# Towards Understanding the Mixture-of-Experts Layer in Deep Learning

**Zixiang Chen**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
chenzx19@cs.ucla.edu

**Yihe Deng**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
yihedeng@cs.ucla.edu

**Yue Wu**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
ywu@cs.ucla.edu

**Quanquan Gu**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
qgu@cs.ucla.edu

**Yuanzhi Li**
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
yuanzhil@andrew.cmu.edu

## Abstract

The Mixture-of-Experts (MoE) layer, a sparsely-activated model controlled by a router, has achieved great success in deep learning. However, the understanding of such architecture remains elusive. In this paper, we formally study how the MoE layer improves the performance of neural network learning and why the mixture model will not collapse into a single model. Our empirical results suggest that the cluster structure of the underlying problem and the non-linearity of the expert are pivotal to the success of MoE. This motivates us to consider a challenging classification problem with intrinsic cluster structures. Theoretically, we proved that this problem is hard to solve by a single expert such as a two-layer convolutional neural network (CNN). Yet with the MoE layer with each expert being a two-layer CNN, the problem can be solved successfully. In particular, our theory shows that the router can learn the cluster-center features, which helps divide the input complex problem into simpler classification sub-problems that individual experts can conquer. To our knowledge, this is the first theoretical result toward formally understanding the mechanism of the MoE layer for deep learning.

## 1 Introduction

The Mixture-of-Expert (MoE) structure (Jacobs et al., 1991; Jordan and Jacobs, 1994) is a classic design that substantially scales up the model capacity and only introduces small computation overhead. In recent years, the MoE layer (Eigen et al., 2013; Shazeer et al., 2017), which is an extension of the MoE model to deep neural networks, has achieved remarkable success in deep learning. Generally speaking, an MoE layer contains many experts that share the same network architecture and are trained by the same algorithm, with a gating (or routing) function that routes individual inputs to a few experts among all the candidates. Through the sparse gating function, the router in the MoE layer

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

can route each input to the top-$K(K \geq 2)$ best experts (Shazeer et al., 2017), or the single ($K = 1$) best expert (Fedus et al., 2021). This routing scheme only costs the computation of $K$ experts for a new input, which enjoys fast inference time.

Despite the great empirical success of the MoE layer, the theoretical understanding of such architecture is still elusive. In practice, all experts have the same structure, initialized from the same weight distribution (Fedus et al., 2021) and are trained with the same optimization configuration. The router is also initialized to dispatch the data uniformly. It is unclear why the experts can diverge to different functions that are specialized to make predictions for different inputs, and why the router can automatically learn to dispatch data, especially when they are all trained using simple *local search algorithms* such as gradient descent. Therefore, we aim to answer the following questions:

*Why do the experts in MoE diversify instead of collapsing into a single model? And how can the router learn to dispatch the data to the right expert?*

In this paper, in order to answer the above question, we consider the natural "mixture of classification" data distribution with cluster structure and theoretically study the behavior and benefit of the MoE layer. We focus on the simplest setting of the mixture of linear classification, where the data distribution has multiple clusters, and each cluster uses separate (linear) feature vectors to represent the labels. In detail, we consider the data generated as a combination of feature patches, cluster patches, and noise patches (See Definition 3.1 for more details). We study training an MoE layer based on the data generated from the "mixture of classification" distribution using gradient descent, where each expert is chosen to be a two-layer CNN. The main contributions of this paper are summarized as follows:

- We first prove a negative result (Theorem 4.1) that any single expert, such as two-layer CNNs with arbitrary activation function, cannot achieve a test accuracy of more than $87.5\%$ on our data distribution.
- Empirically, we found that the mixture of linear experts performs better than the single expert but is still significantly worse than the mixture of non-linear experts. Figure 1 provides such a result in a special case of our data distribution with four clusters. *Although a mixture of linear models can represent the labeling function of this data distribution with* $100\%$ *accuracy, it fails to learn so after training*. We can see that the underlying cluster structure cannot be recovered by the mixture of linear experts, and neither the router nor the experts are diversified enough after training. In contrast, the mixture of non-linear experts can correctly recover the cluster structure and diversify.
- Motivated by the negative result and the experiment on the toy data, we study a sparsely-gated MoE model with two-layer CNNs trained by gradient descent. We prove that this MoE model can achieve nearly $100\%$ test accuracy *efficiently* (Theorem 4.2).
- Along with the result on the test accuracy, we formally prove that each expert of the sparsely-gated MoE model will be specialized to a specific portion of the data (i.e., at least one cluster), which is determined by the initialization of the weights. In the meantime, the router can learn the cluster-center features and route the input data to the right experts.
- Finally, we also conduct extensive experiments on both synthetic and real datasets to corroborate our theory.

**Notation.** We use lower case letters, lower case bold face letters, and upper case bold face letters to denote scalars, vectors, and matrices respectively. We denote a union of disjoint sets $(A_i : i \in I)$ by $\sqcup_{i\in I} A_i$. For a vector $\mathbf{x}$, we use $\|\mathbf{x}\|_2$ to denote its Euclidean norm. For a matrix $\mathbf{W}$, we use $\|\mathbf{W}\|_F$ to denote its Frobenius norm. Given two sequences $\{x_n\}$ and $\{y_n\}$, we denote $x_n = \mathcal{O}(y_n)$ if $|x_n| \leq C_1|y_n|$ for some absolute positive constant $C_1$, $x_n = \Omega(y_n)$ if $|x_n| \geq C_2|y_n|$ for some absolute positive constant $C_2$, and $x_n = \Theta(y_n)$ if $C_3|y_n| \leq |x_n| \leq C_4|y_n|$ for some absolute constants $C_3, C_4 > 0$. We also use $\widetilde{\mathcal{O}}(\cdot)$ to hide logarithmic factors of $d$ in $\mathcal{O}(\cdot)$. Additionally, we denote $x_n = \mathrm{poly}(y_n)$ if $x_n = \mathcal{O}(y_n^D)$ for some positive constant $D$, and $x_n = \mathrm{polylog}(y_n)$ if $x_n = \mathrm{poly}(\log(y_n))$. We also denote by $x_n = o(y_n)$ if $\lim_{n\to\infty} x_n/y_n = 0$. Finally we use $[N]$ to denote the index set $\{1, \ldots, N\}$.

## 2 Related Work

**Mixture of Experts Model.** The mixture of experts model (Jacobs et al., 1991; Jordan and Jacobs, 1994) has long been studied in the machine learning community. These MoE models are based

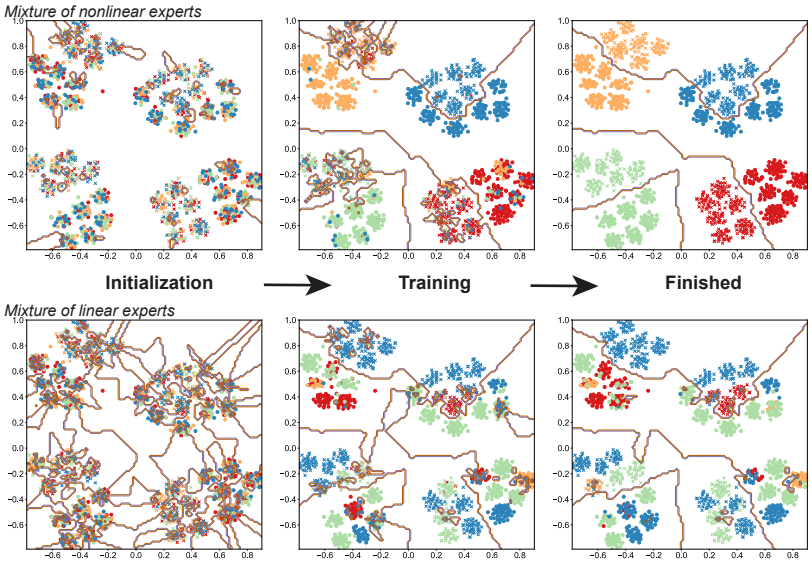


**Figure 1: Visualization of the training of MoE with nonlinear expert and linear expert**. Different colors denote router's dispatch to different experts. The lines denote the decision boundary of the MoE model. The data points are visualized on 2d space via t-SNE (Van der Maaten and Hinton, 2008). The MoE architecture follows section 3 where nonlinear experts use activation function $\sigma(z) = z^3$. For this visualization, we let the expert number $M = 4$ and cluster number $K = 4$. We generate $n = 1,600$ data points from the distribution illustrated in Section 3 with $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (1, 2)$, and $\sigma_p = 1$. More details of the visualization are discussed in Appendix A.

on various base expert models such as support vector machine (Collobert et al., 2002) , Gaussian processes (Tresp, 2001), or hidden Markov models (Jordan et al., 1997). In order to increase the model capacity to deal with the complex vision and speech data, Eigen et al. (2013) extended the MoE structure to the deep neural networks, and proposed a deep MoE model composed of multiple layers of routers and experts. Shazeer et al. (2017) simplified the MoE layer by making the output of the gating function sparse for each example, which greatly improves the training stability and reduces the computational cost. Since then, the MoE layer with different base neural network structures (Shazeer et al., 2017; Dauphin et al., 2017; Vaswani et al., 2017) has been proposed and achieved tremendous successes in a variety of language tasks. Very recently, Fedus et al. (2021) improved the performance of the MoE layer by routing one example to only a single expert instead of $K$ experts, which further reduces the routing computation while preserving the model quality.

**Mixture of Linear Regressions/Classifications.** In this paper, we consider a "mixture of classification" model. This type of models can be dated back to (De Veaux, 1989; Jordan and Jacobs, 1994; Faria and Soromenho, 2010) and has been applied to many tasks including object recognition (Quattoni et al., 2004) human action recognition (Wang and Mori, 2009), and machine translation (Liang et al., 2006). In order to learn the unknown parameters for mixture of linear regressions/classification model, (Anandkumar et al., 2012; Hsu et al., 2012; Chaganty and Liang, 2013; Anandkumar et al., 2014; Li and Liang, 2018) studies the method of moments and tensor factorization. Another line of work studies specific algorithms such as Expectation-Maximization (EM) algorithm (Khalili and Chen, 2007; Yi et al., 2014; Balakrishnan et al., 2017; Wang et al., 2015).

**Theoretical Understanding of Deep Learning.** In recent years, great efforts have been made to establish the theoretical foundation of deep learning. A series of studies have proved the convergence (Jacot et al., 2018; Li and Liang, 2018; Du et al., 2019; Allen-Zhu et al., 2019b; Zou et al., 2018) and generalization (Allen-Zhu et al., 2019a; Arora et al., 2019a,b; Cao and Gu, 2019) guarantees in the so-called "neural tangent kernel" (NTK) regime, where the parameters stay close to the initialization, and the neural network function is approximately linear in its parameters. A recent line of works (Allen-Zhu and Li, 2019; Bai and Lee, 2019; Allen-Zhu and Li, 2020a,b,c; Li et al., 2020; Cao et al., 2022; Zou et al., 2021; Wen and Li, 2021) studied the learning dynamic of neural networks beyond the NTK regime. It is worthwhile to mention that our analysis of the MoE model is also beyond the NTK regime.

## 3 Problem Setting and Preliminaries

We consider an MoE layer with each expert being a two-layer CNN trained by gradient descent (GD) over $n$ independent training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ generated from a data distribution $\mathcal{D}$. In this section, we will first introduce our data model $\mathcal{D}$, and then explain our neural network model and the details of the training algorithm.

### 3.1 Data distribution

We consider a binary classification problem over $P$-patch inputs, where each patch has $d$ dimensions. In particular, each labeled data is represented by $(\mathbf{x}, y)$, where input $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(P)}) \in (\mathbb{R}^d)^P$ is a collection of $P$ patches and $y \in \{\pm 1\}$ is the data label. We consider data generated from $K$ clusters. Each cluster $k \in [K]$ has a label signal vector $\mathbf{v}_k$ and a cluster-center signal vector $\mathbf{c}_k$ with $\|\mathbf{v}_k\|_2 = \|\mathbf{c}_k\|_2 = 1$. For simplicity, we assume that all the signals $\{\mathbf{v}_k\}_{k\in[K]} \cup \{\mathbf{c}_k\}_{k\in[K]}$ are orthogonal with each other.

**Definition 3.1.** A data pair $(\mathbf{x}, y) \in (\mathbb{R}^d)^P \times \{\pm 1\}$ is generated from the distribution $\mathcal{D}$ as follows.

- Uniformly draw a pair $(k, k')$ with $k \neq k'$ from $\{1, \ldots, K\}$.
- Generate the label $y \in \{\pm 1\}$ uniformly, generate a Rademacher random variable $\epsilon \in \{\pm 1\}$.
- Independently generate random variables $\alpha, \beta, \gamma$ from distribution $\mathcal{D}_\alpha, \mathcal{D}_\beta, \mathcal{D}_\gamma$. In this paper, we assume there exists absolute constants $C_1, C_2$ such that almost surely $0 < C_1 \leq \alpha, \beta, \gamma \leq C_2$.
- Generate $\mathbf{x}$ as a collection of $P$ patches: $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(P)}) \in (\mathbb{R}^d)^P$, where
  - **Feature signal.** One and only one patch is given by $y\alpha\mathbf{v}_k$.
  - **Cluster-center signal.** One and only one patch is given by $\beta\mathbf{c}_k$.
  - **Feature noise.** One and only one patch is given by $\epsilon\gamma\mathbf{v}_{k'}$.
  - **Random noise.** The rest of the $P - 3$ patches are Gaussian noises that are independently drawn from $N(0, (\sigma_p^2/d) \cdot \mathbf{I}_d)$ where $\sigma_p$ is an absolute constant.

**How to learn this type of data?** Since the positions of signals and noises are not specified in Definition 3.1, it is natural to use the CNNs structure that applies the same function to each patch. We point out that the strength of the feature noises $\gamma$ can be as large as the strength of the feature signals $\alpha$. As we will see later in Theorem 4.1, this classification problem is hard to learn with a single expert, such as any two-layer CNNs (any activation function with any number of neurons). However, such a classification problem has an intrinsic clustering structure that may be utilized to achieve better performance. Examples can be divided into $K$ clusters $\cup_{k\in[K]}\Omega_k$ based on the cluster-center signals: an example $(\mathbf{x}, y) \in \Omega_k$ if and only if at least one patch of $\mathbf{x}$ aligns with $\mathbf{c}_k$. It is not difficult to show that the binary classification sub-problem over $\Omega_k$ can be easily solved by an individual expert. We expect the MoE can learn this data cluster structure from the cluster-center signals.

**Significance of our result.** Although this data can be learned by existing works on a mixture of linear classifiers with sophisticated algorithms (Anandkumar et al., 2012; Hsu et al., 2012; Chaganty and Liang, 2013), the focus of our paper is training a mixture of nonlinear neural networks, a more practical model used in real applications. When an MoE is trained by variants of gradient descent, we show that the experts *automatically learn to specialize on each cluster*, while the router *automatically learns to dispatch the data to the experts according to their specialty*. Although from a representation point of view, it is not hard to see that the concept class can be represented by MoEs, our result is very significant as we prove that gradient descent from random initialization can find a good MoE with non-linear experts efficiently. To make our results even more compelling, we empirically show that MoE with linear experts, despite also being able to represent the concept class, *cannot* be trained to find a good classifier efficiently.

