}\left(\frac{\eta\tau l m}{\sigma}\right) - \widetilde{O}\left(\eta^2 m^2\right)
\end{aligned}$$

Let us denote the event,

$$\begin{aligned}
&\mathcal{E}_0:\\
&\quad \Delta L(\theta^{(t)},\theta^{(t+1)},x',y')\\
&\quad \geq \eta \sum_{r\in[m/2],s\in[2]} \left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \mathcal{L}(\theta^{(t)},x',y')}{\partial w_{r,s}^{(t)}}\right\rangle - \widetilde{O}\left(\frac{\eta\tau l m}{\sigma}\right) - \widetilde{O}\left(\eta^2 m^2\right)
\end{aligned}$$

Then, $\mathbb{P}\left[\mathcal{E}_0\right] \geq 1 - \frac{1}{poly(m,n,p,\delta,\frac{1}{\epsilon})}$ (i.e., w.h.p.) and hence $\mathbb{P}\left[\neg\mathcal{E}_0\right] \leq \frac{1}{poly(m,n,p,\delta,\frac{1}{\epsilon})}$

Also, let us define the event,

$$\begin{aligned}
&\mathcal{E}_1:\\
&\quad \left|\mathcal{L}(\theta^{(t)},x',y')\right| = \tilde{O}(m), \left|\left|\frac{\partial \mathcal{L}(\theta^{(t)},x',y')}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{O}(1) \text{ and } \left|\left|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{O}(1)
\end{aligned}$$

Then, $\mathbb{P}\left[\mathcal{E}_1\right] \geq 1 - \frac{1}{poly(m,n,p,\delta,\frac{1}{\epsilon})}$ and hence $\mathbb{P}\left[\neg\mathcal{E}_1\right] \leq \frac{1}{poly(m,n,p,\delta,\frac{1}{\epsilon})}$

Now, the expected gradient at iteration t, $\frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} := \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)},x',y')}{\partial w_{r,s}^{(t)}}\right]$

Therefore condition on $\mathcal{E}_1$,

$$\begin{aligned}\frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} &= \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)}, x', y')}{\partial w_{r,s}^{(t)}}|\mathcal{E}_1\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)}, x', y')}{\partial w_{r,s}^{(t)}}|\mathcal{E}_0, \mathcal{E}_1\right]\mathbb{P}\left[\mathcal{E}_0|\mathcal{E}_1\right] + \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)}, x', y')}{\partial w_{r,s}^{(t)}}|\neg\mathcal{E}_0, \mathcal{E}_1\right]\mathbb{P}\left[\neg\mathcal{E}_0|\mathcal{E}_1\right]\end{aligned}$$

Which implies,

$$\left\|\frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} - \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)}, x', y')}{\partial w_{r,s}^{(t)}}|\mathcal{E}_0, \mathcal{E}_1\right]\right\| \leq \frac{\tilde{O}(1)}{poly(m, n, p, \delta, \frac{1}{\epsilon})}$$

Again, condition on $\mathcal{E}_1$,

$$\begin{aligned}&\Delta L(\theta^{(t)}, \theta^{(t+1)}) := \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(\theta^{(t)}, x', y') - \mathcal{L}(\theta^{(t+1)}, x', y')|\mathcal{E}_1\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(\theta^{(t)}, x', y') - \mathcal{L}(\theta^{(t+1)}, x', y')|\mathcal{E}_0, \mathcal{E}_1\right]\mathbb{P}\left[\mathcal{E}_0|\mathcal{E}_1\right] \\ &\quad + \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(\theta^{(t)}, x', y') - \mathcal{L}(\theta^{(t+1)}, x', y')|\neg\mathcal{E}_0, \mathcal{E}_1\right]\mathbb{P}\left[\neg\mathcal{E}_0|\mathcal{E}_1\right] \\ &\geq \eta \sum_{r\in[m/2], s\in[2]}\left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \mathbb{E}_{\mathcal{D}}\left[\frac{\partial \mathcal{L}(\theta^{(t)}, x', y')}{\partial w_{r,s}^{(t)}}|\mathcal{E}_0, \mathcal{E}_1\right]\right\rangle - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right) \\ &\quad - \frac{\tilde{O}(m)}{poly(m, n, p, \delta, \frac{1}{\epsilon})} \\ &\geq \eta \sum_{r\in[m/2], s\in[2]}\left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\rangle - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right) - \frac{\tilde{O}(m)}{poly(m, n, p, \delta, \frac{1}{\epsilon})} \\ &\quad - \frac{\tilde{O}(\eta m)}{poly(m, n, p, \delta, \frac{1}{\epsilon})} \\ &\geq \eta \sum_{r\in[m/2], s\in[2]}\left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\rangle - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right)\end{aligned}$$

Now, w.h.p.

$$\left\|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} - \frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\| = \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$$

Therefore,

$$\left\langle \frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}, \frac{\partial \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\rangle \geq \left\|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\|^2 - \tilde{O}\left(\frac{1}{\sqrt{B}}\right)$$

Therefore,

$$\begin{aligned}\Delta L(\theta^{(t)}, \theta^{(t+1)}) &\geq \eta \sum_{r\in[m], s\in[2]}\left\|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\|^2 - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right) - \eta\tilde{O}\left(\frac{m}{\sqrt{B}}\right) \\ &\geq \frac{\eta m\gamma^3}{l^2}\tilde{\Omega}\left(\sum_{s\in[2]}(v_s^{(t)})^2\right) - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right) - \eta\tilde{O}\left(\frac{m}{\sqrt{B}}\right) \\ &\geq \frac{\eta m\gamma^3}{l^2}\tilde{\Omega}\left((v^{(t)})^2\right) - \tilde{O}\left(\frac{\eta\tau lm}{\sigma}\right) - \tilde{O}\left(\eta^2 m^2\right) - \eta\tilde{O}\left(\frac{m}{\sqrt{B}}\right)\end{aligned}$$

Now selecting, $\eta = \tilde{O}\left(\frac{\gamma^3 (v^{(t)})^2}{ml^2}\right)$, $B = \tilde{\Omega}\left(\frac{l^4}{\gamma^6 (v^{(t)})^4}\right)$, $\tau = \tilde{O}\left(\frac{\sigma\gamma^3 (v^{(t)})^2}{l^3}\right)$ and hence for $t = \tilde{O}\left(\frac{\sigma\gamma^3 (v^{(t)})^2}{\eta l^3}\right)$, we get,

$$\Delta L(\theta^{(t)}, \theta^{(t+1)}) \geq \frac{\eta m \gamma^3}{l^2} \tilde{\Omega}\left((v^{(t)})^2\right)$$