### 3.2 Structure of the MoE layer

An MoE layer consists of a set of $M$ "expert networks" $f_1, \ldots, f_M$, and a gating network which is generally set to be linear (Shazeer et al., 2017; Fedus et al., 2021). Denote by $f_m(\mathbf{x}; \mathbf{W})$ the output of the $m$-th expert network with input $x$ and parameter $\mathbf{W}$. Define an $M$-dimensional vector $\mathbf{h}(\mathbf{x}; \mathbf{\Theta}) = \sum_{p\in[P]} \mathbf{\Theta}^\top \mathbf{x}^{(p)}$ as the output of the gating network parameterized by $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M] \in \mathbb{R}^{d\times M}$. The output $F$ of the MoE layer can be written as follows:

$$F(\mathbf{x}; \mathbf{\Theta}, \mathbf{W}) = \textstyle\sum_{m\in\mathcal{T}_\mathbf{x}} \pi_m(\mathbf{x}; \mathbf{\Theta}) f_m(\mathbf{x}; \mathbf{W}),$$

where $\mathcal{T}_{\mathbf{x}} \subseteq [M]$ is a set of selected indices and $\pi_m(\mathbf{x}; \mathbf{\Theta})$'s are route gate values given by

$$\pi_m(\mathbf{x}; \mathbf{\Theta}) = \frac{\exp(h_m(\mathbf{x}; \mathbf{\Theta}))}{\sum_{m'=1}^{M} \exp(h_{m'}(\mathbf{x}; \mathbf{\Theta}))}, \forall m \in [M].$$

**Expert Model.** In practice, one often uses nonlinear neural networks as experts in the MoE layer. In fact, we found that the non-linearity of the expert is essential for the success of the MoE layer (see Section 6). For $m$-th expert, we consider a convolution neural network as follows:

$$f_m(\mathbf{x}; \mathbf{W}) = \sum_{j \in [J]} \sum_{p=1}^{P} \sigma\big(\langle \mathbf{w}_{m,j}, \mathbf{x}^{(p)} \rangle\big), \tag{3.1}$$

where $\mathbf{w}_{m,j} \in \mathbb{R}^d$ is the weight vector of the $j$-th filter (i.e., neuron) in the $m$-th expert, $J$ is the number of filters (i.e., neurons). We denote $\mathbf{W}_m = [\mathbf{w}_{m,1}, \ldots, \mathbf{w}_{m,J}] \in \mathbb{R}^{d \times J}$ as the weight matrix of the $m$-th expert and further let $\mathbf{W} = \{\mathbf{W}_m\}_{m \in [M]}$ as the collection of expert weight matrices. For nonlinear CNN, we consider the cubic activation function $\sigma(z) = z^3$, which is one of the simplest nonlinear activation functions (Vecci et al., 1998). We also include the experiment for other activation functions such as RELU in Appendix Table 7.

**Top-1 Routing Model.** A simple choice of the selection set $\mathcal{T}_{\mathbf{x}}$ is the whole experts set $\mathcal{T}_{\mathbf{x}} = [M]$ (Jordan and Jacobs, 1994), which is the case for the so-called soft-routing model. However, it will be time consuming to use soft-routing in deep learning. In this paper, we consider "switch routing", which is introduced by Fedus et al. (2021) to make the gating network sparse and save the computation time. For each input $\mathbf{x}$, instead of using all the experts, we only pick one expert from $[M]$, i.e., $|\mathcal{T}_{\mathbf{x}}| = 1$. In particular, we choose $\mathcal{T}_{\mathbf{x}} = \text{argmax}_m\{h_m(\mathbf{x}; \mathbf{\Theta})\}$.

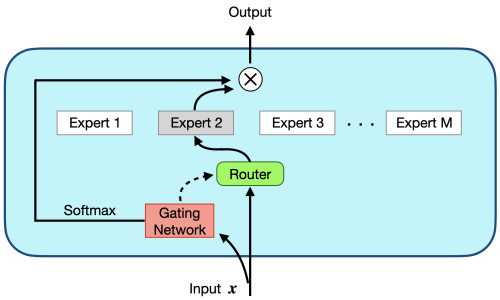


**Figure 2: Illustration of an MoE layer.** For each input $\mathbf{x}$, the router will only select one expert to perform computations. The choice is based on the output of the gating network (dotted line). The expert layer returns the output of the selected expert (gray box) multiplied by the route gate value (softmax of the gating function output).

**Algorithm 1** Gradient descent with random initialization

**Require:** Number of iterations $T$, expert learning rate $\eta$, router learning rate $\eta_r$, initialization scale $\sigma_0$, training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

1: Generate each entry of $\mathbf{W}^{(0)}$ independently from $N(0, \sigma_0^2)$.
2: Initialize each entry of $\mathbf{\Theta}^{(0)}$ as zero.
3: **for** $t = 0, 2, \ldots, T-1$ **do**
4: Generate each entry of $\mathbf{r}^{(t)}$ independently from Unif[0,1].
5: Update $\mathbf{W}^{(t+1)}$ as in (3.4).
6: Update $\mathbf{\Theta}^{(t+1)}$ as in (3.5).
7: **end for**
8: **return** $(\mathbf{\Theta}^{(T)}, \mathbf{W}^{(T)})$.

### 3.3 Training Algorithm

Given the training data $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we train $F$ with gradient descent to minimize the following empirical loss function:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i F(\mathbf{x}_i; \mathbf{\Theta}, \mathbf{W})\big), \tag{3.2}$$

where $\ell$ is the logistic loss defined as $\ell(z) = \log(1 + \exp(-z))$. We initialize $\mathbf{\Theta}^{(0)}$ to be zero and initialize each entry of $\mathbf{W}^{(0)}$ by i.i.d $\mathcal{N}(0, \sigma_0^2)$. Zero initialization of the gating network is widely used in MoE training. As discussed in Shazeer et al. (2017), it can help avoid out-of-memory errors and initialize the network in a state of approximately equal expert load (see (5.1) for the definition of expert load).

Instead of directly using the gradient of empirical loss (3.2) to update weights, we add perturbation to the router and use the gradient of the perturbed empirical loss to update the weights. In particular, the training example $\mathbf{x}_i$ will be distributed to $\text{argmax}_m\{h_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) + r_{m,i}^{(t)}\}$ instead, where $\{r_{m,i}^{(t)}\}_{m \in [M], i \in [n]}$ are random noises. Adding noise term is a widely used training strategy

for sparsely-gated MoE layer (Shazeer et al., 2017; Fedus et al., 2021), which can encourage exploration across the experts and stabilize the MoE training. In this paper, we draw $\{r_{m,i}^{(t)}\}_{m\in[M],i\in[n]}$ independently from the uniform distribution Unif$[0,1]$ and denotes its collection as $\mathbf{r}^{(t)}$. Therefore, the perturbed empirical loss at iteration $t$ can be written as

$$\mathcal{L}^{(t)}(\mathbf{\Theta}^{(t)},\mathbf{W}^{(t)}) = \frac{1}{n}\sum_{i=1}^{n}\ell\big(y_i\pi_{m_{i,t}}(\mathbf{x}_i;\mathbf{\Theta}^{(t)})f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\big), \tag{3.3}$$

where $m_{i,t} = \mathrm{argmax}_m\{h_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)}) + r_{m,i}^{(t)}\}$. Starting from the initialization $\mathbf{W}^{(0)}$, the gradient descent update rule for the experts is

$$\mathbf{W}_m^{(t+1)} = \mathbf{W}_m^{(t)} - \eta\cdot\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}(\mathbf{\Theta}^{(t)},\mathbf{W}^{(t)})/\|\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}(\mathbf{\Theta}^{(t)},\mathbf{W}^{(t)})\|_F, \forall m\in[M], \tag{3.4}$$

where $\eta > 0$ is the expert learning rate. Starting from the initialization $\mathbf{\Theta}^{(0)}$, the gradient update rule for the gating network is

$$\boldsymbol{\theta}_m^{(t+1)} = \boldsymbol{\theta}_m^{(t)} - \eta_r\cdot\nabla_{\boldsymbol{\theta}_m}\mathcal{L}^{(t)}(\mathbf{\Theta}^{(t)},\mathbf{W}^{(t)}), \forall m\in[M], \tag{3.5}$$

where $\eta_r > 0$ is the router learning rate. In practice, the experts are trained by Adam to make sure they have similar learning speeds. Here we use a normalized gradient which can be viewed as a simpler alternative to Adam (Jelassi et al., 2021).

## 4 Main Results

In this section, we will present our main results. We first provide a negative result for learning with a single expert.

**Theorem 4.1** (Single expert performs poorly)**.** Suppose $\mathcal{D}_\alpha = \mathcal{D}_\gamma$ in Definition 3.1, then any function with the form $F(\mathbf{x}) = \sum_{p=1}^{P} f(\mathbf{x}^{(p)})$ will get large test error $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yF(\mathbf{x})\le 0\big)\ge 1/8$.

Theorem 4.1 indicates that if the feature noise has the same strength as the feature signal i.e., $\mathcal{D}_\alpha = \mathcal{D}_\gamma$, any two-layer CNNs with the form $F(\mathbf{x}) = \sum_{j\in[J]} a_j\sum_{p\in[P]}\sigma(\mathbf{w}_j^\top\mathbf{x}^{(p)} + b_j)$ can't perform well on the classification problem defined in Definition 3.1 where $\sigma$ can be any activation function. Theorem 4.1 also shows that a simple ensemble of the experts may not improve the performance because the ensemble of the two-layer CNNs is still in the form of the function defined in Theorem 4.1.

As a comparison, the following theorem gives the learning guarantees for training an MoE layer that follows the structure defined in Section 3.2 with cubic activation function.

**Theorem 4.2** (Nonlinear MoE performs well)**.** Suppose the training data size $n = \Omega(d)$. Choose experts number $M = \Theta(K\log K\log\log d)$, filter size $J = \Theta(\log M\log\log d)$, initialization scale $\sigma_0\in[d^{-1/3}, d^{-0.01}]$, learning rate $\eta = \widetilde{O}(\sigma_0), \eta_r = \Theta(M^2)\eta$. Then with probability at least $1-o(1)$, Algorithm 1 is able to output $(\mathbf{\Theta}^{(T)},\mathbf{W}^{(T)})$ within $T = \widetilde{O}(\eta^{-1})$ iterations such that the non-linear MoE defined in Section 3.2 satisfies that

- Training error is zero, i.e., $y_iF(\mathbf{x}_i;\mathbf{\Theta}^{(T)},\mathbf{W}^{(T)}) > 0, \forall i\in[n]$.
- Test error is nearly zero, i.e., $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yF(\mathbf{x};\mathbf{\Theta}^{(T)},\mathbf{W}^{(T)})\le 0\big) = o(1)$.

More importantly, the experts can be divided into a disjoint union of $K$ non-empty sets $[M] = \sqcup_{k\in[K]}\mathcal{M}_k$ and

- (Each expert is good on one cluster) Each expert $m\in\mathcal{M}_k$ performs good on the cluster $\Omega_k$, $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}(yf_m(\mathbf{x};\mathbf{W}^{(T)})\le 0|(\mathbf{x},y)\in\Omega_k) = o(1)$.
- (Router only distributes example to good expert) With probability at least $1-o(1)$, an example $\mathbf{x}\in\Omega_k$ will be routed to one of the experts in $\mathcal{M}_k$.

Theorem 4.2 shows that a non-linear MoE performs well on the classification problem in Definition 3.1. In addition, the router will learn the cluster structure and divide the problem into $K$ simpler subproblems, each of which is associated with one cluster. In particular, each cluster will be classified accurately by a subset of experts. On the other hand, each expert will perform well on at least one cluster.

Furthermore, together with Theorem 4.1, Theorem 4.2 suggests that there exist problem instances in Definition 3.1 (i.e., $\mathcal{D}_\alpha = \mathcal{D}_\gamma$) such that an MoE provably outperforms a single expert.

## 5 Overview of Key Techniques

A successful MoE layer needs to ensure that the router can learn the cluster-center features and divide the complex problem in Definition 3.1 into simpler linear classification sub-problems that individual experts can conquer. Finding such a gating network is difficult because this problem is highly non-convex. In the following, we will introduce the main difficulties in analyzing the MoE layer and the corresponding key techniques to overcome those barriers.

**Main Difficulty 1: Discontinuities in Routing.** Compared with the traditional soft-routing model, the sparse routing model saves computation and greatly reduces the inference time. However, this form of sparsity also causes discontinuities in routing (Shazeer et al., 2017). In fact, even a small perturbation of the gating network outputs $\mathbf{h}(\mathbf{x};\mathbf{\Theta})+\boldsymbol{\delta}$ may change the router behavior drastically if the second largest gating network output is close to the largest gating network output.

**Key Technique 1: Stability by Smoothing.** We point out that the noise term added to the gating network output ensures a smooth transition between different routing behavior, which makes the router more stable. This is proved in the following lemma.

**Lemma 5.1.** Let $\mathbf{h},\widehat{\mathbf{h}}\in\mathbb{R}^M$ to be the output of the gating network and $\{r_m\}_{m=1}^M$ to be the noise independently drawn from Unif[0,1]. Denote $\mathbf{p},\widehat{\mathbf{p}}\in\mathbb{R}^M$ to be the probability that experts get routed, i.e., $p_m=\mathbb{P}(\mathrm{argmax}_{m'\in[M]}\{h_{m'}+r_{m'}\}=m)$, $\widehat{p}_m=\mathbb{P}(\mathrm{argmax}_{m'\in[M]}\{\widehat{h}_{m'}+r_{m'}\}=m)$. Then we have that $\|\mathbf{p}-\widehat{\mathbf{p}}\|_\infty\le M^2\|\mathbf{h}-\widehat{\mathbf{h}}\|_\infty$.

Lemma 5.1 implies that when the change of the gating network outputs at iteration $t$ and $t'$ is small, i.e., $\|\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(t)})-\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(t')})\|_\infty$, the router behavior will be similar. So adding noise provides a smooth transition from time $t$ to $t'$. It is also worth noting that $\mathbf{\Theta}$ is zero initialized. So $\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(0)})=0$ and thus each expert gets routed with the same probability $p_m=1/M$ by symmetric property. Therefore, at the early of the training when $\|\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(t)})-\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(0)})\|_\infty$ is small, router will almost uniformly pick one expert from $[M]$, which helps exploration across experts.

**Main Difficulty 2: No "Real" Expert.** At the beginning of the training, the gating network is zero, and the experts are randomly initialized. Thus it is hard for the router to learn the right features because all the experts look the same: they share the same network architecture and are trained by the same algorithm. The only difference is the initialization. Moreover, if the router makes a mistake at the beginning of the training, the experts may amplify the mistake because the experts will be trained based on mistakenly dispatched data.

**Key Technique 2: Experts from Exploration.** Motivated by the key technique 1, we introduce an exploration stage to the analysis of MoE layer during which the router almost uniformly picks one expert from $[M]$. This stage starts at $t=0$ and ends at $T_1=\lfloor\eta^{-1}\sigma_0^{0.5}\rfloor\ll T=\widetilde{O}(\eta^{-1})$ and the gating network remains nearly unchanged $\|\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(t)})-\mathbf{h}(\mathbf{x};\mathbf{\Theta}^{(0)})\|_\infty=O(\sigma_0^{1.5})$. Because the experts are treated almost equally during exploration stage, we can show that the experts become specialized to some specific task only based on the initialization. In particular, the experts set $[M]$ can be divided into $K$ nonempty disjoint sets $[M]=\sqcup_k\mathcal{M}_k$, where $\mathcal{M}_k:=\{m|\,\mathrm{argmax}_{k'\in[K],j\in[J]}\langle\mathbf{v}_{k'},\mathbf{w}_{m,j}^{(0)}\rangle=k\}$. For nonlinear MoE with cubic activation function, the following lemma further shows that experts in different set $\mathcal{M}_k$ will diverge at the end of the exploration stage.

**Lemma 5.2.** Under the same condition as in Theorem 4.2, with probability at least $1-o(1)$, the following equations hold for all expert $m\in\mathcal{M}_k$,

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T_1)})\le 0\big|(\mathbf{x},y)\in\Omega_k\big)=o(1),$$
$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T_1)})\le 0\big|(\mathbf{x},y)\in\Omega_{k'}\big)=\Omega\big(1/K\big),\forall k'\ne k.$$

Lemma 5.2 implies that, at the end of the exploration stage, the expert $m\in\mathcal{M}_k$ can achieve nearly zero test error on the cluster $\Omega_k$ but high test error on the other clusters $\Omega_{k'},k'\ne k$.