□

H. Lemmas Used to Prove the Theorem 4.5

In joint-training pMoE i.e., for any iteration t of the Step-2 of Algorithm 2, the gradient of the loss for single-sample with respect to the hidden nodes of the experts:

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = -y a_{r,s} v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(t)}, x^{(j)} \rangle \geq 0}\right) \tag{19}$$

and the corresponding ***pseudo-gradient***:

$$\frac{\widetilde{\partial} \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = -y a_{r,s} v^{(t)}(\theta^{(t)}, x, y) \left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(0)}, x^{(j)} \rangle \geq 0}\right) \tag{20}$$

and the expected pseudo-gradient:

$$\begin{aligned}
\frac{\widetilde{\partial} \hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}} &= \mathbb{E}_{\mathcal{D}}\left[\frac{\widetilde{\partial} \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}\right] \\
&= -\frac{a_{r,s}}{2}\left(\mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(0)}, x^{(j)} \rangle \geq 0}\right) \Big| y = +1\right]\right. \\
&\qquad \left. -\mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(0)}, P_j x \rangle \geq 0}\right) \Big| y = -1\right]\right) \\
&= -\frac{a_{r,s}}{2} P_{r,s}^{(t)}
\end{aligned}$$

with,

$$\begin{aligned}
P_{r,s}^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}&\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(0)}, x^{(j)} \rangle \geq 0}\right) \Big| y = +1\right] \\
&- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{l} \sum_{j \in J_s(w_s^{(t)}, x)} G_{j,s}(w_s^{(t)}, x) x^{(j)} 1_{\langle w_{r,s}^{(0)}, x^{(j)} \rangle \geq 0}\right) \Big| y = -1\right]
\end{aligned}$$

Lemma H.1. *W.h.p. over the random initialization of the hidden nodes of the experts defined in (8), for every $(x, y) \sim \mathcal{D}$ and for every $\tau > 0$, for every $t = \tilde{O}\left(\frac{\tau l}{\eta}\right)$ of the Step-2 of Algorithm 2, we have that for at least $\left(1 - \frac{2e\tau n}{\sigma}\right)$ fraction of $r \in [m/k]$ of the expert $s \in [k]$:*

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} = \frac{\widetilde{\partial} \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}} \text{ and } |\langle w_{r,s}^{(t)}, x^{(j)} \rangle| \geq \tau, \forall j \in [n]$$

Proof. Using similar argument as in Lemma G.1 and as $\sum_{j\in J_s(w_s^{(t)},x)} G_{j,s}(w_s^{(t)},x)=1$ w.h.p. $\left|\left|\frac{\partial\mathcal{L}(\theta^{(t)},x,y)}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{O}\left(\frac{1}{l}\right)$ so as the mini-batch gradient, $\left|\left|\frac{\partial\mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{O}\left(\frac{1}{l}\right)$.
Therefore, $\left|\left|w_{r,s}^{(t)} - w_{r,s}^{(0)}\right|\right| = \tilde{O}\left(\frac{\eta t}{l}\right)$.

Now, for every $\tau > 0$, considering the set $\mathcal{H}_s := \left\{r\in[m/k] : \forall j\in[n], |\langle w_{r,s}^{(0)}, x^{(j)}\rangle| \geq 2\tau\right\}$ and following the same procedure as in Lemma G.1 we can complete the proof.

□

Lemma H.2. *For the expert $s\in[k]$ and any possible fixed set $\{v^{(t)}(\theta^{(t)},x,y)G_{j,s}(w_s^{(t)},x) : (x,y)\sim\mathcal{D}, j\in J_s(w_s^{(t)},x)\}$ (that does not depend on $w_{r,s}^{(0)}$) such that $v_s^{(t)} = v_{1,s}^{(t)} = \max\{v_{1,s}^{(t)}, v_{2,s}^{(t)}\}$, we have:*

$$\mathbb{P}\left[||P_{r,s}^{(t)}|| = \widetilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)\right] = \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$$

Proof. We know that,

$$\begin{aligned}P_{r,s}^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}&\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} G_{j,s}(w_s^{(t)},x)x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)\Big|y=+1\right]\\&- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} G_{j,s}(w_s^{(t)},x)x^{(j)}1_{\langle w_{r,s}^{(0)},x^{(j)}\rangle\geq 0}\right)\Big|y=-1\right]\end{aligned}$$

Therefore,

$$\begin{aligned}&h(w_{r,s}^{(0)}) := \langle P_{r,s}, w_{r,s}^{(0)}\rangle\\&= \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} G_{j,s}(w_s^{(t)},x)\textbf{ReLU}\left(\langle w_{r,s}^{(0)},x^{(j)}\rangle\right)\right)\Big|y=+1\right]\\&\quad- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(w_s^{(t)},x)} G_{j,s}(w_s^{(t)},x)\textbf{ReLU}\left(\langle w_{r,s}^{(0)},x^{(j)}\rangle\right)\right)\Big|y=-1\right]\end{aligned}$$

Now, decomposing $w_{r,s}^{(0)} = \alpha o_1 + \beta$ with $\beta\perp o_1$ we get,

$$\begin{aligned}h(w_{r,s}^{(0)}) &= \frac{v_{1,s}^{(t)}}{l}\textbf{ReLU}(\alpha)\\&+ \mathbb{E}_{\mathcal{D}|y=+1,\mathcal{E}_{1,s}^{(t)}}\left[p_{1,s}^{(t)}v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)/j_{o_1}} G_{j,s}\textbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle + \langle\beta,x^{(j)}\rangle\right)\right)\right]\\&+ \mathbb{E}_{\mathcal{D}|y=+1,\neg\mathcal{E}_{1,s}^{(t)}}\left[(1-p_{1,s}^{(t)})v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)} G_{j,s}\textbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle + \langle\beta,x^{(j)}\rangle\right)\right)\right]\\&- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)} G_{j,s}\textbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle + \langle\beta,x^{(j)}\rangle\right)\right)\right]\\&= \phi(\alpha) - l(\alpha)\end{aligned}$$

where,

$$\begin{aligned}\phi(\alpha) := &\frac{v_{1,s}^{(t)}}{l}\mathbf{ReLU}\left(\alpha\right)\\ &+ \mathbb{E}_{\mathcal{D}|y=+1,\mathcal{E}_{1,s}^{(t)}}\left[p_{1,s}^{(t)}v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)/j_{o_1}}G_{j,s}\mathbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle+\langle\beta,x^{(j)}\rangle\right)\right)\right]\\ &+ \mathbb{E}_{\mathcal{D}|y=+1,\neg\mathcal{E}_{1,s}^{(t)}}\left[(1-p_{1,s}^{(t)})v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)}G_{j,s}\mathbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle+\langle\beta,x^{(j)}\rangle\right)\right)\right]\end{aligned}$$

and

$$l(\alpha) := \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{l}\sum_{j\in J_s(x)}G_{j,s}\mathbf{ReLU}\left(\alpha\langle o_1,x^{(j)}\rangle+\langle\beta,x^{(j)}\rangle\right)\right)\right]$$