**Main Difficulty 3: Expert Load Imbalance.** Given the training data set $S=\{(\mathbf{x}_i,y_i)\}_{i=1}^n$, the load of expert $m$ at iterate $t$ is defined as

$$\mathrm{Load}_m^{(t)}=\textstyle\sum_{i\in[n]}\mathbb{P}(m_{i,t}=m),\tag{5.1}$$

where $\mathbb{P}(m_{i,t}=m)$ is probability that the input $\mathbf{x}_i$ being routed to expert $m$ at iteration $t$. Eigen et al. (2013) first described the load imbalance issues in the training of the MoE layer. The gating

network may converge to a state where it always produces large $\text{Load}_m^{(t)}$ for the same few experts. This imbalance in expert load is self-reinforcing, as the favored experts are trained more rapidly and thus are selected even more frequently by the router (Shazeer et al., 2017; Fedus et al., 2021). Expert load imbalance issue not only causes memory and performance problems in practice, but also impedes the theoretical analysis of the expert training.

**Key Technique 3: Normalized Gradient Descent.** Lemma 5.2 shows that the experts will diverge into $\sqcup_{k\in[K]}\mathcal{M}_k$. Normalized gradient descent can help different experts in the same $\mathcal{M}_k$ being trained at the same speed regardless of the imbalance load caused by the router. Because the self-reinforcing circle no longer exists, the load imbalance issue will get mitigated. In particular, the router will treat different experts in the same $\mathcal{M}_k$ almost equally and dispatch almost the same amount of data to them during the early stage of training (See Section E.2 in Appendix for detail), which is enough for the router to learn the cluster-center features. However, we can't guarantee load balance for an arbitrary long training period if we only use normalized gradient descent. That's the reason Theorem 4.2 requires early stopping. This load imbalance issue can be further avoided by adding load balancing loss (Eigen et al., 2013; Shazeer et al., 2017; Fedus et al., 2021), or using advanced MoE layer structure such as BASE Layers (Lewis et al., 2021; Dua et al., 2021) and Hash Layers (Roller et al., 2021).

**Road Map:** Here we provide the road map of the proof of Theorem 4.2 and the full proof is presented in Appendix E. The training process can be decomposed into several stages. The first stage is called *Exploration stage*. During this stage, the experts will diverge into $K$ professional groups $\sqcup_{k=1}^{K}\mathcal{M}_k = [M]$. In particular, we will show that $\mathcal{M}_k$ is not empty for all $k \in [K]$. Besides, for all $m \in \mathcal{M}_k$, $f_m$ is a good classifier over $\Omega_k$. The second stage is called *router learning stage*. During this stage, the router will learn to dispatch $\mathbf{x} \in \Omega_k$ to one of the experts in $\mathcal{M}_k$. Finally, we will give the generalization analysis for the MoEs from the previous two stages.

## 6 Experiments

In this section, we conduct experiments to validate our theory. The code and data for our experiments can be found on Github [1].

Setting 1:$\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 1$

| | Test accuracy (%) | Dispatch Entropy |
|---|---|---|
| Single (linear) | 68.71 | NA |
| Single (nonlinear) | 79.48 | NA |
| MoE (linear) | $92.99 \pm 2.11$ | $1.300 \pm 0.044$ |
| MoE (nonlinear) | $\mathbf{99.46 \pm 0.55}$ | $\mathbf{0.098 \pm 0.087}$ |

Setting 2: $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 2$

| | Test accuracy (%) | Dispatch Entropy |
|---|---|---|
| Single (linear) | 60.59 | NA |
| Single (nonlinear) | 72.29 | NA |
| MoE (linear) | $88.48 \pm 1.96$ | $1.294 \pm 0.036$ |
| MoE (nonlinear) | $\mathbf{98.09 \pm 1.27}$ | $\mathbf{0.171 \pm 0.103}$ |

**Table 1: Comparison between MoE (linear) and MoE (nonlinear)** in our setting. We report results of top-1 gating with noise for both linear and nonlinear models. Over ten random experiments, we report the average value $\pm$ standard deviation for both test accuracy and dispatch entropy.

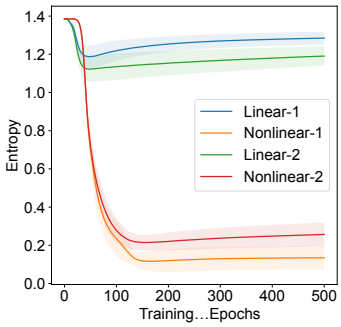


**Figure 3: Illustration of router dispatch entropy.** We demonstrate the change of entropy of MoE during training on the synthetic data. MoE (linear)-1 and MoE (nonlinear)-1 refer to Setting 1 in Table 1. MoE (linear)-2 and MoE (nonlinear)-2 refer to Setting 2 in Table 1.

### 6.1 Synthetic-data Experiments

**Datasets.** We generate $16,000$ training examples and $16,000$ test examples from the data distribution defined in Definition 3.1 with cluster number $K = 4$ , patch number $P = 4$ and dimension $d = 50$. We randomly shuffle the order of the patches of $\mathbf{x}$ after we generate data $(\mathbf{x}, y)$. We consider two

[1] https://github.com/uclaml/MoE

**Table 2:** Comparison between MoE and single model on CIFAR-10 and CIFAR-10-Rotate datasets. We report the average test accuracy over 10 random experiments $\pm$ the standard deviation.

| | | CIFAR-10 (%) | CIFAR-10-Rotate (%) |
|---|---|---|---|
| CNN | Single | $80.68 \pm 0.45$ | $76.78 \pm 1.79$ |
| | MoE | $80.31 \pm 0.62$ | $\mathbf{79.60 \pm 1.25}$ |
| MobileNetV2 | Single | $92.45 \pm 0.25$ | $85.76 \pm 2.91$ |
| | MoE | $92.23 \pm 0.72$ | $\mathbf{89.85 \pm 2.54}$ |
| ResNet18 | Single | $95.51 \pm 0.31$ | $88.23 \pm 0.96$ |
| | MoE | $95.32 \pm 0.68$ | $\mathbf{92.60 \pm 2.01}$ |

parameter settings: 1. $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 3)$ and $\sigma_p = 1$; 2. $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 3)$ and $\sigma_p = 2$. Note that Theorem 4.1 shows that when $\alpha$ and $\gamma$ follow the same distribution, neither single linear expert or single nonlinear expert can give good performance. Here we consider a more general and difficult setting when $\alpha$ and $\gamma$ are from different distributions.

**Models.** We consider the performances of single linear CNN, single nonlinear CNN, linear MoE, and nonlinear MoE. The single nonlinear CNN architecture follows (3.1) with cubic activation function, while single linear CNN follows (3.1) with identity activation function. For both linear and nonlinear MoEs, we consider a mixture of 8 experts with each expert being a single linear CNN or a single nonlinear CNN. Finally, we train single models with gradient descent and train the MoEs with Algorithm 1. We run 10 random experiments and report the average accuracy with standard deviation.

**Evaluation.** To evaluate how well the router learned the underlying cluster structure of the data, we define the entropy of the router's dispatch as follows. Denote by $n_{k,m}$ the number of data in cluster $K$ that are dispatched to expert $m$. The total number of data dispatched to expert $m$ is $n_m = \sum_{k=1}^{K} n_{k,m}$ and the total number of data is $n = \sum_{k=1}^{K} \sum_{m=1}^{M} n_{k,m}$. The dispatch entropy is then defined as

$$\text{entropy} = -\sum_{m=1, n_m \neq 0}^{M} \frac{n_m}{n} \sum_{k=1}^{K} \frac{n_{k,m}}{n_m} \cdot \log\left(\frac{n_{k,m}}{n_m}\right). \tag{6.1}$$

When each expert receives the data from at most one cluster, the dispatch entropy will be zero. And a uniform dispatch will result in the maximum dispatch entropy.

As shown in Table 1, the linear MoE does not perform as well as the nonlinear MoE in Setting 1, with around 6% less test accuracy and much higher variance. With stronger random noise (Setting 2), the difference between the nonlinear MoE and linear MoE becomes even more significant. We also observe that the final dispatch entropy of nonlinear MoE is nearly zero while that of the linear MoE is large. In Figure 3, we further demonstrate the change of dispatch entropy during the training process. The dispatch entropy of nonlinear MoE significantly decreases, while that of linear MoE remains large. Such a phenomenon indicates that the nonlinear MoE can successfully learn the underlying cluster structure of the data while the linear MoE fails to do so.

### 6.2 Real-data Experiments

We further conduct experiments on real image datasets and demonstrate the importance of the clustering data structure to the MoE layer in deep neural networks.

**Datasets.** We consider the **CIFAR-10** dataset (Krizhevsky, 2009) and the 10-class classification task. Furthermore, we create a **CIFAR-10-Rotate** dataset that has a strong underlying cluster structure that is independent of its labeling function. Specifically, we rotate the images by 30 degrees and merge the rotated dataset with the original one. The task is to predict if the image is rotated, which is a binary classification problem. We deem that some of the classes in CIFAR-10 form underlying clusters in CIFAR-10-Rotate. In Appendix A, we explain in detail how we generate CIFAR-10-Rotate and present some specific examples.

**Models.** For the MoE, we consider a mixture of 4 experts with a linear gating network. For the expert/single model architectures, we consider a CNN with 2 convolutional layers (architecture details are illustrated in Appendix A.) For a more thorough evaluation, we also consider expert/single models

with architecture including **MobileNetV2** (Sandler et al., 2018) and **ResNet18** (He et al., 2016). The training process of MoE also follows Algorithm 1.

The experiment results are shown in Table 2, where we compare single and mixture models of different architectures over CIFAR-10 and CIFAR-10-Rotate datasets. We observe that the improvement of MoEs over single models differs largely on the different datasets. On CIFAR-10, the performance of MoEs is very close to the single models. However, on the CIFAR-10-Rotate dataset, we can observe a significant performance improvement from single models to MoEs. Such results indicate the advantage of MoE over single models depends on the task and the cluster structure of the data.

**Visualization.** In Figure 4, we visualize the latent embedding learned by MoEs (ResNet18) for the 10-class classification task in CIFAR-10 as well as the binary classification task in CIFAR-10-Rotate. We visualize the data with the same label $y$ to see if cluster structures exist within each class. For CIFAR-10, we choose $y = 1$ ("car"), and plot the latent embedding of the data using t-SNE on the left sub-figure, which does not show an salient cluster structure. For CIFAR-10-Rotate, we choose $y = 1$ ("rotated") and visualize the data using t-SNE in the middle sub-figure. Here, we can observe a clear clustering structure even though the class signal is not provided during training. We take a step further to investigate what is in each cluster in the right sub-figure. We can observe that most of the examples in the "frog" class fall into one cluster, while examples of "ship" class mostly fall into the other cluster.

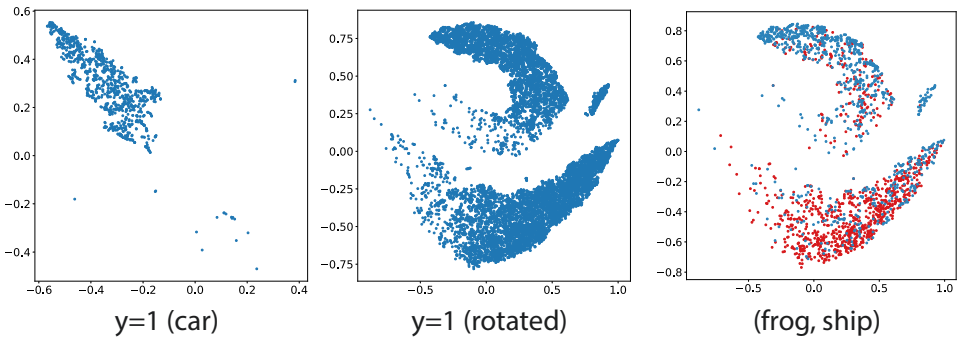


**Figure 4:** Visualization of the latent embedding on CIFAR-10 and CIFAR-10-Rotate with chosen label $y$. The left sub-figure denotes the visualization of CIFAR-10 when label $y$ is chosen to be 1 (car). The central sub-figure represents the visualization of CIFAR-10-Rotate when label $y$ is chosen to be 1 (rotated). On the right sub-figure, red denotes that the data is from the ship class, and blue denotes that the data is from the frog class.

## 7 Conclusion and Future Work

In this work, we formally study the mechanism of the Mixture of Experts (MoE) layer for deep learning. To our knowledge, we provide the first theoretical result toward understanding how the MoE layer works in deep learning. Our empirical evidence reveals that the cluster structure of the data plays an important role in the success of the MoE layer. Motivated by these empirical observations, we study a data distribution with cluster structure and show that Mixture-of-Experts provably improves the test accuracy of a single expert of two-layer CNNs.

There are several important future directions. First, our current results are for CNNs. It is interesting to extend our results to other neural network architectures, such as transformers. Second, our data distribution is motivated by the classification problem of image data. We plan to extend our analysis to other types of data (e.g., natural language data).

## Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and area chair for their helpful comments. ZC, YD, YW and QG are supported in part by the National Science Foundation CAREER Award 1906169, BIGDATA IIS-1855099, IIS-2008981, and the Sloan Research Fellowship. YL is supported in part by the NSF RI2007517. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

## References


ALLEN-ZHU, Z. and LI, Y. (2019). What can ResNet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*.

ALLEN-ZHU, Z. and LI, Y. (2020a). Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413* .

ALLEN-ZHU, Z. and LI, Y. (2020b). Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190* .

ALLEN-ZHU, Z. and LI, Y. (2020c). Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816* .

ALLEN-ZHU, Z., LI, Y. and LIANG, Y. (2019a). Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*.

ALLEN-ZHU, Z., LI, Y. and SONG, Z. (2019b). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*.

ANANDKUMAR, A., GE, R., HSU, D., KAKADE, S. M. and TELGARSKY, M. (2014). Tensor decompositions for learning latent variable models. *Journal of machine learning research* **15** 2773–2832.

ANANDKUMAR, A., HSU, D. and KAKADE, S. M. (2012). A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings.

ARORA, S., DU, S., HU, W., LI, Z. and WANG, R. (2019a). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*.

ARORA, S., DU, S. S., HU, W., LI, Z., SALAKHUTDINOV, R. and WANG, R. (2019b). On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*.

BAI, Y. and LEE, J. D. (2019). Beyond linearization: On quadratic and higher-order approximation of wide neural networks. *arXiv preprint arXiv:1910.01619* .

BALAKRISHNAN, S., WAINWRIGHT, M. J. and YU, B. (2017). Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics* **45** 77–120.

BLARD, T. (2020). French sentiment analysis with bert. `https://github.com/TheophileBlard/french-sentiment-analysis-with-bert`.

CAO, Y., CHEN, Z., BELKIN, M. and GU, Q. (2022). Benign overfitting in two-layer convolutional neural networks. *arXiv preprint arXiv:2202.06526* .

CAO, Y. and GU, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*.

CHAGANTY, A. T. and LIANG, P. (2013). Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*. PMLR.

COLLOBERT, R., BENGIO, S. and BENGIO, Y. (2002). A parallel mixture of svms for very large scale problems. *Neural computation* **14** 1105–1114.

DAUPHIN, Y. N., FAN, A., AULI, M. and GRANGIER, D. (2017). Language modeling with gated convolutional networks. In *International conference on machine learning*. PMLR.

DE VEAUX, R. D. (1989). Mixtures of linear regressions. *Computational Statistics & Data Analysis* **8** 227–245.

DEVLIN, J., CHANG, M., LEE, K. and TOUTANOVA, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* **abs/1810.04805**.

DU, S. S., ZHAI, X., POCZOS, B. and SINGH, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*.