Now as $\phi(\alpha)$ and $l(\alpha)$ both are convex functions, using the same procedure as in Lemma G.1 we can complete the proof. □

Lemma H.3. *Let $v_s^{(t)} = \max\{v_{1,s}^{(t)}, v_{2,s}^{(t)}\}$. Then, for every $v_s^{(t)} > 0$, for $m = \tilde{\Omega}\left(\frac{klp^3\delta^{3/2}}{(v_s^{(t)})^2}\right)$, for every possible set $\{v^{(t)}(\theta^{(t)},x,y)G_{j,s}(w_s^{(t)},x) : (x,y)\sim\mathcal{D}, j\in J_s(w_s^{(t)},x)\}$ (that depends on $w_{r,s}^{(0)}$), there exist at least $\Omega\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r\in[m/k]$ of the expert $s\in[k]$ such that,*

$$\left\|\frac{\tilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Proof. Let us pick S samples to form $\mathbf{S} = \{(x_i,y_i)\}_{i=1}^S$ with $S/2$ many samples from $y=+1$ such that $\frac{1}{2}p_{1,s}^{(t)}S$ many samples of them satisfy the event $\mathcal{E}_{1,s}^{(t)}$ and $S/2$ many samples from $y=-1$ such that $\frac{1}{2}p_{2,s}^{(t)}S$ many samples of them satisfy the event $\mathcal{E}_{2,s}^{(t)}$. We denote the subset of $\mathbf{S}$ satisfying the event $\mathcal{E}_{1,s}^{(t)}$ by $\mathbf{S}_1$ and the subset of $\mathbf{S}$ satisfying the event $\mathcal{E}_{2,s}^{(t)}$ by $\mathbf{S}_2$. Therefore, $|\mathbf{S}_1| = \frac{1}{2}p_{1,s}^{(t)}S$ and $|\mathbf{S}_2| = \frac{1}{2}p_{2,s}^{(t)}S$. Now, w.h.p. :

$$\left|v_{1,s}^{(t)} - \frac{2}{p_{1,s}^{(t)}S}\sum_{(x_i,y_i)\in\mathbf{S}_1}p_{1,s}^{(t)}G_{j_{o_1},s}^{(t)}(x_i)v^{(t)}(\theta^{(t)},x_i,y_i)\right| = \tilde{O}\left(\frac{1}{\sqrt{p_{1,s}^{(t)}S}}\right) \text{ and}$$

$$\left|v_{2,s}^{(t)} - \frac{2}{p_{2,s}^{(t)}S}\sum_{(x_i,y_i)\in\mathbf{S}_2}p_{2,s}^{(t)}G_{j_{o_2},s}^{(t)}(x_i)v^{(t)}(\theta^{(t)},x_i,y_i)\right| = \tilde{O}\left(\frac{1}{\sqrt{p_{2,s}^{(t)}S}}\right)$$

This implies that, as long as $S = \tilde{\Omega}\left(\frac{1}{(v_s^{(t)})^2}\right)$, we will have that,

$$\max\left\{\frac{2}{p_{1,s}^{(t)}S}\sum_{(x_i,y_i)\in\mathbf{S}_1}p_{1,s}^{(t)}G_{j_{o_1},s}^{(t)}(x_i)v^{(t)}(\theta^{(t)},x_i,y_i),\right.$$
$$\left.\frac{2}{p_{2,s}^{(t)}S}\sum_{(x_i,y_i)\in\mathbf{S}_2}p_{2,s}^{(t)}G_{j_{o_2},s}^{(t)}(x_i)v^{(t)}(\theta^{(t)},x_i,y_i)\right\} \in \left[\frac{1}{2}v_s^{(t)}, \frac{3}{2}v_s^{(t)}\right]$$

Now using the same procedure as in Lemma G.3 and using Lemma H.2 we can show that, for a fixed set $\{v^{(t)}(\theta^{(t)},x_i,y_i)G_{j,s}(w_s^{(t)},x_i) : (x_i,y_i)\in\mathbf{S}, j\in J_s(w_s^{(t)},x_i)\}$ as long as $S = \tilde{\Omega}\left(\frac{1}{(v_s^{(t)})^2}\right)$, the probability that there are less than $O\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/k]$ such that $\left\|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x_i, y_i)}{\partial w_{r,s}^{(t)}}\right\|$ is $\tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$ is no more than p_{fix} where, $p_{\text{fix}} \le \exp\left(-\Omega\left(\frac{m}{kp\sqrt{\delta}}\right)\right)$.

Now, for every $\bar{\varepsilon} > 0$, for two different $\{v^{(t)}(\theta^{(t)}, x_i, y_i)G_{j,s}(w_s^{(t)}, x_i) : (x_i, y_i) \in \mathbf{S}, j \in J_s(w_s^{(t)}, x_i)\}$, $\{v'^{(t)}(\theta^{(t)}, x_i, y_i)G'_{j,s}(w_s^{(t)}, x_i) : (x_i, y_i) \in \mathbf{S}, j \in J_s(w_s^{(t)}, x_i)\}$ such that $\forall (x_i, y_i) \in \mathbf{S}, j \in J_s(w_s^{(t)}, x_i)$, $|v^{(t)}(\theta^{(t)}, x_i, y_i)G_{j,s}(w_s^{(t)}, x_i) - v'^{(t)}(\theta^{(t)}, x_i, y_i)G'_{j,s}(w_s^{(t)}, x_i)| \le \bar{\varepsilon}$, w.h.p.,

$$\Bigg\|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} \frac{-ya_{r,s}}{l} \sum_{j\in J_s(w_s^{(t)},x_i)} \Big(v^{(t)}(\theta^{(t)}, x_i, y_i)G_{j,s}(w_s^{(t)}, x_i) - v'^{(t)}(\theta^{(t)}, x_i, y_i)G'_{j,s}(w_s^{(t)}, x_i)\Big) x_i^{(j)} 1_{\langle w_{r,s}^{(0)}, x_i^{(j)}\rangle \ge 0}\Bigg\| = \tilde{O}(\bar{\varepsilon})$$

Therefore taking $\bar{\varepsilon}$-net with $\bar{\varepsilon} = \tilde{\Theta}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$ we can show that the probability that there exists $\{v^{(t)}(\theta^{(t)}, x_i, y_i)G_{j,s}(w_s^{(t)}, x_i) : (x_i, y_i) \in \mathbf{S}, j \in J_s(w_s^{(t)}, x_i)\}$ such that there are no more than $O\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/k]$ with $\left\|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x_i, y_i)}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$ is no more than, $p \le p_{\text{fix}}\left(\frac{v_s^{(t)}}{\bar{\varepsilon}}\right)^{Sl} \le \exp\left(-\Omega\left(\frac{m}{kp\sqrt{\delta}}\right) + Sl\log\left(\frac{v_s^{(t)}}{\bar{\varepsilon}}\right)\right)$.

Hence, for $m = \widetilde{\Omega}\left(kSlp\sqrt{\delta}\right)$ with $S = \tilde{\Omega}\left(\frac{1}{(v_s^{(t)})^2}\right)$, w.h.p. for every possible choice of $\{v^{(t)}(\theta^{(t)}, x_i, y_i)G_{j,s}(w_s^{(t)}, x_i) : (x_i, y_i) \in \mathbf{S}, j \in J_s(w_s^{(t)}, x_i)\}$, there are at least $\Omega\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r \in [m/k]$ such that,

$$\left\|\frac{1}{S}\sum_{(x_i,y_i)\in\mathbf{S}} \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x_i, y_i)}{\partial w_{r,s}^{(t)}}\right\| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now as $\left\|\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x_i, y_i)}{\partial w_{r,s}^{(t)}}\right\| = \widetilde{O}(1/l)$, using the same procedure as in Lemma G.3 we can complete the proof which gives us $m = \tilde{\Omega}\left(\frac{klp^3\delta^{3/2}}{(v_s^{(t)})^2}\right)$. ☐

Lemma H.4. *Let us define* $v^{(t)} := \sqrt{\sum_{s\in[k]}(v_s^{(t)})^2}$ *where* $v_s^{(t)} = \max\{v_{1,s}^{(t)}, v_{2,s}^{(t)}\}$ *for all* $s \in [k]$*;* $\gamma := \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$*. Then, by selecting learning rate* $\eta = \tilde{O}\left(\frac{\gamma^3(v^{(t)})^2 l^3}{mk^2}\right)$ *and batch size* $B = \tilde{\Omega}\left(\frac{k^2}{\gamma^6(v^{(t)})^4}\right)$*, at each iteration* t *of the Step-2 of Algorithm 2 such that* $t = \tilde{O}\left(\frac{\sigma\gamma^3(v^{(t)})^2 l^2}{\eta nk}\right)$*, w.h.p. we can ensure that,*

$$\Delta L(\theta^{(t)}, \theta^{(t+1)}) \ge \frac{\eta m\gamma^3}{l^2}\tilde{\Omega}\left((v^{(t)})^2\right)$$

Proof. As w.h.p. $\left\|\frac{\tilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_{r,s}^{(t)}}\right\| = \widetilde{O}(1/l)$, for a randomly sampled batch $\mathcal{B}_t$ of size B, by selecting $\tau = \frac{\sigma\gamma}{4enB}$ in Lemma H.1 and using the same procedure as in Lemma G.4, we can show that for at least $\gamma/2$ fraction of $r \in [m/k]$ of expert $s \in [k]$:

$$\left|\left|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_{r,s}^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v_s^{(t)}}{lp\sqrt{\delta}}\right)$$

Now, for any $(x', y') \sim \mathcal{D}$, from Lemma H.1 we know that for at least $1 - \frac{2e\tau n}{\sigma}$ fraction of $r \in [m/k]$ of any expert $s \in [k]$, the loss function is $\tilde{O}(1/l)$-Lipschitz smooth and also $\tilde{O}(1/l)$-smooth.

Therefore, using same procedure as in Lemma G.4 we can complete the proof. □

I. Lemmas Used to Prove the Theorem 4.3

For the single CNN model, as all the patches of an input $(x, y) \sim \mathcal{D}$ are sent to the model (i.e., there is no router), the gradient of the single sample loss function w.r.t. hidden node $r \in [m]$,

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_r^{(t)}} = -y a_r v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(t)}, x^{(j)}\rangle \geq 0}\right) \tag{21}$$

the corresponding **pseudo-gradient**,

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_r^{(t)}} = -y a_r v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)$$

and the expected pseudo-gradient,

$$\begin{aligned}
\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_r^{(t)}} &= \mathbb{E}_{\mathcal{D}}\left[\frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_r^{(t)}}\right] \\
&= -\frac{a_r}{2}\left(\mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)\Big| y = +1\right]\right. \\
&\qquad \left. -\mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, P_j x\rangle \geq 0}\right)\Big| y = -1\right]\right) \\
&= -\frac{a_r}{2} P_r^{(t)}
\end{aligned}$$

where,

$$\begin{aligned}
P_r^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}&\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n])} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)\Big| y = +1\right] \\
&- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)\Big| y = -1\right]
\end{aligned}$$

Lemma I.1. *W.h.p. over the random initialization, for every $(x, y) \sim \mathcal{D}$ and for every $\tau > 0$, for every iteration $t = \tilde{O}\left(\frac{\tau}{\eta}\right)$ of the minibatch SGD, we have that for at least $\left(1 - \frac{2e\tau n}{\sigma}\right)$ fraction of $r \in [m]$:*

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_r^{(t)}} = \frac{\widetilde{\partial}\mathcal{L}(\theta^{(t)}, x, y)}{\partial w_r^{(t)}} \text{ and } |\langle w_r^{(t)}, x^{(j)}\rangle| \geq \tau, \forall j \in [n]$$

Proof. Using similar argument as in Lemma G.1 we can show that w.h.p., $\left|\left|\frac{\partial \mathcal{L}(\theta^{(t)},x,y)}{\partial w_r^{(t)}}\right|\right| = \tilde{O}\left(1\right)$ so as the mini-batch gradient, $\left|\left|\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial w_r^{(t)}}\right|\right| = \tilde{O}\left(1\right)$.

Therefore, $\left|\left|w_r^{(t)} - w_r^{(0)}\right|\right| = \tilde{O}\left(1\right)$.

Now, for every $\tau > 0$, considering the set $\mathcal{H} := \left\{r \in [m] : \forall j \in [n], |\langle w_r^{(0)}, x^{(j)}\rangle| \geq 2\tau\right\}$ and following the same procedure as in Lemma G.1 we can complete the proof.

□

Recall, $v_1^{(t)} := \mathbb{E}_{\mathcal{D}|y=+1}[v^{(t)}(\theta^{(t)}, x, y)|y = +1]$ and $v_2^{(t)} := \mathbb{E}_{\mathcal{D}|y=-1}[v^{(t)}(\theta^{(t)}, x, y)|y = -1]$.