DUA, D., BHOSALE, S., GOSWAMI, V., CROSS, J., LEWIS, M. and FAN, A. (2021). Tricks for training sparse translation models. *arXiv preprint arXiv:2110.08246* .

EIGEN, D., RANZATO, M. and SUTSKEVER, I. (2013). Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314* .

FARIA, S. and SOROMENHO, G. (2010). Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation* **80** 201–225.

FEDUS, W., ZOPH, B. and SHAZEER, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961* .

GO, A., BHAYANI, R. and HUANG, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford* **1** 2009.

HE, K., ZHANG, X., REN, S. and SUN, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

HSU, D. J., KAKADE, S. M. and LIANG, P. S. (2012). Identifiability and unmixing of latent parse trees. *Advances in neural information processing systems* **25**.

JACOBS, R. A., JORDAN, M. I., NOWLAN, S. J. and HINTON, G. E. (1991). Adaptive mixtures of local experts. *Neural computation* **3** 79–87.

JACOT, A., GABRIEL, F. and HONGLER, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*.

JELASSI, S., MENSCH, A., GIDEL, G. and LI, Y. (2021). Adam is no better than normalized sgd: Dissecting how adaptivity improves gan performance .

JORDAN, M. I., GHAHRAMANI, Z. and SAUL, L. K. (1997). Hidden markov decision trees. *Advances in neural information processing systems* 501–507.

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

KHALILI, A. and CHEN, J. (2007). Variable selection in finite mixture of regression models. *Journal of the american Statistical association* **102** 1025–1038.

KINGMA, D. P. and BA, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

KRIZHEVSKY, A. (2009). Learning multiple layers of features from tiny images. Tech. rep.

LEWIS, M., BHOSALE, S., DETTMERS, T., GOYAL, N. and ZETTLEMOYER, L. (2021). Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*. PMLR.

LI, Y. and LIANG, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*.

LI, Y., MA, T. and ZHANG, H. R. (2020). Learning over-parametrized two-layer neural networks beyond ntk. In *Conference on learning theory*. PMLR.

LIANG, P., BOUCHARD-CÔTÉ, A., KLEIN, D. and TASKAR, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

QUATTONI, A., COLLINS, M. and DARRELL, T. (2004). Conditional random fields for object recognition. *Advances in neural information processing systems* **17**.

ROLLER, S., SUKHBAATAR, S., WESTON, J. ET AL. (2021). Hash layers for large sparse models. *Advances in Neural Information Processing Systems* **34** 17555–17566.

SANDLER, M., HOWARD, A., ZHU, M., ZHMOGINOV, A. and CHEN, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

SHAZEER, N., MIRHOSEINI, A., MAZIARZ, K., DAVIS, A., LE, Q., HINTON, G. and DEAN, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* .

SMETANIN, S. and KOMAROV, M. (2019). Sentiment analysis of product reviews in russian using convolutional neural networks. In *2019 IEEE 21st conference on business informatics (CBI)*, vol. 1. IEEE.

TRESP, V. (2001). Mixtures of gaussian processes. *Advances in neural information processing systems* 654–660.

VAN DER MAATEN, L. and HINTON, G. (2008). Visualizing data using t-sne. *Journal of machine learning research* **9**.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł. and POLOSUKHIN, I. (2017). Attention is all you need. In *Advances in neural information processing systems*.

VECCI, L., PIAZZA, F. and UNCINI, A. (1998). Learning and approximation capabilities of adaptive spline activation function neural networks. *Neural Networks* **11** 259–270.

WANG, Y. and MORI, G. (2009). Max-margin hidden conditional random fields for human action recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

WANG, Z., GU, Q., NING, Y. and LIU, H. (2015). High dimensional em algorithm: Statistical optimization and asymptotic normality. *Advances in neural information processing systems* **28**.

WEN, Z. and LI, Y. (2021). Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*. PMLR.

YI, X., CARAMANIS, C. and SANGHAVI, S. (2014). Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*. PMLR.

ZOU, D., CAO, Y., LI, Y. and GU, Q. (2021). Understanding the generalization of adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371* .

ZOU, D., CAO, Y., ZHOU, D. and GU, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888* .

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] This paper gives the first theoretical result toward formally understanding the mechanism of the MoE layer for deep learning.
   (b) Did you describe the limitations of your work? [Yes] We note in Section 5 that our analysis of the MoE layer need early stopping which we believe can be waived by adding some well-signed regularization. We will explore this in future work.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We seek to mathematically understand the MoE layer in Deep Learning, it is not clear what potential negative impacts a deeper theoretical understanding of this algorithm would bring.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See our Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

# A Experiment Details

## A.1 Visualization

In the visualization of Figure 1, MoE (linear) and MoE (nonlinear) are trained according to Algorithm 1 by normalized gradient descent with learning rate $0.001$ and gradient descent with learning rate $0.1$. According to Definition 3.1, we set $K = 4$, $P = 4$ and $d = 50$ and choose $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (1, 2)$ and $\sigma_p = 1$, and generate $3,200$ data examples. We consider mixture of $M = 4$ experts for both MoE (linear) and MoE (nonlinear). For each expert, we set the number of neurons/filters $J = 16$. We train MoEs on $1,600$ data examples and visualize classification result and decision boundary on the remaining $1,600$ examples. The data examples are visualized via t-SNE (Van der Maaten and Hinton, 2008). When visualizing the data points and decision boundary on the 2d space, we increase the magnitude of random noise patch by 3 so that the positive/negative examples and decision boundaries can be better viewed.

## A.2 Synthetic-data Experiments

**Synthetic-data experiment setup.** For the experiments on synthetic data, we generate the data according to Definition 3.1 with $K = 4$, $P = 4$ and $d = 50$. We consider four parameter settings:

- $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 3)$ and $\sigma_p = 1$;
- $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 3)$ and $\sigma_p = 2$;
- $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 2)$ and $\sigma_p = 1$;
- $\alpha \sim \text{Uniform}(0.5, 2)$, $\beta \sim \text{Uniform}(1, 2)$, $\gamma \sim \text{Uniform}(0.5, 2)$ and $\sigma_p = 2$.

We consider mixture of $M = 8$ experts for all MoEs and $J = 16$ neurons/filters for all experts. For single models, we consider $J = 128$ neurons/filters. We train MoEs using Algorithm 1. Specifically, we train the experts by normalized gradient descent with learning rate $0.001$ and the gating network by gradient descent with learning rate $0.1$. We train single linear/nonlinear models by Adam (Kingma and Ba, 2014) to achieve the best performance, with learning rate $0.01$ and weight decay 5e-4 for single nonlinear model and learning rate $0.003$ and weight decay $5e-4$ for single linear model.

**Synthetic-data experiment results.** In Table 3, we present the empirical results of single linear CNN, single nonlinear CNN, linear MoE, and nonlinear MoE under settings 3 and 4, where $\alpha$ and $\gamma$ follow the same distribution as we assumed in theoretical analysis. Furthermore, we report the total number of filters for both single CNNs and a mixture of CNNs, where the filter size (equal to $50$) is the same for all single models and experts. For linear and nonlinear MoE, there are $16$ filters for each of the $8$ experts, and therefore $128$ filters in total. Note that in the synthetic-data experiment in the main paper, we let the number of filters of single models be the same as MoEs ($128$). Here, we additionally report the performances of single models with $512$ filters, and see if increasing the model size of single models can beat MoE. From Table 3, we observe that: 1. single models perform poorly in all settings; 2. linear MoEs do not perform as well as nonlinear MoEs. Specifically, the final dispatch entropy of nonlinear MoEs is nearly zero while the dispatch entropy of linear MoEs is consistently larger under settings 1-4. This indicates that nonlinear MoEs successfully uncover the underlying cluster structure while linear MoEs fail to do so. In addition, we can see that even larger single models cannot beat linear MoEs or nonlinear MoEs. This is consistent with Theorem $4.1$, where a single model fails under such data distribution regardless of its model size. Notably, by comparing the results in Table 1 and Table 3, we can see that a single nonlinear model suffers from overfitting as we increase the number of filters.

**Router dispatch examples.** We demonstrate specific examples of router dispatch for MoE (nonlinear) and MoE (linear). The examples of initial and final router dispatch for MoE (nonlinear) are shown in Table 4 and Table 5. Under the dispatch for nonlinear MoE, each expert is given either no data or data that comes from one cluster only. The entropy of such dispatch is thus $0$. The test accuracy of MoE trained under such a dispatch is either $100\%$ or very close to $100\%$, as the expert can be easily trained on the data from one cluster only. An example of the final dispatch for MoE (linear) is shown in Table 6, where clusters are not well separated and an expert gets data from different clusters. The test accuracy under such dispatch is lower ($90.61\%$).

**Table 3: Comparison between MoE (linear) and MoE (nonlinear)** in our setting. We report results of top-1 gating with noise for both linear and nonlinear models. Over ten random experiments, we report the average value $\pm$ standard deviation for both test accuracy and dispatch entropy.

Setting 1:$\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 1$

| | Test accuracy (%) | Dispatch Entropy | Number of Filters |
|---|---|---|---|
| Single (linear) | 68.71 | NA | 128 |
| Single (linear) | 67.63 | NA | 512 |
| Single (nonlinear) | 79.48 | NA | 128 |
| Single (nonlinear) | 78.18 | NA | 512 |
| MoE (linear) | $92.99 \pm 2.11$ | $1.300 \pm 0.044$ | 128 (16*8) |
| MoE (nonlinear) | $\mathbf{99.46 \pm 0.55}$ | $\mathbf{0.098 \pm 0.087}$ | 128 (16*8) |

Setting 2: $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 2$

| | Test accuracy (%) | Dispatch Entropy | Number of Filters |
|---|---|---|---|
| Single (linear) | 60.59 | NA | 128 |
| Single (linear) | 63.04 | NA | 512 |
| Single (nonlinear) | 72.29 | NA | 128 |
| Single (nonlinear) | 52.09 | NA | 512 |
| MoE (linear) | $88.48 \pm 1.96$ | $1.294 \pm 0.036$ | 128 (16*8) |
| MoE (nonlinear) | $\mathbf{98.09 \pm 1.27}$ | $\mathbf{0.171 \pm 0.103}$ | 128 (16*8) |

Setting 3:$\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 2)$, $\sigma_p = 1$

| | Test accuracy (%) | Dispatch Entropy | Number of Filters |
|---|---|---|---|
| Single (linear) | 74.81 | NA | 128 |
| Single (linear) | 74.54 | NA | 512 |
| Single (nonlinear) | 72.69 | NA | 128 |
| Single (nonlinear) | 67.78 | NA | 512 |
| MoE (linear) | $95.93 \pm 1.34$ | $1.160 \pm 0.100$ | 128 (16*8) |
| MoE (nonlinear) | $\mathbf{99.99 \pm 0.02}$ | $\mathbf{0.008 \pm 0.011}$ | 128 (16*8) |

Setting 4: $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 2)$, $\sigma_p = 2$

| | Test accuracy (%) | Dispatch Entropy | Number of Filters |
|---|---|---|---|
| Single (linear) | 74.63 | NA | 128 |
| Single (linear) | 72.98 | NA | 512 |
| Single (nonlinear) | 68.60 | NA | 128 |
| Single (nonlinear) | 61.65 | NA | 512 |
| MoE (linear) | $93.30 \pm 1.48$ | $1.160 \pm 0.155$ | 128 (16*8) |
| MoE (nonlinear) | $\mathbf{98.92 \pm 1.18}$ | $\mathbf{0.089 \pm 0.120}$ | 128 (16*8) |

**Table 4:** Dispatch details of MoE (nonlinear) with test accuracy 100%.

| Expert number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Initial dispatch | 1921 | 2032 | 1963 | 1969 | 2075 | 1980 | 2027 | 2033 |
| Final dispatch | 0 | 3979 | 4009 | 0 | 0 | 3971 | 0 | 4041 |
| Cluster 1 | 0 | 0 | 0 | 0 | 0 | 3971 | 0 | 0 |
| Cluster 2 | 0 | 0 | 4009 | 0 | 0 | 0 | 0 | 0 |
| Cluster 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4041 |
| Cluster 4 | 0 | 3979 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 5:** Dispatch details of MoE (nonlinear) with test accuracy $99.95\%$.

| Expert number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Initial dispatch | 1978 | 2028 | 2018 | 1968 | 2000 | 2046 | 2000 | 1962 |
| Final dispatch | 3987 | 4 | 3975 | 6 | 0 | 1308 | 4009 | 2711 |
| Cluster 1 | 0 | 0 | 3971 | 0 | 0 | 0 | 0 | 0 |
| Cluster 2 | 0 | 0 | 0 | 0 | 0 | 4 | 4005 | 0 |
| Cluster 3 | 8 | 4 | 4 | 6 | 0 | 1304 | 4 | 2711 |
| Cluster 4 | 3979 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 6:** Dispatch details of MoE (linear) with test accuracy $90.61\%$.

| Expert number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Initial dispatch | 1969 | 2037 | 1983 | 2007 | 1949 | 1905 | 2053 | 2097 |
| Final dispatch | 136 | 2708 | 6969 | 5311 | 27 | 87 | 4 | 758 |
| Cluster 1 | 0 | 630 | 1629 | 1298 | 27 | 87 | 4 | 296 |
| Cluster 2 | 136 | 1107 | 1884 | 651 | 0 | 0 | 0 | 231 |
| Cluster 3 | 0 | 594 | 1976 | 1471 | 0 | 0 | 0 | 0 |
| Cluster 4 | 0 | 377 | 1480 | 1891 | 0 | 0 | 0 | 231 |

**MoE during training.** We further provide figures that illustrate the feature learning and center learning process of each expert $\mathbf{W}_m = [\mathbf{w}_{m,1}, \ldots, \mathbf{w}_{m,J}]$ and the router $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$, with $J$ as the number of filters/neurons and $M$ as the number of experts. We observe the feature learning process (change of $\max_j \langle \mathbf{w}_{m,j}, \mathbf{v}_k \rangle$) and center learning process (change of $\max_j \langle \mathbf{w}_{m,j}, \mathbf{c}_k \rangle$) of each expert $\mathbf{w}_m$ for each feature signal $\mathbf{v}_k$ and center signal $\mathbf{c}_k$. Similarly, for the weight of the router $\boldsymbol{\theta}_m$, we observe the the feature learning process (change of $\langle \boldsymbol{\theta}_m, \mathbf{v}_k \rangle$) and center learning process (change of $\langle \boldsymbol{\theta}_m, \mathbf{c}_k \rangle$) for each feature signal $\mathbf{v}_k$ and center signal $\mathbf{c}_k$. In Figure 5, we demonstrate the training process of MoE (nonlinear), and in Figure 6, we demonstrate the training process of MoE (linear). Each colored line denotes a value of $k$. The data is the same as setting 1 in Table 1, with $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$ and $\sigma_p = 1$. We can observe that, in the top left sub-figure of Figure 5 for MoE (nonlinear), feature learning ($\max_j \langle \mathbf{w}_{m,j}, \mathbf{v}_k \rangle$) exhibit a property that each expert picks up one feature signal quickly. Similarly, as shown in the bottom right sub-figure, the router picks up the corresponding center signal. Meanwhile, the nonlinear experts almost do not learn center signals and the magnitude of the inner products between router weight and feature signals remain small. However, for MoE (linear), as shown in the top two sub-figures of Figure 6, an expert does not learn a specific feature signal, but instead learns multiple feature and center signals. Moreover, as demonstrated in the bottom sub-figures of Figure 6, the magnitude of the inner products between router weight and feature signals can be even larger than the inner products between router weight and center signals.

**Verification of Theorem 4.1.** In Table 7, we provide the performances of single models with different activation functions under setting 3, where $\alpha, \gamma \in (1, 2)$ follow the same distribution. In Table 8, we further report the performances of single models with different activation functions under setting 1 and setting 2. Empirically, even when $\alpha$ and $\gamma$ do not share the same distribution, single models still fail. Note that, for Tables 7 and 8, the numbers of filters for single models are 128.