Lemma I.2. *For any possible fixed set* $\{v^{(t)}(\theta^{(t)}, x, y) : (x, y) \sim \mathcal{D}\}$ *(that does not depend on* $w_r^{(0)}$*) such that* $v^{(t)} = v_1^{(t)} = \max\{v_1^{(t)}, v_2^{(t)}\}$*, we have:*

$$\mathbb{P}\left[||P_r^{(t)}|| = \widetilde{\Omega}\left(\frac{v^{(t)}}{np\sqrt{\delta}}\right)\right] = \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$$

Proof. We know that,

$$\begin{aligned} P_r^{(t)} = \mathbb{E}_{\mathcal{D}|y=+1}&\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)\Big|y = +1\right] \\ &- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} x^{(j)} 1_{\langle w_r^{(0)}, x^{(j)}\rangle \geq 0}\right)\Big|y = -1\right] \end{aligned}$$

Therefore,

$$\begin{aligned} &h(w_r^{(0)}) := \langle P_r, w_r^{(0)}\rangle \\ &= \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} \textbf{ReLU}\left(\langle w_r^{(0)}, x^{(j)}\rangle\right)\right)\Big|y = +1\right] \\ &\quad - \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} \textbf{ReLU}\left(\langle w_r^{(0)}, x^{(j)}\rangle\right)\right)\Big|y = -1\right] \end{aligned}$$

Now, decomposing $w_r^{(0)} = \alpha o_1 + \beta$ with $\beta \perp o_1$ we get,

$$\begin{aligned} h(w_r^{(0)}) &= \frac{v^{(t)}}{n}\textbf{ReLU}\left(\alpha\right) \\ &+ \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]/j_{o_1}} \textbf{ReLU}\left(\alpha\langle o_1, x^{(j)}\rangle + \langle \beta, x^{(j)}\rangle\right)\right)\right] \\ &- \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)}, x, y)\left(\frac{1}{n}\sum_{j\in[n]} \textbf{ReLU}\left(\alpha\langle o_1, x^{(j)}\rangle + \langle \beta, x^{(j)}\rangle\right)\right)\right] \\ &= \phi(\alpha) - l(\alpha) \end{aligned}$$

where,

$$\phi(\alpha) := \frac{v^{(t)}}{n}\textbf{ReLU}\left(\alpha\right) + \mathbb{E}_{\mathcal{D}|y=+1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{n}\sum_{j\in[n]/j_{o_1}}\textbf{ReLU}\left(\alpha\langle o_1, x^{(j)}\rangle + \langle \beta, x^{(j)}\rangle\right)\right)\right]$$

and

$$l(\alpha) := \mathbb{E}_{\mathcal{D}|y=-1}\left[v^{(t)}(\theta^{(t)},x,y)\left(\frac{1}{n}\sum_{j\in[n]}\textbf{ReLU}\left(\alpha\langle o_1, x^{(j)}\rangle + \langle \beta, x^{(j)}\rangle\right)\right)\right]$$

Now as $\phi(\alpha)$ and $l(\alpha)$ both are convex functions, using the same procedure as in Lemma G.1 we can complete the proof. □

Lemma I.3. *Let $v^{(t)} = \max\{v_1^{(t)}, v_2^{(t)}\}$. Then, for every $v^{(t)} > 0$, for $m = \tilde{\Omega}\left(\frac{n^2p^3\delta^{3/2}}{(v^{(t)})^2}\right)$, for every possible set $\{v^{(t)}(\theta^{(t)},x,y)(w_s^{(t)},x) : (x,y)\sim\mathcal{D}\}$ (that depends on $w_r^{(0)}$), there exist at least $\Omega\left(\frac{1}{p\sqrt{\delta}}\right)$ fraction of $r\in[m]$ such that,*

$$\left|\left|\frac{\widetilde{\partial}\hat{\mathcal{L}}(\theta^{(t)})}{\partial w_r^{(t)}}\right|\right| = \tilde{\Omega}\left(\frac{v^{(t)}}{np\sqrt{\delta}}\right)$$

Proof. Similar as in the proof of Lemma G.3, by picking S samples from the distribution $\mathcal{D}$ to form the set $\mathbf{S} = \{(x_i,y_i)\}_{i=1}^S$ such that $S/2$ many samples from $y=+1$ (denoting the sub-set by $\mathbf{S}_{+1}$) and $S/2$ many samples from $y=-1$ (denoting the sub-set by $\mathbf{S}_{-1}$), we can show that w.h.p.,

$$\left|v_1^{(t)} - \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_{+1}} v^{(t)}(\theta^{(t)},x_i,y_i)\right| = \tilde{O}\left(\frac{1}{\sqrt{S}}\right) \text{ and}$$
$$\left|v_2^{(t)} - \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_{-1}} v^{(t)}(\theta^{(t)},x_i,y_i)\right| = \tilde{O}\left(\frac{1}{\sqrt{S}}\right)$$

This implies that, as long as $S = \tilde{\Omega}\left(\frac{1}{(v^{(t)})^2}\right)$ we have,

$$\max\left\{\frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_{+1}} v^{(t)}(\theta^{(t)},x_i,y_i), \frac{1}{S/2}\sum_{(x_i,y_i)\in\mathbf{S}_{-1}} v^{(t)}(\theta^{(t)},x_i,y_i)\right\} \in \left[\frac{1}{2}v^{(t)}, \frac{3}{2}v^{(t)}\right]$$

Now using Lemma I.2 and following similar procedure as in Lemma G.3 we can complete the proof. □

Lemma I.4. *With $v^{(t)} = \max\{v_1^{(t)}, v_2^{(t)}\}$ and $\gamma = \Omega\left(\frac{1}{p\sqrt{\delta}}\right)$, by selecting learning rate $\eta = \tilde{O}\left(\frac{\gamma^3(v^{(t)})^2}{mn^2}\right)$ and batch-size $B = \tilde{\Omega}\left(\frac{n^4}{\gamma^6(v^{(t)})^4}\right)$, for $t = \tilde{O}\left(\frac{\sigma\gamma^3(v^{(t)})^2}{\eta n^3}\right)$ iterations of SGD, w.h.p. we can ensure that,*

$$\Delta L(\theta^{(t)},\theta^{(t+1)}) \geq \frac{\eta m\gamma^3}{n^2}\tilde{\Omega}\left((v^{(t)})^2\right)$$

Proof. Using Lemma I.1 and I.3 and following similar technique as in Lemma G.4, the proof can be completed. □

J. Auxiliary Lemmas

Lemma J.1. *(Li & Liang, 2018) Let $\psi : \mathbb{R} \to \mathbb{R}$ and $\zeta : \mathbb{R} \to \mathbb{R}$ are convex functions. Let $\{\partial\psi(x)\}$ and $\{\partial\zeta(x)\}$ are the sets of sub-gradient of ψ and ζ at x respectively such that $\partial_{max}\psi(x) = max\{\partial\psi(x)\}$, $\partial_{max}\zeta(x) = max\{\partial\zeta(x)\}$, $\partial_{min}\psi(x) = min\{\partial\psi(x)\}$ and $\partial_{min}\zeta(x) = min\{\partial\zeta(x)\}$. Then for any $\tau \geq 0$ such that $\gamma = (\partial_{max}\psi(\tau/2) - \partial_{min}\psi(-\tau/2) - (\partial_{max}\zeta(\tau/2) - \partial_{min}\zeta(-\tau/2)$,*

$$\mathbb{P}_{\alpha\sim U(-\tau,\tau)}\left[|\psi(\alpha) - \zeta(\alpha)| \geq \tfrac{\tau\gamma}{512}\right] \geq \tfrac{1}{64}$$

Lemma J.2. *(Li & Liang, 2018) Let for any $i \in [m]$, the function $h_i : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz smooth and there exists $r \in [m]$ such that for all $i \in [m-r]$ the function h_i is also L-smooth. Furthermore, let us assume that the function $g : \mathbb{R} \to \mathbb{R}$ is both L-Lipschitz smooth and L-smooth. Let define $f(w) := g\left(\sum_{i\in[m]} h_i(w_i)\right)$ where $w \in \mathbb{R}^{dm}$ such that $w_i \in \mathbb{R}^d$. Then for every $\xi \in \mathbb{R}^{dm}$ such that $\xi_i \in \mathbb{R}^d$ with $||\xi_i|| \leq \rho$, we have:*

$$g\left(\sum_{i\in[m]} h_i(w_i+\xi_i)\right) - g\left(\sum_{i\in[m]} h_i(w_i)\right) \leq \sum_{i\in[m-r]} \left\langle \frac{\partial f(w)}{\partial w_i}, \xi_i \right\rangle + L^3m^2\rho^2 + L^2 r\rho$$

K. Proof of the Non-linear Separability of the Data-model

Lemma K.1. *As long as $l^* = \Omega(1)$, the distribution $\mathcal{D}$ is **NOT** linearly separable.*

Proof. We will prove the Lemma by contradiction.

Now, if the distribution, $\mathcal{D}$ is linearly separable, then there exists a hyperplane $h = \left[h^{(1)T}, h^{(2)T}, ..., h^{(n)T}\right]$ with $||h|| = 1$ (here, $h^{(j)}$ represents the j-th patch of the hyperplane for $j \in [n]$) such that,

$$\forall (x_1, y = +1) \sim \mathcal{D} \text{ and } (x_2, y = -1) \sim \mathcal{D}, x_1^T h - x_2^T h \geq 0 \tag{22}$$

Now, as the class-discriminative patterns o_1 and o_2 can occur at any position $j \in [n]$, $||h^{(j)}||^2 = \Theta(\frac{1}{n});\ \forall j \in [n]$.

Now, $\forall j \in [n]$, we can decompose $h^{(j)}$ as $h^{(j)} = a_j o_1 + b_j o_2$.

Then, $|a_j| = |b_j| = \Theta\left(\frac{1}{\sqrt{n(1-\delta_d)}}\right), \forall j \in [n]$ as $||o_1|| = ||o_2|| = 1$.

Now,

$$x_1^T h - x_2^T h = \langle o_1, h^{(j_{o_1})}\rangle - \langle o_2, h^{(j_{o_2})}\rangle + \sum_{j\in[n]/j_{o_1}} \langle x_1^{(j)}, h^{(j)}\rangle - \sum_{j\in[n]/j_{o_2}} \langle x_2^{(j)}, h^{(j)}\rangle$$

Now,

$$\begin{aligned}
&\langle o_1, h^{(j_{o_1})}\rangle - \langle o_2, h^{(j_{o_2})}\rangle = (a_{j_{o_1}} - b_{j_{o_2}}) - (a_{j_{o_2}} - b_{j_{o_1}})\delta_d \\
&\leq |a_{j_{o_1}} - b_{j_{o_2}}| - |a_{j_{o_1}} - b_{j_{o_2}}|\delta_d \qquad \text{[WLOG, let assume } \delta_d < 0] \\
&= O\left(\sqrt{\frac{1-\delta_d}{n}}\right)
\end{aligned}$$

Now,

$$\begin{aligned}
&\sum_{j\in[n]/j_{o_2}} \langle x_2^{(j)}, h^{(j)}\rangle - \sum_{j\in[n]/j_{o_1}} \langle x_1^{(j)}, h^{(j)}\rangle = \sum_{j\in[n]/j_{o_2}} \langle x_2^{(j)}, a_j o_1 + b_j o_2\rangle - \sum_{j\in[n]/j_{o_1}} \langle x_1^{(j)}, a_j o_1 + b_j o_2\rangle \\
&= O\left(l^* \sqrt{\frac{(1-\delta_d)}{n}}\right)
\end{aligned}$$