**Load balancing loss.** In Table 9, we present the results of linear MoE with load balancing loss and directly compare it with nonlinear MoE without load balancing loss. Load balancing loss guarantees that the experts receive similar amount of data and prevents MoE from activating only one or few experts. However, on the data distribution that we study, load balancing loss is not the key to the success of MoE: the single experts cannot perform well on the entire data distribution and must diverge to learn different labeling functions with respect to each cluster.

**Initialization and Expert Divergence.** In Table 10, we consider nonlinear MoE with load balancing loss and the same initialization for all the experts. The synthetic data used in this experiments is the same as setting 1 in Table 3. Recall that the data distribution we study cannot be learned by any single model: experts must diverge to learn different labeling functions. We observe from Table 10

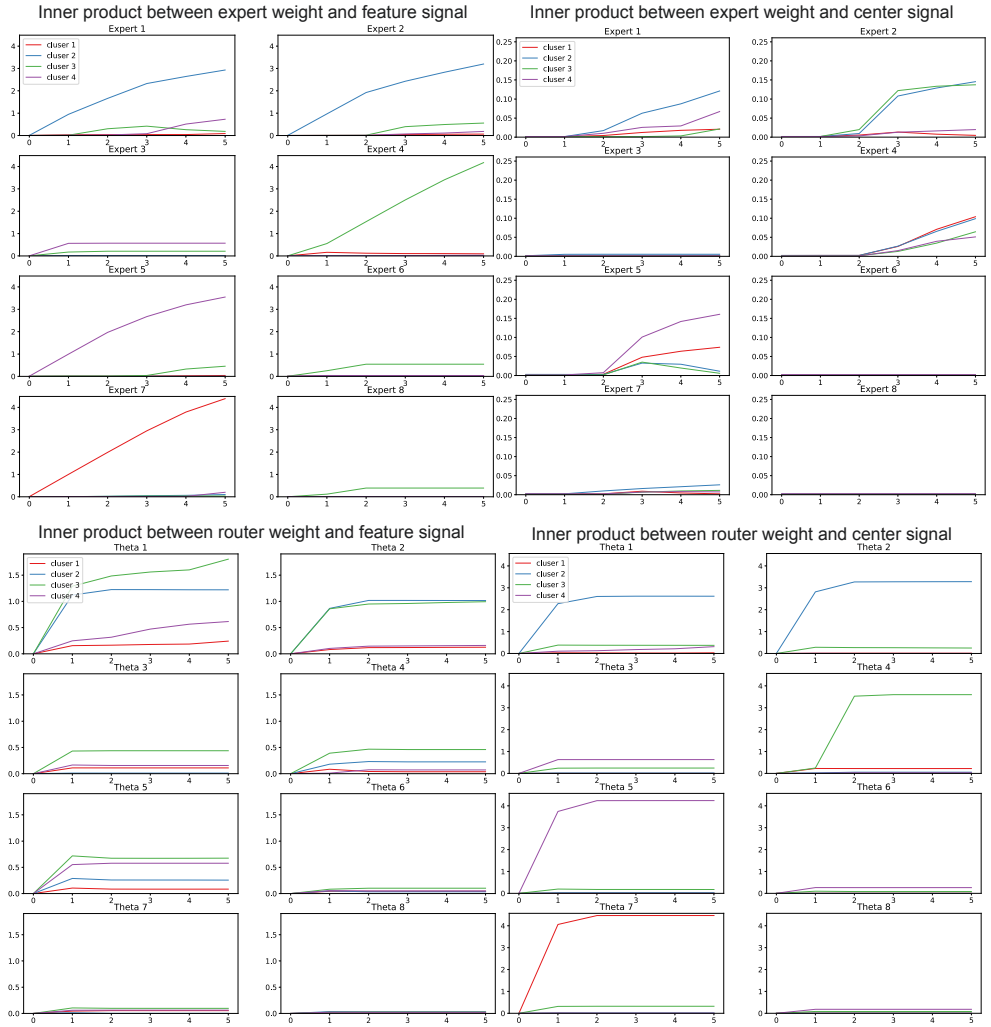


**Figure 5: Mixture of nonlinear experts.** Upper: visualization of the feature learning ($\max_j \langle \mathbf{w}_{m,j}, \mathbf{v}_k \rangle$) and center learning ($\max_j \langle \mathbf{w}_{m,j}, \mathbf{c}_k \rangle$) of each expert $\mathbf{w}_m$ for each feature $\mathbf{v}_k$ and cluster signal $\mathbf{c}_k$. Lower: visualization of the feature learning ($\langle \boldsymbol{\theta}_m, \mathbf{v}_k \rangle$) and center learning ($\langle \boldsymbol{\theta}_m, \mathbf{c}_k \rangle$) of the router weight $\boldsymbol{\theta}_m$ for each feature signal $\mathbf{v}_k$ and cluster signal $\mathbf{c}_k$.

**Table 7: Verification of Theorem 4.1 (single expert performs poorly)**. Test accuracy of single linear/nonlinear models with different activation functions. Data is generated according to Definition 3.1 with $\alpha, \gamma \in (1, 2)$, $\beta \in (1, 2)$ and $\sigma_p = 1$.

| Activation | Optimal Accuracy (%) | Test Accuracy (%) |
|---|---|---|
| Linear | 87.50% | 74.81% |
| Cubic | 87.50% | 72.69% |
| Relu | 87.50% | 73.45% |
| Celu | 87.50% | 76.91% |
| Gelu | 87.50% | 74.01% |
| Tanh | 87.50% | 74.76% |

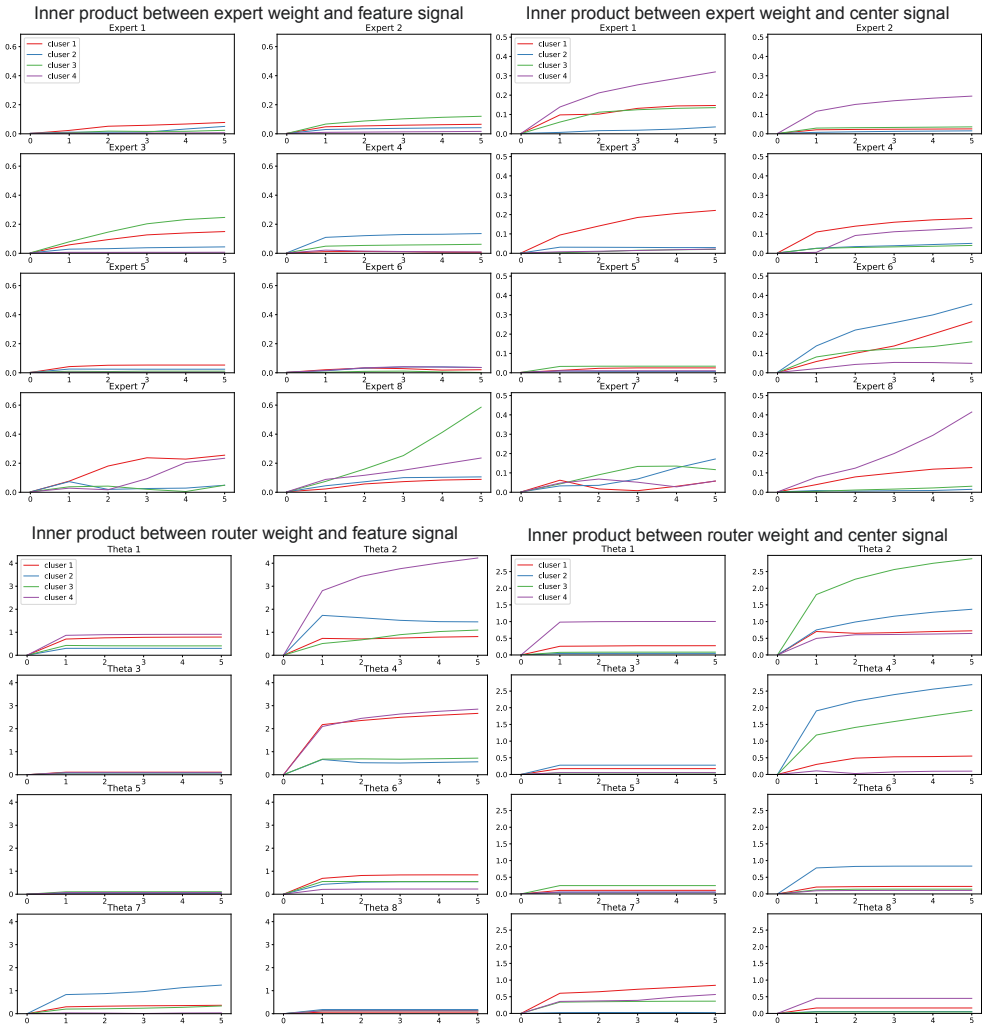


**Figure 6: Mixture of linear experts.** Upper: visualization of the feature learning ($\max_j \langle \mathbf{w}_{m,j}, \mathbf{v}_k \rangle$) and center learning ($\max_j \langle \mathbf{w}_{m,j}, \mathbf{c}_k \rangle$) of each expert $\mathbf{w}_m$ for each feature $\mathbf{v}_k$ and cluster signal $\mathbf{c}_k$. Lower: visualization of the feature learning ($\langle \boldsymbol{\theta}_m, \mathbf{v}_k \rangle$) and center learning ($\langle \boldsymbol{\theta}_m, \mathbf{c}_k \rangle$) of the router weight $\boldsymbol{\theta}_m$ for each feature signal $\mathbf{v}_k$ and cluster signal $\mathbf{c}_k$.

**Table 8: Single expert performs poorly (setting 1&2).** Test accuracy of single linear/nonlinear models with different activation functions. Data is generated according to Definition 3.1 with $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 1$ for setting 1. And we have $\alpha \in (0.5, 2)$, $\beta \in (1, 2)$, $\gamma \in (0.5, 3)$, $\sigma_p = 1$ for setting 2.

| Activation | Setting 1 | Setting 2 |
|---|---|---|
| Linear | 68.71% | 60.59% |
| Cubic | 79.48% | 72.29% |
| Relu | 72.28% | 80.12% |
| Celu | 81.75% | 78.99% |
| Gelu | 79.04% | 82.01% |
| Tanh | 81.72% | 81.03% |

**Table 9: Load balancing loss.** We report the results for linear MoE with load balancing loss and compare them with our previous results on nonlinear MoE without load balancing loss. Over ten random experiments, we report the average test accuracy (%) $\pm$ standard deviation. Setting 1-4 follows the data distribution introduced above.

| | Linear MoE with Load Balancing | Nonlinear MoE without Load Balancing |
|---|---|---|
| Setting 1 | $93.81 \pm 1.02$ | $\mathbf{99.46 \pm 0.55}$ |
| Setting 2 | $89.20 \pm 2.20$ | $\mathbf{98.09 \pm 1.27}$ |
| Setting 3 | $95.12 \pm 0.58$ | $\mathbf{99.99 \pm 0.02}$ |
| Setting 4 | $92.50 \pm 1.55$ | $\mathbf{98.92 \pm 1.18}$ |

**Table 10:** Nonlinear MoE with the **same initialization** and load balancing loss. The synthetic data is from setting 1. We report the average value $\pm$ standard deviation over 10 runs for both test accuracy and dispatch entropy.

| | Number of experts $M = 8$ | | Number of experts $M = 32$ | |
|---|---|---|---|---|
| Load Balancing Coeff | Accuracy (%) | Dispatch Entropy | Accuracy (%) | Dispatch Entropy |
| 0.1 | $72.18 \pm 1.16$ | $1.358 \pm 0.010$ | $70.88 \pm 0.60$ | $1.381 \pm 0.002$ |
| 0.03 | $79.97 \pm 1.61$ | $1.237 \pm 0.041$ | $77.02 \pm 0.51$ | $1.252 \pm 0.010$ |
| 0.01 | $78.59 \pm 2.19$ | $1.252 \pm 0.048$ | $79.15 \pm 0.87$ | $1.221 \pm 0.014$ |

that, with the same initialization, nonlinear MoEs exhibit performances that are very similar to a single nonlinear expert.

**Load Balancing Loss and Normalized Gradient Descent.** In Table 11, we report the average test accuracy for nonlinear MoE with regard to using load balancing loss and/or normalized gradient descent. The synthetic data is the same as in setting 1 and we choose number of experts $M = 32$. All the experts are randomly initialized. We observe that using normalized gradient descent or load balancing loss (or both) can lead to successful learning of the data distribution. However, without using normalized gradient or load balancing loss will result in failure of learning the data distribution.

### A.3 Experiments on Image Data

**Datasets.** We consider CIFAR-10 (Krizhevsky, 2009) with the 10-class classification task, which contains $50,000$ training examples and $10,000$ testing examples. For CIFAR-10-Rotate, we design a binary classification task by copying and rotating all images by 30 degree and let the model predict if an image is rotated. In Figure 7, we demonstrate the positive and negative examples of CIFAR-10-Rotate. Specifically, we crop the rotated images to $(24, 24)$, and resize to $(32, 32)$ for model architectures that are designed on image size $(32, 32)$. And we further apply random Gaussian noise to all images to avoid the models taking advantage of image resolutions.

**Models.** For the simple CNN model, we consider CNN with 2 convolutional layers, both with kernel size 3 and ReLU activation followed by max pooling with size 2 and a fully connected layer. The number of filters of each convolutional layer is respectively 64, 128.

**CIFAR-10 Setup.** For real-data experiments on CIFAR-10, we apply the commonly used transforms on CIFAR-10 before each forward pass: random horizontal flips and random crops (padding the

**Table 11: Ablation study of normalized gradient descent and load balancing loss.** We report the average test accuracy $\pm$ standard deviation over 10 runs for nonlinear MoE. We consider the following four configurations: 1. normalized GD with load balancing; 2. GD with load balancing; 3. normalized GD without load balancing; 4 GD without load balancing.

| | Number of experts $M = 32$ | |
|---|---|---|
| | Normalized GD | GD |
| With Load Balancing | $99.01 \pm 0.97$ | $98.64 \pm 0.34$ |
| Without Load Balancing | $99.47 \pm 0.48$ | $79.53 \pm 1.41$ |

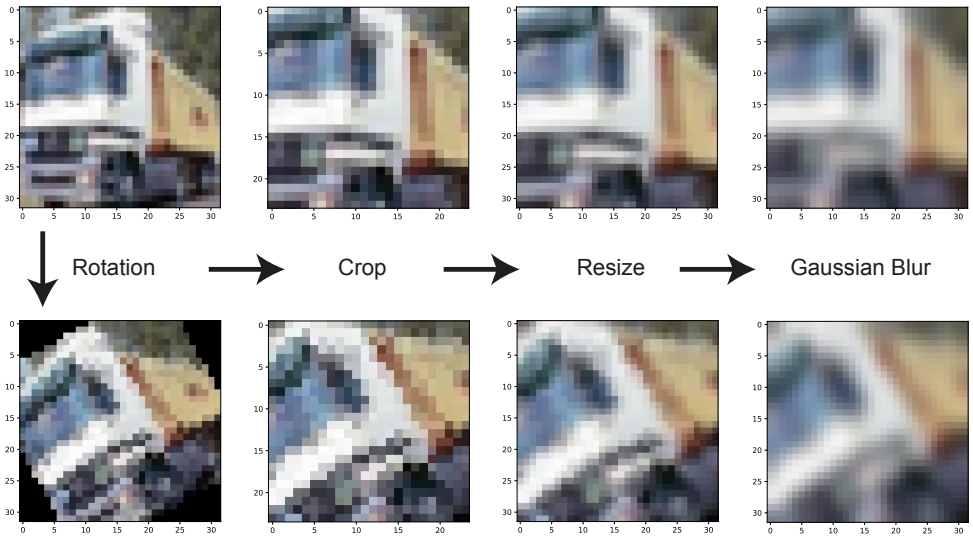


**Figure 7: Examples of the CIFAR-10-Rotate dataset.** Both the original image and the rotated image are processed in the same way, where we crop the image to $(24, 24)$, resize to $(32, 32)$ and apply random Gaussian blur.

**Table 12:** The test accuracy of the single classifier vs. MoE classifier.

| | Single | MoE |
|---|---|---|
| Accuracy | $74.13\%$ | $76.22\%$ |

images on all sides with $4$ pixels and randomly cropping to $(32, 32)$). And as conventionally, we normalize the data by channel. We train the single CNN model by SGD with learning rate $0.01$, momentum $0.9$ and weight decay 5e-4. And we train single MobileNetV2 and single ResNet18 by SGD with learning rate $0.1$, momentum $0.9$ and weight decay 5e-4 to achieve the best performances. We train MoEs according to Algorithm 1. Specifically, for MoE (ResNet18) and MoE (MobileNetV2), we use normalized gradient descent with learning rate $0.1$ and SGD with learning rate 1e-4, both with momentum $0.9$ and weight decay of 5e-4. For MoE (CNN), we use normalized gradient descent with learning rate $0.01$ and SGD with learning rate 1e-4, both with momentum $0.9$ and weight decay of 5e-4. We consider top-1 gating with noise and load balancing loss for MoE, where the multiplicative coefficient of load balancing loss is set at 1e-3. All models are trained for $200$ epochs to achieve convergence.

**CIFAR-10-Rotate Setup.** For experiments on CIFAR10-Rotate, the data is normalized by channel as the same as in CIFAR-10 before each forward pass. We train the single CNN, single MobileNetV2 and single ResNet18 by SGD with learning rate $0.01$, momentum $0.9$ and weight decay 5e-4 to achieve the best performances. And we train MoEs by Algorithm 1 with normalized gradient descent learning rate $0.01$ on the experts and with SGD of learning rate 1e-4 on the gating networks, both with momentum $0.9$ and weight decay of 5e-4. We consider top-1 gating with noise and load balancing loss for MoE, where the multiplicative coefficient for load balancing loss is set at 1e-3. All models are trained for $50$ epochs to achieve convergence.

### A.4 Experiments on Language Data

Here we provide a simple example of how MoE would work for multilingual tasks. We gather multilingual sentiment analysis data from the source of English (Sentiment140 (Go et al., 2009)) which is randomly sub-sampled to $200,000$ examples, Russian (RuReviews (Smetanin and Komarov, 2019)) which contains $90,000$ examples, and French (Blard, 2020) which contains $200,000$ examples. We randomly split the dataset into $80\%$ training data and $20\%$ test data. We use a pre-trained BERT multilingual base model (Devlin et al., 2018) to generate text embedding for each text. For the single model, we train an $1$-layer neural network with cubic activation. For MoE, we let $M = 4$ with each

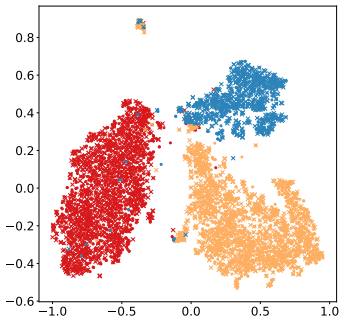


**Figure 8:** The distribution of text embedding of the multilingual sentiment analysis dataset. The embedding is generated by the pre-trained BERT multilingual base model and visualized on 2d space using t-SNE. Each color denotes a linguistic source, including English, French, and Russian.

**Table 13:** The final router dispatch details with regard to the linguistic source of the test data.

| | Expert 1 | Expert 2 | Expert 3 | Expert 4 |
|---|---|---|---|---|
| English | $1,374$ | $3,745$ | $2,999$ | $\mathbf{31,882}$ |
| French | $\mathbf{23,470}$ | $3,335$ | $\mathbf{13,182}$ | $13$ |
| Russian | $833$ | $\mathbf{9,405}$ | $7,723$ | $39$ |

expert sharing the same architecture as the single model. In Figure 8, we show the visualization of the text embeddings in the 2d space via t-SNE, where each color denotes a linguistic source. Here, $\cdot$ represents a positive example and $\times$ represents a negative example. We can observe that data from different linguistic sources naturally form different clusters.

In Table 12, we demonstrate the test accuracy of the single classifier and MoE on the multilingual sentiment analysis dataset, where we can observe an performance improvement of MoE over single model. And in Table 13, we show the final router dispatch details of MoE to each expert with regard to the language of the text. Notably, MoE learned to distribute examples largely according to the language.

## B Proof of Theorem 4.1

Because we are using CNNs as experts, different ordering of the patches won't affect the value of $F(\mathbf{x})$. So for $(\mathbf{x}, y)$ drawn from $\mathcal{D}$ in Definition 3.1, we can assume that the first patch $\mathbf{x}^{(1)}$ is feature signal, the second patch $\mathbf{x}^{(2)}$ is cluster-center signal, the third patch $\mathbf{x}^{(3)}$ is feature noise. The other patches $\mathbf{x}^{(p)}, p \geq 4$ are random noises. Therefore, we can rewrite $\mathbf{x} = [\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma \epsilon \mathbf{v}_{k'}, \boldsymbol{\xi}]$, where $\boldsymbol{\xi} = [\boldsymbol{\xi}_4, \ldots, \boldsymbol{\xi}_P]$ is a Gaussian matrix of size $\mathbb{R}^{d \times (P-3)}$.

*Proof of Theorem 4.1.* Conditioned on the event that $y = -\epsilon$, points $([\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, -\gamma y \mathbf{v}_{k'}, \boldsymbol{\xi}], y)$, $\big([-\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma y \mathbf{v}_{k'}, \boldsymbol{\xi}], -y\big)$, $\big([\gamma y \mathbf{v}_{k'}, \beta \mathbf{c}_{k'}, -\alpha y \mathbf{v}_k, \boldsymbol{\xi}], y\big)$, $\big([-\gamma y \mathbf{v}_{k'}, \beta \mathbf{c}_{k'}, \alpha y \mathbf{v}_k, \boldsymbol{\xi}], -y\big)$ follow the same distribution because $\gamma$ and $\alpha$ follow the same distribution, and $y$ and $-y$ follow the same distribution. Therefore, we have

$$
\begin{aligned}
&4\mathbb{P}\big(yF(\mathbf{x}) \leq 0 | \epsilon = -y\big) \\
&= \mathbb{E}\Bigg[\underbrace{\mathbb{1}(yF([\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, -\gamma y \mathbf{v}_{k'}, \boldsymbol{\xi}]) \leq 0)}_{I_1} + \underbrace{\mathbb{1}(-yF([-\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma y \mathbf{v}_{k'}, \boldsymbol{\xi}]) \leq 0)}_{I_2} \\
&\quad + \underbrace{\mathbb{1}(yF([\gamma y \mathbf{v}_{k'}, \beta \mathbf{c}_{k'}, -\alpha y \mathbf{v}_k, \boldsymbol{\xi}]) \leq 0)}_{I_3} + \underbrace{\mathbb{1}(-yF([-\gamma y \mathbf{v}_{k'}, \beta \mathbf{c}_{k'}, \alpha y \mathbf{v}_k, \boldsymbol{\xi}]) \leq 0)}_{I_4}\Bigg].
\end{aligned}
$$

It is easy to verify the following fact

$$
\begin{aligned}
&\Big(yF([\alpha y\mathbf{v}_k, \beta\mathbf{c}_k, -\gamma y\mathbf{v}_{k'}, \boldsymbol{\xi}])\Big) + \Big(-yF([-\alpha y\mathbf{v}_k, \beta\mathbf{c}_k, \gamma y\mathbf{v}_{k'}, \boldsymbol{\xi}])\Big) \\
&\quad + \Big(yF([\gamma y\mathbf{v}_{k'}, \beta\mathbf{c}_{k'}, -\alpha y\mathbf{v}_k, \boldsymbol{\xi}])\Big) + \Big(-yF([-\gamma y\mathbf{v}_{k'}, \beta\mathbf{c}_{k'}, \alpha y\mathbf{v}_k, \boldsymbol{\xi}])\Big) \\
&= \Big(yf(\alpha y\mathbf{v}_k) + yf(\beta\mathbf{c}_k) + yf(-\gamma y\mathbf{v}_{k'}) + \sum_{p=4}^{P} yf(\boldsymbol{\xi}_p)\Big) \\
&\quad + \Big(-yf(-\alpha y\mathbf{v}_k) - yf(\beta\mathbf{c}_k) - yf(\gamma y\mathbf{v}_{k'}) - \sum_{p=4}^{P} yf(\boldsymbol{\xi}_p)\Big) \\
&\quad + \Big(yf(\gamma y\mathbf{v}_{k'}) + yf(\beta\mathbf{c}_{k'}) + yf(-\alpha y\mathbf{v}_k) + \sum_{p=4}^{P} yf(\boldsymbol{\xi}_p)\Big) \\
&\quad + \Big(-yf(-\gamma y\mathbf{v}_{k'}) - yf(\beta\mathbf{c}_{k'}) - yf(\alpha y\mathbf{v}_k) - \sum_{p=4}^{P} yf(\boldsymbol{\xi}_p)\Big) \\
&= 0.
\end{aligned}
$$

By pigeonhole principle, at least one of $I_1, I_2, I_3, I_4$ is non-zero. This further implies that $4\mathbb{P}\big(yF(\mathbf{x}) \le 0 | \epsilon = -y\big) \ge 1$. Applying $\mathbb{P}(\epsilon = -y) = 1/2$, we have that

$$\mathbb{P}\big(yF(\mathbf{x}) \le 0\big) \ge \mathbb{P}\big(yF(\mathbf{x}) \le 0) | \epsilon = -y\big)\mathbb{P}(\epsilon = -y) \ge 1/8,$$

which completes the proof. □

# C Smoothed Router

In this section, we will show that the noise term provides a smooth transition between different routing behavior. All the results in this section is independent from our NN structure and its initialization. We first present a general version of Lemma 5.1 with its proof.

**Lemma C.1** (Extension of Lemma 5.1)**.** Let $\mathbf{h}, \widehat{\mathbf{h}} \in \mathbb{R}^M$ to be the output of the gating network and $\{r_m\}_{m=1}^M$ to be the noise independently drawn from $\mathcal{D}_r$. Denote $\mathbf{p}, \widehat{\mathbf{p}} \in \mathbb{R}^M$ to be the probability that experts get routed, i.e., $p_m = \mathbb{P}(\mathrm{argmax}_{m' \in [M]}\{h_{m'} + r_{m'}\} = m)$, $\widehat{p}_m = \mathbb{P}(\mathrm{argmax}_{m' \in [M]}\{\widehat{h}_{m'} + r_{m'}\} = m)$. Suppose the probability density function of $\mathcal{D}_r$ is bounded by $\kappa$, Then we have that $\|\mathbf{p} - \widehat{\mathbf{p}}\|_\infty \le (\kappa M^2) \cdot \|\mathbf{h} - \widehat{\mathbf{h}}\|_\infty$.

*Proof.* Given random variable $\{r_m\}_{m=1}^M$, let us first consider the event that $\mathrm{argmax}_m\{h_m + r_m\} \ne \mathrm{argmax}_m\{\widehat{h}_m + r_m\}$. Let $m_1 = \mathrm{argmax}_m\{h_m + r_m\}$ and $m_2 = \mathrm{argmax}_m\{\widehat{h}_m + r_m\}$, then we have that

$$h_{m_1} + r_{m_1} \ge h_{m_2} + r_{m_2}, \widehat{h}_{m_2} + r_{m_2} \ge \widehat{h}_{m_1} + r_{m_1},$$

which implies that

$$\widehat{h}_{m_2} - \widehat{h}_{m_1} \ge r_{m_1} - r_{m_2} \ge h_{m_2} - h_{m_1}. \tag{C.1}$$

Define $C(m_1, m_2) = (\widehat{h}_{m_2} - \widehat{h}_{m_1} + h_{m_2} - h_{m_1})/2$, then (C.1) implies that

$$|r_{m_1} - r_{m_2} - C(m_1, m_2)| \le |\widehat{h}_{m_2} - \widehat{h}_{m_1} - h_{m_2} + h_{m_1}|/2 \le \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty. \tag{C.2}$$

Therefore, we have that,

$$
\begin{aligned}
&\mathbb{P}(\operatorname*{argmax}_m\{h_m + r_m\} \ne \operatorname*{argmax}_m\{\widehat{h}_m + r_m\}) \\
&\le \mathbb{P}(\exists m_1 \ne m_2 \in [M], \text{ s.t. } |r_{m_1} - r_{m_2} - C(m_1, m_2)| \le \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty) \\
&\le \sum_{m_1 < m_2} \mathbb{P}\big(|r_{m_1} - r_{m_2} - C(m_1, m_2)| \le \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty\big) \\
&= \sum_{m_1 < m_2} \mathbb{E}\Big[\mathbb{P}\big(r_{m_2} + C(m_1, m_2) - \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty \le r_{m_1} \le r_{m_2} + C(m_1, m_2) + \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty\big)\Big| r_{m_2}\Big] \\
&\le (\kappa M^2) \cdot \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty,
\end{aligned}
$$

where the first inequality is by (C.2), the second inequality is by union bound and the last inequality is due to the fact that the probability density function of $r_{m_1}$ is bounded by $\kappa$. Then we have that for $i \in [M]$,

$$\begin{aligned}
|p_i - \widehat{p}_i| &\le \left| \mathbb{E}\Big[ \mathbb{1}\big( \operatorname*{argmax}_m\{\widehat{h}_m + r_m\} = i\big) - \mathbb{1}\big( \operatorname*{argmax}_m\{h_m + r_m\} = i\big) \Big] \right| \\
&\le \mathbb{E}\left| \mathbb{1}\big( \operatorname*{argmax}_m\{\widehat{h}_m + r_m\} = i\big) - \mathbb{1}\big( \operatorname*{argmax}_m\{h_m + r_m\} = i\big) \right| \\
&\le \mathbb{P}\big( \operatorname*{argmax}_m\{\widehat{h}_m + r_m\} \neq \operatorname*{argmax}_m\{h_m + r_m\}\big) \\
&\le (\kappa M^2) \cdot \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty,
\end{aligned}$$

which completes the proof. □

**Remark C.2.** A widely used choice of $\mathcal{D}_r$ in Lemma C.1 is uniform noise Unif[a, b], in which case the density function can be upper bounded by $1/(b-a)$. Another widely used choice of $\mathcal{D}_r$ is Gaussian noise $\mathcal{N}(0, \sigma_r^2)$, in which case the density function can be upper bounded by $1/(\sigma_r\sqrt{2\pi})$. Increase the range of uniform noise or increase the variance of the Gaussian noise will result in a smaller density function upper bound and a smoother behavior of routing. In our paper, we consider unif[0,1] for simplicity, in which case the the density function can be upper bounded by 1 ($\kappa = 1$).

The following Lemma shows that when two gate network outputs are close, the router will distribute the examples to those corresponding experts with nearly the same probability.

**Lemma C.3.** Let $\mathbf{h} \in \mathbb{R}^M$ be the output of the gating network and $\{r_m\}_{m=1}^M$ be the noise independently drawn from Unif[0,1]. Denote the probability that experts get routed by $\mathbf{p}$, i.e., $p_m = \mathbb{P}(\operatorname{argmax}_{m'}\{h_{m'} + r_{m'}\} = m)$. Then we have that

$$|p_m - p_{m'}| \le M^2 |h_m - h_{m'}|.$$

*Proof.* Construct $\widehat{\mathbf{h}}$ as copy of $\mathbf{h}$ and permute its $m, m'$-th element. Denote the corresponding probability vector as $\widehat{\mathbf{p}}$. Then it is obviously that $|p_m - p_{m'}| = \|\mathbf{p} - \widehat{\mathbf{p}}\|_\infty$ and $|h_m - h_{m'}| = \|\widehat{\mathbf{h}} - \mathbf{h}\|_\infty$. Applying Lemma 5.1 completes the proof. □

The following lemma shows that the router won't route examples to the experts with small gating network outputs, which saves computation and improves the performance.