Therefore, for $l^* = \Omega(1)$ there is contradiction with (22). ☐

L. WRN and WRN-pMoE Architectures Implemented in the Experiments

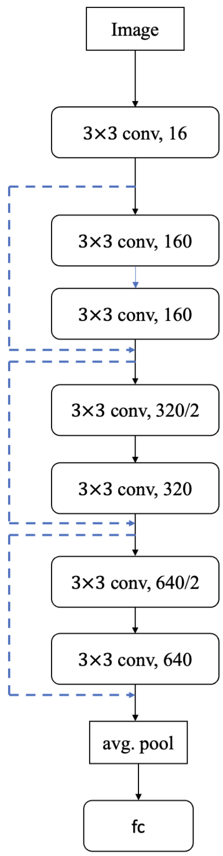

Figure 17. The WRN architecture implemented to learn CelebA dataset

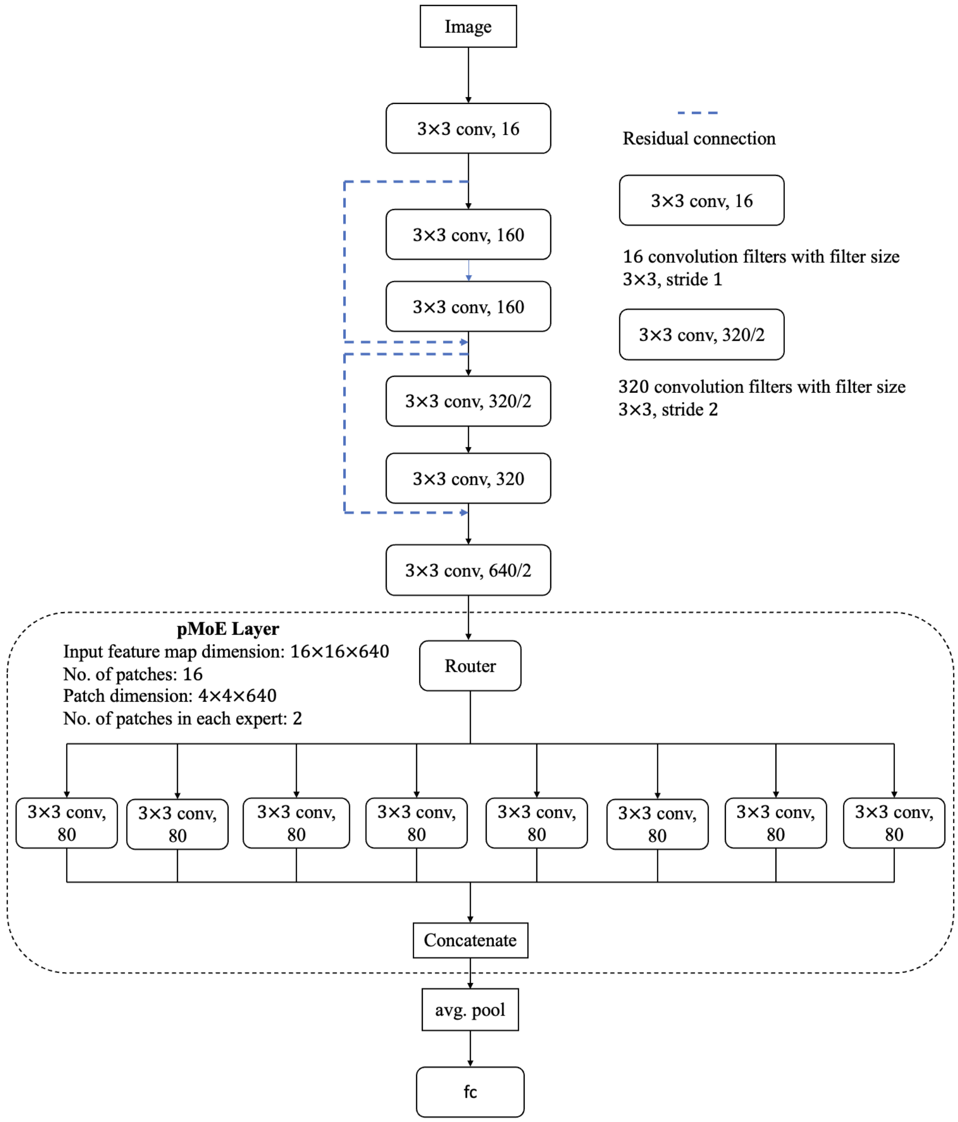

Figure 18. The WRN-pMoE architecture implemented to learn CelebA dataset

M. Extension to Multi-class Classification

Let us consider c-class classification problem where $c > 2$. Then, we have $(x, y) \sim \mathcal{D}_c$ where $y \in \{1, 2, ..., c\}$ for the multi-class distribution $\mathcal{D}_c$.

The multi-class data model:
Now, according to the data model presented in section 4.2, we have $\{o_1, o_2, ..., o_c\}$ as class-discriminative pattern set. $\forall j, j' \in [c]$ such that $j \neq j'$, we define $\delta_{d_{j,j'}} := \langle o_j, o_{j'} \rangle$. We further define $\delta_d := \max\{\delta_{d_{j,j'}}\}$. Then,

$$\delta = \frac{1}{(1 - \max\{\delta_{d_{j,j'}}^2, \delta_r^2\}_{j,j' \in [c], j \neq j'})}$$

The multi-class pMoE model:
The pMoE model for multi-class case is given by,

$$\forall i \in [c], f_{M_i}(\theta, x) = \sum_{s=1}^{k}\sum_{r=1}^{\frac{m}{k}} \frac{a_{r,s,i}}{l} \sum_{j \in J_s(w_s,x)} \textbf{ReLU}(\langle w_{r,s}, x^{(j)} \rangle) G_{j,s}(w_s, x) \tag{23}$$

An illustration of (23) is given in Figure 19.

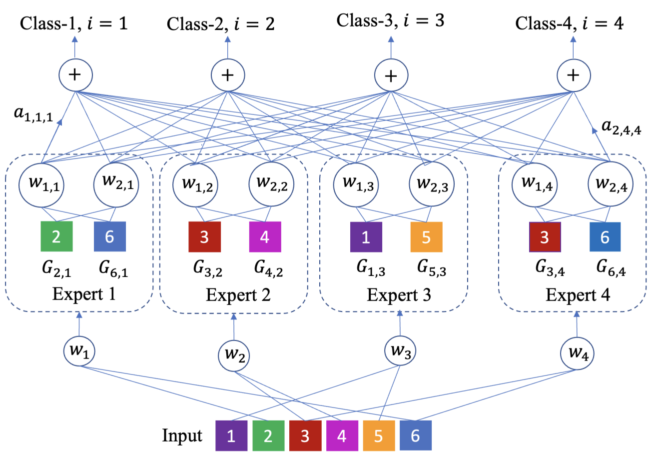

Figure 19. An illustration of the pMoE model in (23) with $c = 4, k = 4, m = 8, n = 6$ and $l = 2$.

For mult-class case, we replace the logistic loss function by the softmax loss function (also known as cross-entropy loss). For the training dataset $\{x_j, y_j\}_{j=1}^N$, we minimize the following empirical risk minimization problem:

$$\min_{\theta} : \quad L(\theta) = \frac{1}{N}\sum_{j=1}^{N} \log \frac{\sum_{i=1}^{c} \exp\left(f_{M_i}(\theta, x_j)\right)}{\exp\left(f_{M_{y_j}}(\theta, x_j)\right)} \tag{24}$$

M.1. The Multi-class Separate-training pMoE

Number of experts: For the multi-class separate-training pMoE, we take $k = c$, i.e. number of experts is equal to the number of classes.

Training algorithm:
Input : Training data $\{(x_i, y_i)\}_{i=1}^N$, learning rates η_r and η, number of iterations T_r and T, batch-sizes B_r and B
Step-1: Initialize $w_s^{(0)}, w_{r,s}^{(0)}, a_{r,s}, \forall s \in \{1, 2\}, r \in [m/k]$ according to (7) and (8)
Step-2: (Pair-wise router training) We train the router, i.e. the gating-kernels $w_1, w_2, ..., w_c$ using pair-wise training describe below:

1. At first, we separate the training set of N_r samples into c disjoint subsets $\{N_{r,1}, N_{r,2}, ..., N_{r,c}\}$ according to the class-labels.
2. Now, we prepare $c/2$ pairs of training sets $\{(N_{r,1}, N_{r,2}), (N_{r,3}, N_{r,4}), ..., (N_{r,c-1}, N_{r,c})\}$ (here WLOG we assume that c is even).
3. Under each pair $(N_{r,i}, N_{r,i+1})$, we re-define the label as $y = +1$ and $y = -1$ for the class i and $i + 1$ respectively and train the gating-kernels w_i and w_{i+1} by minimizing (6) for T_r iterations
4. After the end of pair-wise training for all the pairs $\{(N_{r,1}, N_{r,2}), (N_{r,3}, N_{r,4}), ..., (N_{r,c-1}, N_{r,c})\}$, we receive $w_1^{(T_r)}, w_2^{(T_r)}, ..., w_c^{(T_r)}$ as the learned gating-kernels.