**Lemma C.4.** Suppose the noise $\{r_m\}_{m=1}^M$ are independently drawn from Unif[0,1] and $h_m(\mathbf{x}; \mathbf{\Theta}) \le \max_{m'} h_{m'}(\mathbf{x}; \mathbf{\Theta}) - 1$, example $\mathbf{x}$ will not get routed to expert $m$.

*Proof.* Because $h_m(\mathbf{x}; \mathbf{\Theta}) \le \max_{m'} h_{m'}(\mathbf{x}; \mathbf{\Theta}) - 1$ implies that for any Uniform noise $\{r_{m'}\}_{m' \in [M]}$ we have that

$$h_m(\mathbf{x}; \mathbf{\Theta}) + r_m \le \max_{m'} h_{m'}(\mathbf{x}; \mathbf{\Theta}) \le \max_{m'}\{h_{m'}(\mathbf{x}; \mathbf{\Theta}) + r_{m'}\},$$

where the first inequality is by $r_m \le 1$, the second inequality is by $r_{m'} \ge 0, \forall m' \in [M]$. □

# D Initialization of the Model

Before we look into the detailed proof of Theorem 4.2, let us first discuss some basic properties of the data distribution and our MoE model. For simplicity of notation, we simplify $(\mathbf{x}_i, y_i) \in \Omega_k$ as $i \in \Omega_k$.

**Training Data Set Property.** Because we are using CNNs as experts, different ordering of the patches won't affect the value of $F(\mathbf{x})$. So for $(\mathbf{x}, y)$ drawn from $\mathcal{D}$ in Definition 3.1, we can assume that the first patch $\mathbf{x}^{(1)}$ is feature signal, the second patch $\mathbf{x}^{(2)}$ is cluster-center signal, the third patch $\mathbf{x}^{(3)}$ is feature noise. The other patches $\mathbf{x}^{(p)}, p \ge 4$ are random noises. Therefore, we can rewrite $\mathbf{x} = [\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma \epsilon \mathbf{v}_{k'}, \boldsymbol{\xi}]$, where $\boldsymbol{\xi} = [\boldsymbol{\xi}_4, \ldots, \boldsymbol{\xi}_P]$ is a Gaussian matrix of size $\mathbb{R}^{d \times (P-3)}$. According to the type of the feature noise, we further divide $\Omega_k$ into $\Omega_k = \cup \Omega_{k,k'}$ based on the feature noise, i.e. $\mathbf{x} \in \Omega_{k,k'}$ if $\mathbf{x} = [\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma \epsilon \mathbf{v}_{k'}, \boldsymbol{\xi}]$. To better characterize the router training,

we need to break down $\Omega_{k,k'}$ into $\Omega^+_{k,k'}$ and $\Omega^-_{k,k'}$. Denote by $\Omega^+_{k,k'}$ the set that $\{y_i = \epsilon_i | i \in \Omega_{k,k'}\}$, by $\Omega^-_{k,k'}$ the set that $\{y_i = -\epsilon_i | i \in \Omega_{k,k'}\}$.

**Lemma D.1.** With probability at least $1 - \delta$, the following properties hold for all $k \in [K]$,

$$\sum_{i\in\Omega_k} y_i\beta_i^3 = \widetilde{O}(\sqrt{n}), \sum_{i\in\Omega_k} \alpha_i^3 = \mathbb{E}[\alpha^3]\cdot n/K + \widetilde{O}(\sqrt{n}), \sum_{i\in\Omega_k} y_i\epsilon_i\gamma_i^3 = \widetilde{O}(\sqrt{n}), \tag{D.1}$$

$$\sum_{i\in\Omega^+_{k,k'}} y_i\alpha_i = \widetilde{O}(\sqrt{n}), \sum_{i\in\Omega^-_{k,k'}} y_i\alpha_i = \widetilde{O}(\sqrt{n}), \sum_{i\in\Omega^+_{k,k'}} \epsilon_i\gamma_i = \widetilde{O}(\sqrt{n}), \tag{D.2}$$

$$\sum_{i\in\Omega^-_{k,k'}} \epsilon_i\gamma_i = \widetilde{O}(\sqrt{n}), \sum_{i\in\Omega_k} \beta_i = \mathbb{E}[\beta]\cdot n/K + \widetilde{O}(\sqrt{n}). \tag{D.3}$$

*Proof.* Fix $k \in [K]$, by Hoeffding's inequality we have that with probability at least $1 - \delta/8K$,

$$\sum_{i\in\Omega_k} y_i\beta_i^3 = \sum_{i=1}^n y_i\beta_i^3 \mathbb{1}\left((\mathbf{x}_i, y_i) \in \Omega_k\right) = \widetilde{O}(\sqrt{n}),$$

where the last equality is by the fact that the expectation of $y\beta^3 \mathbb{1}\left((\mathbf{x}, y) \in \Omega_k\right)$ is zero. Fix $k \in [K]$, by Hoeffding's inequality we have that with probability at least $1 - \delta/8K$,

$$\sum_{i\in\Omega_k} \alpha_i^3 = \sum_{i=1}^n \alpha_i^3 \mathbb{1}\left((\mathbf{x}_i, y_i) \in \Omega_k\right) = \frac{n\mathbb{E}[\alpha^3]}{K} + \widetilde{O}(\sqrt{n}),$$

where the last equality is by the fact that the expectation of $\alpha^3 \mathbb{1}\left((\mathbf{x}, y) \in \Omega_k\right)$ is $\mathbb{E}[\alpha^3]/K$. Fix $k \in [K]$, by Hoeffding's inequality we have that with probability at least $1 - \delta/8K$,

$$\sum_{i\in\Omega_k} y_i\epsilon_i\gamma_i^3 = \sum_{i=1}^n y_i\epsilon_i\gamma_i^3 \mathbb{1}\left((\mathbf{x}_i, y_i) \in \Omega_k\right) = \widetilde{O}(\sqrt{n}),$$

where the last equality is by the fact that the expectation of $y\epsilon\gamma^3 \mathbb{1}\left((\mathbf{x}, y) \in \Omega_k\right)$ is zero. Now we have proved the bounds in (D.1). We can get other bounds in (D.2) and (D.3) similarly. Applying union bound over $[K]$ completes the proof. □

**Lemma D.2.** Suppose that $d = \Omega(\log(4nP/\delta))$, with probability at least $1 - \delta$, the following inequalities hold for all $i \in [n], k \in [K], p \geq 4$,

- $\|\boldsymbol{\xi}_{i,p}\|_2 = O(1)$,
- $\langle \mathbf{v}_k, \boldsymbol{\xi}_{i,p}\rangle \leq \widetilde{O}(d^{-1/2})$, $\langle \mathbf{c}_k, \boldsymbol{\xi}_{i,p}\rangle \leq \widetilde{O}(d^{-1/2})$, $\langle \boldsymbol{\xi}_{i,p}, \boldsymbol{\xi}_{i',p'}\rangle \leq \widetilde{O}(d^{-1/2})$, $\forall (i', p') \neq (i, p)$.

*Proof of Lemma D.2.* By Bernstein's inequality, with probability at least $1 - \delta/(2nP)$ we have

$$\left|\|\boldsymbol{\xi}_{i,p}\|_2^2 - \sigma_p^2\right| \leq O(\sigma_p^2\sqrt{d^{-1}\log(4nP/\delta)}).$$

Therefore, as long as $d = \Omega(\log(4nP/\delta))$, we have $\|\boldsymbol{\xi}_{i,p}\|_2^2 \leq 2$. Moreover, clearly $\langle \boldsymbol{\xi}_{i,p}, \boldsymbol{\xi}_{i',p'}\rangle$ has mean zero, $\forall (i, p) \neq (i', p')$. Then by Bernstein's inequality, with probability at least $1 - \delta/(6n^2P^2)$ we have

$$|\langle \boldsymbol{\xi}_{i,p}, \boldsymbol{\xi}_{i',p'}\rangle| \leq 2\sigma_p^2\sqrt{d^{-1}\log(12n^2P^2/\delta)}.$$

Similarly, $\langle \mathbf{v}_k, \boldsymbol{\xi}_{i,p}\rangle$ and $\langle \mathbf{c}_k, \boldsymbol{\xi}_{i,p}\rangle$ have mean zero. Then by Bernstein's inequality, with probability at least $1 - \delta/(3nPK)$ we have

$$|\langle \boldsymbol{\xi}_{i,p}, \mathbf{v}_k\rangle| \leq 2\sigma_p\sqrt{d^{-1}\log(6nPK/\delta)}, |\langle \boldsymbol{\xi}_{i,p}, \mathbf{c}_k\rangle| \leq 2\sigma_p\sqrt{d^{-1}\log(6nPK/\delta)}.$$

Applying a union bound completes the proof. □

**MoE Initialization Property.**

We divide the experts into $K$ sets based on the initialization.

**Definition D.3.** Fix expert $m \in [M]$, denote $(k_m^*, j_m^*) = \operatorname{argmax}_{j,k}\langle \mathbf{v}_k, \mathbf{w}_{m,j}^{(0)}\rangle$. Fix cluster $k \in [K]$, denote the profession experts set as $\mathcal{M}_k = \{m | k_m^* = k\}$.

**Lemma D.4.** For $M \geq \Theta(K\log(K/\delta))$, $J \geq \Theta(\log(M/\delta))$, the following inequalities hold with probability at least $1-\delta$.

- $\max_{(j,k)\neq(j_m^*,k_m^*)}\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle \leq \big(1-\delta/\big(3MJ^2K^2\big)\big)\langle \mathbf{w}_{m,j_m^*}^{(0)}, \mathbf{v}_{k_m^*}\rangle$ for all $m \in [M]$
- $\langle \mathbf{w}_{m,j_m^*}^{(0)}, \mathbf{v}_{k_m^*}\rangle \geq 0.01\sigma_0$ for all $m \in [M]$.
- $|\mathcal{M}_k| \geq 1$ for all $k \in [K]$.

*Proof.* Recall that $\mathbf{w}_{m,j} \sim \mathcal{N}(0, \sigma_0^2 I_d)$. Notice that signals $\mathbf{v}_1, \ldots, \mathbf{v}_K$ are orthogonal. Given fixed $m \in [M]$, we have that $\{\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle | j \in [J], k \in [K]\}$ are independent and individually draw from $\mathcal{N}(0, \sigma_0^2)$ we have that

$$\mathbb{P}(\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle < 0.01\sigma_0) < 0.9.$$

Therefore, we have that

$$\mathbb{P}(\max_{j,k}\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle < 0.01\sigma_0) < 0.9^{KJ}.$$

Therefore, as long as $J \geq \Theta(K^{-1}\log(M/\delta))$, fix $m \in [M]$ we can guarantee that with probability at least $1-\delta/(3M)$,

$$\max_{j,k}\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle > 0.01\sigma_0.$$

Take $G = \delta/(3MJ^2K^2)$, by Lemma F.1 we have that with probability at least $1-\delta/(3M)$,

$$\max_{(j,k)\neq(j_m^*,k_m^*)}\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle \leq (1-G)\langle \mathbf{w}_{m,j_m^*}^{(0)}, \mathbf{v}_{k_m^*}\rangle.$$

By the symmetric property, we have that for all $k \in [K], m \in [M]$,

$$\mathbb{P}(k = k_m^*) = K^{-1}.$$

Therefore, the probability that $|\mathcal{M}_k|$ at least include one element is as follows,

$$\mathbb{P}(|\mathcal{M}_k| \geq 1) \geq 1-(1-K^{-1})^M.$$

By union bound we get that

$$\mathbb{P}(|\mathcal{M}_k| \geq 1, \forall k) \geq 1-K(1-K^{-1})^M \geq 1-K\exp(-M/K) \geq 1-\delta/3,$$

where the last inequality is by condition $M \geq K\log(3K/\delta)$. Therefore, with probability at least $1-\delta/3$, $|\mathcal{M}_k| \geq 1, \forall k$.

Applying Union bound, we have that with probability at least $1-\delta$,

$$\max_{(j,k)\neq(j_m^*,k_m^*)}\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}_k\rangle \leq \big(1-\delta/\big(3MJ^2K^2\big)\big)\langle \mathbf{w}_{m,j_m^*}^{(0)}, \mathbf{v}_{k_m^*}\rangle,$$
$$\langle \mathbf{w}_{m,j_m^*}^{(0)}, \mathbf{v}_{k_m^*}\rangle \geq 0.01\sigma_0, \forall m \in [M],$$
$$|\mathcal{M}_k| \geq 1, \forall k \in [K].$$

□

**Lemma D.5.** Suppose the conclusions in Lemma D.2 hold, then with probability at least $1-\delta$ we have that $|\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}\rangle| \leq \widetilde{O}(\sigma_0)$ for all $\mathbf{v} \in \{\mathbf{v}_k\}_{k\in[K]} \cup \{\mathbf{c}_k\}_{k\in[K]} \cup \{\boldsymbol{\xi}_{i,p}\}_{i\in[n],p\in[P-3]}, m \in [M], j \in [J]$.

*Proof.* Fix $\mathbf{v} \in \{\mathbf{v}_k\}_{k\in[K]} \cup \{\mathbf{c}_k\}_{k\in[K]} \cup \{\boldsymbol{\xi}_{i,p}\}_{i\in[n],p\in[P-3]}, m \in [M], j \in [J]$, we have that $\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}\rangle \sim \mathcal{N}(0, \sigma_0^2\|\mathbf{v}\|_2^2)$ and $\|\mathbf{v}\|_2 = O(1)$. Therefore, with probability at least $1-\delta/(nPMJ)$ we have that $|\langle \mathbf{w}_{m,j}^{(0)}, \mathbf{v}\rangle| \leq \widetilde{O}(\sigma_0)$. Applying union bound completes the proof. □

## E Proof of Theorem 4.2

In this section we always assume that the conditions in Theorem 4.2 holds. It is easy to show that all the conclusions in this section D hold with probability at least $1 - O(1/\log d)$. The results in this section hold when all the conclusions in Section D hold. For simplicity of notation, we simplify $(\mathbf{x}_i, y_i) \in \Omega_{k,k'}$ as $i \in \Omega_{k,k'}$, and $\ell'(y_i \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}))$ as $\ell'_{i,t}$.

Recall that at iteration $t$, data $\mathbf{x}_i$ is routed to the expert $m_{i,t}$. Here $m_{i,t}$ should be interpreted as a random variable. The gradient of MoE model at iteration $t$ can thus be computed as follows

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)} &= \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)})(1 - \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)})) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&\quad - \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} \neq m) \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&= \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&\quad - \frac{1}{n} \sum_{i,p} \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)}, \qquad \text{(E.1)} \\
\nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)} &= \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \mathbf{x}_i^{(p)}. \qquad \text{(E.2)}
\end{aligned}$$

Following lemma shows implicit regularity in the gating network training.