Step-3:(Expert training)
Using the learned gating-kernels $w_1^{(T_r)}, w_2^{(T_r)}, ..., w_c^{(T_r)}$ in Step-2 and using the same procedure as in Step-3 of Algorithm 1 we train the experts.

The multi-class counterpart of the Lemma 4.1:
Now, using the same proof techniques as for Lemma 4.1 (i.e. following same procedures as in section D and E) we can show that, we need $N_r = \Omega(\frac{c^2n^2}{(1-\delta_d)^2})$ training samples to ensure,

$$\arg_{j\in[n]}(x^{(j)} = o_i) \in J_i(w_i^{(T_r)}, x) \qquad \forall (x, y = i) \sim \mathcal{D}_c \text{ and } \forall i \in [c]$$

The multi-class counterpart of the Theorem 4.2:
We redefine the **value-function** for each class $i \in [c]$ as,

$$v_{i,a}^{(t)}(\theta^{(t)}, x, y = a) := \begin{cases} \frac{\sum_{j\neq a} e^{f_{M_j}(\theta^{(t)},x)}}{\sum_{j=1}^{c} e^{f_{M_j}(\theta^{(t)},x)}}; & \text{if, } i = a \\ \\ -\frac{e^{f_{M_i}(\theta^{(t)},x)}}{\sum_{j=1}^{c} e^{f_{M_j}(\theta^{(t)},x)}}; & \text{otherwise} \end{cases} \tag{25}$$

Now using similar techniques as in the proof of Theorem 4.2 (i.e. following same procedure as in the proof of Theorem F.3 and section G) we can show that for every $\epsilon > 0$, we need number of hidden nodes $m \geq M_S = \Omega\left(l^{10}p^{12}\delta^6c^{11}/\epsilon^{16}\right)$, batch-size $B = \Omega\left(l^4p^6\delta^3c^6/\epsilon^8\right)$ for $T = O\left(l^4p^6\delta^3c^6/\epsilon^8\right)$ iterations (i.e. $N_S = \Omega(l^8p^{12}\delta^6c^{12}/\epsilon^{16})$) to ensure,

$$\underset{(x,y)\sim\mathcal{D}_c}{\mathbb{P}}\left[\forall j \in [c], j \neq y, f_{M_y}(\theta^{(T)}, x) > f_{M_j}(\theta^{(T)}, x)\right] \geq 1 - \epsilon$$

M.2. The Multi-class Joint-training pMoE

Training algorithm: Same as the Algorithm 2 except that for multi-class case the loss function is softmax instead of logistic loss.

The multi-class counterpart of the Theorem 4.5:
Using the value-function define in (25) and as long as the Assumption 4.4 satisfied for all the classes $i \in [c]$, following the similar techniques as in the proof of Theorem 4.5 (i.e. following same procedure as in the proof of Theorem F.5 and section H), we can show that for every $\epsilon > 0$, we need number of hidden nodes $m \geq M_J = \Omega\left(k^3n^2l^6p^{12}\delta^6c^8/\epsilon^{16}\right)$, batch-size $B = \Omega\left(k^2l^4p^6\delta^3c^4/\epsilon^8\right)$ for $T = O\left(k^2l^2p^6\delta^3c^4/\epsilon^8\right)$ iterations (i.e. $N_J = \Omega(k^4l^6p^{12}\delta^6c^8/\epsilon^{16})$) to ensure,

$$\underset{(x,y)\sim\mathcal{D}_c}{\mathbb{P}}\left[\forall j \in [c], j \neq y, f_{M_y}(\theta^{(T)}, x) > f_{M_j}(\theta^{(T)}, x)\right] \geq 1 - \epsilon$$

N. Details of the Results in Table 1

Complexity in forward pass. The computational complexity of a non-overlapping convolution operation by a filter of dimension d on an input sample of n patches (of same dimension as the filter) is $O(nd)$ (Vaswani et al., 2017). Therefore, the complexity of forward pass of a batch of size B through a convolution layer of m neurons is $O(Bmnd)$. Similarly, the forward pass complexity of a the batch through the experts (of same total number of neurons as in the convolution layer) of a pMoE layer is $O(Bmld)$. The operations in a pMoE router includes convolution (with complexity $O(nd)$), softmax operation (with complexity $O(1)$) and TOP-l operation (with complexity $O(nl)$ when $l \ll n$). Therefore, the overall forward pass complexity of a pMoE router with k expert is $O(Bknd)$.

Complexity in backward pass. The gradient of neurons in convolution layer for an input sample is given in (21), which implies that the complexity of the gradient calculation is $O(nd)$ (addition of n vectors of dimension d) and hence the backward pass complexity of CNN is $O(Bmnd)$. Similarly, the backward pass complexity of pMoE experts is $O(Bmld)$. Now the gradient of gating kernels in pMoE router is given in (26), which implies that the complexity of the gradient calculation is $O(l^2d)$ (addition of l^2 vectors of dimension d) and hence the backward pass complexity of pMoE router is $O(Bkl^2d)$.

$$\frac{\partial \mathcal{L}(\theta^{(t)}, x, y)}{\partial w_s^{(t)}} = -yv^{(t)}(\theta^{(t)}, x, y)\left(\sum_{r\in[m]} a_{r,s}\left(\frac{1}{l}\sum_{j\in J_s(x)} \textbf{ReLU}(\langle w_{r,s}^{(t)}, x^{(j)}\rangle)G_{j,s}(\sum_{i\in J_s(x)/j}(x^{(j)}-x^{(i)})G_{i,s})\right)\right) \tag{26}$$

Complexity to achieve ϵ generalization error. From Theorem 4.5, to achieve ϵ generalization error we need $O(k^2l^2/\epsilon^8)$ iterations of training in pMoE, which implies that the computational complexity to achieve ϵ error in pMoE is $O(Bmk^2l^3d/\epsilon^8)$. Similarly, using the results from Theorem 4.3, the corresponding complexity in CNN is $O(Bmn^5d/\epsilon^8)$.