**Lemma E.1.** For all $t \geq 0$, we have that $\sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)} = \mathbf{0}$ and thus $\sum_m \boldsymbol{\theta}_m^{(t)} = \sum_m \boldsymbol{\theta}_m^{(0)}$. In particular, when $\mathbf{\Theta}$ is zero initialized, then $\sum_m \boldsymbol{\theta}_m^{(t)} = 0$

*Proof.* We first write out the gradient of $\boldsymbol{\theta}_m$ for all $m \in [M]$,

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)} &= \frac{1}{n} \sum_{i \in [n], p \in [P]} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&\quad - \frac{1}{n} \sum_{i \in [n], p \in [P]} \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)}.
\end{aligned}$$

Take summation from $m = 1$ to $m = M$, then we have

$$\begin{aligned}
\sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)} &= \frac{1}{n} \sum_{i \in [n], p \in [P]} \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i; \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&\quad - \frac{1}{n} \sum_{i \in [n], p \in [P]} \ell'_{i,t} \pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i f_{m_{i,t}}(\mathbf{x}_i, \mathbf{W}^{(t)}) \mathbf{x}_i^{(p)} \\
&= 0.
\end{aligned}$$

□

Notice that the gradient at iteration $t$ in (E.1) and (E.2) is depend on the random variable $m_{i,t}$, the following lemma shows that it can be approximated by its expectation.

**Lemma E.2.** With probability at least $1 - 1/d$, for all the vector $\mathbf{v} \in \{\mathbf{v}_k\}_{k \in [K]} \cup \{\mathbf{c}_k\}_{k \in [K]}$, $m \in [M]$, $j \in [J]$, we have the following equations hold $|\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle - \mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle]| = \widetilde{O}(n^{-1/2}(\sigma_0 + \eta t)^3)$, $|\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle - \mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle]| = \widetilde{O}(n^{-1/2}(\sigma_0 + \eta t)^2)$, for all $t \leq d^{100}$. Here $\mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle]$ and $\mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle]$ can be computed as follows,

$$\mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle] = \frac{1}{n} \sum_{i,p} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{v} \rangle$$
$$- \frac{1}{n} \sum_{i,p,m'} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_{m'}(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{v} \rangle$$
$$\mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle] = \frac{1}{n} \sum_{i,p} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \langle \mathbf{x}_i^{(p)}, \mathbf{v} \rangle.$$

*Proof.* Because we are using normalized gradient descent, $\|\mathbf{w}_{m,j}^{(t)} - \mathbf{w}_{m,j}^{(0)}\|_2 \leq O(\eta t)$ and thus by Lemma D.5 we have $|\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle| \leq \widetilde{O}(\sigma_0 + \eta t)$. Therefore,

$$\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle = \frac{1}{n} \sum_i \underbrace{\sum_p \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \langle \mathbf{x}_i^{(p)}, \mathbf{v} \rangle}_{A_i},$$

where $A_i$ are independent random variables with $|A_i| \leq \widetilde{O}\big((\sigma_0 + \eta t)^2\big)$. Applying Hoeffding's inequality gives that with probability at least $1 - 1/(4d^{101}MJK)$ we have that $|\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle - \mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle]| = \widetilde{O}(n^{-1/2}(\sigma_0 + \eta t)^2)$. Applying union bound gives that with probability at least $1 - 1/(2d)$, $|\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle - \mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v} \rangle]| = \widetilde{O}(n^{-1/2}(\sigma_0 + \eta t)^2), \forall m \in [M], j \in [J], t \leq d^{100}$. Similarly, we can prove $|\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle - \mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v} \rangle]| = \widetilde{O}(n^{-1/2}(\sigma_0 + \eta t)^3)$.

□

### E.1 Exploration Stage

Denote $T_1 = \lfloor \eta^{-1} \sigma_0^{0.5} \rfloor$. The first stage ends when $t = T_1$. During the first stage training, we can prove that the neural network parameter maintains the following property.

**Lemma E.3.** For all $t \leq T_1$, we have the following properties hold,

- $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle = O(\sigma_0^{0.5}), \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle = O(\sigma_0^{0.5}), \langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle = \widetilde{O}(\sigma_0^{0.5})$,
- $f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) = \widetilde{O}(\sigma_0^{1.5})$,
- $|\ell'_{i,t} - 1/2| \leq \widetilde{O}(\sigma_0^{1.5})$,
- $\|\boldsymbol{\theta}_m^{(t)}\|_2 \leq \widetilde{O}(\sigma_0^{1.5})$,
- $\|\mathbf{h}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})\|_\infty = \widetilde{O}(\sigma_0^{1.5})$, $\pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) = M^{-1} + \widetilde{O}(\sigma_0^{1.5})$,

for all $m \in [M], k \in [k], i \in [n], p \geq 4$.

*Proof.* The first property is obvious since $\|\mathbf{w}_{m,j}^{(t)} - \mathbf{w}_{m,j}^{(0)}\|_2 \leq O(\eta T_1) = O(\sigma_0^{0.5})$ and thus

$$|f_m(\mathbf{x}_i; \mathbf{W}^{(t)})| \leq \sum_{p \in [P]} \sum_{j \in [J]} |\sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle)| = \widetilde{O}(\sigma_0^{1.5}).$$

Then we show that the loss derivative is close to $1/2$ during this stage.

Let $s = y_i \pi_{m_{i,t}}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_{m_{i,t}}(\mathbf{x}_i, \mathbf{W}^{(t)})$, then we have that $|s| = \widetilde{O}(\sigma_0^{1.5})$ and

$$\left| \ell'_{i,t} - \frac{1}{2} \right| = \left| \frac{1}{e^s + 1} - 1/2 \right| \overset{(i)}{\leq} |s| = \widetilde{O}(\sigma_0^{1.5}),$$

where $(i)$ can be proved by considering $|s| \leq 1$ and $|s| > 1$.

Now we prove the fourth bullet in Lemma E.3. Because $|f_m| = \widetilde{O}(\sigma_0^{1.5})$, we can upper bound the gradient of the gating network by

$$\begin{aligned}\|\nabla_{\boldsymbol{\theta}_m}\mathcal{L}^{(t)}\|_2 = \bigg\| &\frac{1}{n}\sum_{i,p}\mathbb{1}(m_{i,t}=m)\ell'_{i,t}\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})y_i f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\mathbf{x}_i^{(p)} \\ &- \frac{1}{n}\sum_{i,p}\ell'_{i,t}\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})y_i f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\mathbf{x}_i^{(p)}\bigg\|_2. \\ = \widetilde{O}(\sigma_0^{1.5}),\end{aligned}$$

where the last inequality is due to $|\ell'_{i,t}| \leq 1$, $\pi_m, \pi_{m_{i,t}} \in [0,1]$ and $\|\mathbf{x}_i^{(p)}\|_2 = O(1)$. This further implies that

$$\|\boldsymbol{\theta}_m^{(t)}\|_2 = \|\boldsymbol{\theta}_m^{(t)} - \boldsymbol{\theta}_m^{(0)}\|_2 \leq \widetilde{O}(\sigma_0^{1.5} t\eta_r) = \widetilde{O}(\sigma_0^{1.5}),$$

where the last inequality is by $\eta_r = \Theta(M^2)\eta$. The proof of $\|\mathbf{h}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\|_\infty \leq O(\sigma_0^{1.5})$ and $\pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) = M^{-1} + O(\sigma_0^{1.5})$ are straight forward given $\|\boldsymbol{\theta}_m^{(t)}\|_2 = \widetilde{O}(\sigma_0^{1.5})$. □

We will first investigate the property of the router.

**Lemma E.4.** $\max_{m\in[M]} |\mathbb{P}(m_{i,t}=m) - 1/M| = \widetilde{O}(\sigma_0^{1.5})$ for all $t \leq T_1$, $i \in [n]$ and $m \in [M]$.

*Proof.* By Lemma E.3 we have that $\|\mathbf{h}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\|_\infty \leq \widetilde{O}(\sigma_0^{1.5})$. Lemma 5.1 further implies that

$$\max_{m\in[M]} |\mathbb{P}(m_{i,t}=m) - 1/M| = \widetilde{O}(\sigma_0^{1.5}).$$

□

**Lemma E.5.** We have following gradient update rules hold for the experts,

$$\begin{aligned}\langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)}, \mathbf{v}_k\rangle &= -\frac{\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2}\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k\rangle) + \widetilde{O}(\sigma_0^{2.5}), \\ \langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)}, \mathbf{c}_k\rangle &= \widetilde{O}(d^{-0.005})\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k\rangle) + \widetilde{O}(\sigma_0^{2.5}), \\ \langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)}, \boldsymbol{\xi}_{i,p}\rangle &= \widetilde{O}(d^{-0.005})\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p}\rangle) + \widetilde{O}(\sigma_0^{2.5})\end{aligned}$$

for all $t \leq T_1, j \in [J], k \in [K], m \in [M], p \geq 4$. Besides, we have the following gradient norm upper bound holds

$$\begin{aligned}\|\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)}\|_2 \leq &\sum_{k\in[K]}\frac{\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2}\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k\rangle) + \sum_{k\in[K]}\widetilde{O}(d^{-0.005})\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k\rangle) \\ &+ \sum_{i\in[n],p\geq 4}\widetilde{O}(d^{-0.005})\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p}\rangle) + \widetilde{O}(\sigma_0^{2.5})\end{aligned}$$

for all $t \leq T_1, j \in [J], m \in [M]$.

*Proof.* The experts gradient can be computed as follows,

$$\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)} = \frac{1}{n}\sum_{i\in[n],p\in[P]}\mathbb{1}(m_{i,t}=m)\ell'_{i,t}f_m(\mathbf{x}_i;\mathbf{W}^{(t)})\pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})y_i\sigma'(\langle\mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)}\rangle)\mathbf{x}_i^{(p)}.$$

We first compute the inner product $\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle$. By Lemma E.2, we have that $|\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle - \mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle]| = \widetilde{O}(n^{-1/2}\sigma_0) \leq \widetilde{O}(\sigma_0^{2.5})$.

$$
\begin{aligned}
\mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle] &= -\frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) y_i \beta_i^3 \|\mathbf{c}_k\|_2^2 \\
&\quad - \frac{1}{n} \sum_{i \in [n], p \geq 4} \mathbb{P}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) y_i \langle \mathbf{c}_k, \boldsymbol{\xi}_{i,p} \rangle \\
&= \bigg[ -\frac{1}{2nM} \sum_{i \in \Omega_k} y_i \beta_i^3 \mathbb{P}(m_{i,t} = m) + \widetilde{O}(\sigma_0^{1.5}) \bigg] \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) + \widetilde{O}(\sigma_0^{2.5}) \\
&= \widetilde{O}(n^{-1/2} + \sigma_0^{1.5}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) + \widetilde{O}(\sigma_0^{2.5}) \\
&= \widetilde{O}(d^{-0.005}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) + \widetilde{O}(\sigma_0^{2.5})
\end{aligned}
$$

where the second equality is due to Lemma E.3 and D.2, the third equality is due to Lemma E.4, the last equality is by the choice of $n$ and $\sigma_0$. Next we compute the inner product $\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}, \mathbf{v}_k \rangle$. By Lemma E.2, we have that $|\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v}_k \rangle - \mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v}_k \rangle]| = \widetilde{O}(n^{-1/2}\sigma_0) \leq \widetilde{O}(\sigma_0^{2.5})$.

$$
\begin{aligned}
\mathbb{E}[\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v}_k \rangle] &= -\frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) \alpha_i^3 \|\mathbf{v}_k\|_2^2 \\
&\quad - \frac{1}{n} \sum_{k' \neq k} \sum_{i \in \Omega_{k',k}} \mathbb{P}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \boldsymbol{\theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) \gamma_i^3 y_i \epsilon_i \|\mathbf{v}_k\|_2^2 \\
&\quad - \frac{1}{n} \sum_{i \in [n], p \geq 4} \mathbb{P}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) y_i \langle \mathbf{v}_k, \boldsymbol{\xi}_{i,p} \rangle \\
&= \bigg[ -\frac{1}{2nM} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \alpha_i^3 - \frac{1}{2nM} \sum_{i \in \Omega_{k',k}} \mathbb{P}(m_{i,t} = m) \gamma_i^3 y_i \epsilon_i + O(\sigma_0^{1.5}) \bigg] \cdot \\
&\quad \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) + \widetilde{O}(\sigma_0^{2.5}) \\
&= \big( \mathbb{E}[\alpha^3] + \widetilde{O}(n^{-1/2} + \sigma_0^{1.5}) \big) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) + \widetilde{O}(\sigma_0^{2.5}) \\
&= \bigg( \frac{\mathbb{E}[\alpha^3]}{2KM^2} + \widetilde{O}(d^{-0.005}) \bigg) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) + \widetilde{O}(\sigma_0^{2.5})
\end{aligned}
$$

where the second equality is due to Lemma E.3 and D.2, the third equality is due to Lemma E.4, the last equality is by the choice of $n$ and $\sigma_0$. Finally we compute the inner product $\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}, \boldsymbol{\xi}_{i,p} \rangle$ as follows

$$
\begin{aligned}
\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle &= -\frac{1}{n} \mathbb{1}(m_{i,t} = m) \ell_{i,t}' \pi_m(\mathbf{x}_i; \boldsymbol{\theta}^{(t)}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) \|\boldsymbol{\xi}_{i,p}\|_2^2 + \widetilde{O}(\sigma_0 d^{-1/2}) \\
&= \widetilde{O}\bigg( \frac{\|\boldsymbol{\xi}_{i,p}\|_2^2}{n} \bigg) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) + \widetilde{O}(\sigma_0 d^{-1/2}) \\
&= \widetilde{O}(d^{-0.005}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) + \widetilde{O}(\sigma_0^{2.5}),
\end{aligned}
$$

where the first equality is due to Lemma D.2, second equality is due to $|\ell_{i,t}'| \leq 1, \pi_m \in [0,1]$ and the third equality is due to Lemma D.2 and our choice of $n, \sigma_0$. Based on previous results, let $B$ be the projection matrix on the linear space spanned by $\{\mathbf{v}_k\}_{k \in [K]} \cup \{\mathbf{c}_k\}_{k \in [K]}$. We can verify that

$$
\begin{aligned}
\|\nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}\|_2 &\leq \|B \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}\|_2 + \|(I - B) \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}\|_2 \\
&\leq \sum_{k \in [K]} \frac{\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2} \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) + \sum_{k \in [K]} \widetilde{O}(d^{-0.005}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) \\
&\quad + \sum_{i \in [n], p \geq 4} \widetilde{O}(d^{-0.005}) \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle) + \widetilde{O}(\sigma_0^{2.5}).
\end{aligned}
$$

□

Because we use normalized gradient descent, all the experts get trained at the same speed. Following lemma shows that expert $m$ will focus on the signal $\mathbf{v}_{k_m^*}$.

**Lemma E.6.** For all $m \in [M]$ and $t \leq T_1$, we have following inequalities hold,

$$\begin{aligned}
\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle &= O(\sigma_0^{0.5}), \\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle &= \widetilde{O}(\sigma_0), \forall (j,k) \neq (j_m^*, k_m^*), \\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], k \in [K], \\
\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], i \in [n], p \geq 4.
\end{aligned}$$

*Proof.* For $t \leq T_1$, the update rule of every expert could be written as,

$$\begin{aligned}
\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{v}_k \rangle &= \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \bigg[ \frac{3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2} \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle^2 + \widetilde{O}(\sigma_0^{2.5}) \bigg], \\
\langle \mathbf{w}_{m,j}^{(t+1)}, \boldsymbol{\xi}_{i,p} \rangle &= \langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \big[ \widetilde{O}(d^{-0.005}) \langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle^2 + \widetilde{O}(\sigma_0^{2.5}) \big], \\
\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{c}_k \rangle &= \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \big[ \widetilde{O}(d^{-0.005}) \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle^2 + \widetilde{O}(\sigma_0^{2.5}) \big]. \qquad \text{(E.3)}
\end{aligned}$$

For $t \leq T_1$, we have that $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle \leq O(\sigma_0^{0.5})$. By comparing the update rule of $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle$ and other inner product presented in (E.3) , We can prove that $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle$ will grow to $\sigma_0^{0.5}$ while other inner product still remain nearly unchanged.

**Comparison with $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle$.** Consider $k \neq k_m^*$. We want to get an upper bound of $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle$, so without loss of generality we can assume $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle = \Omega(\sigma_0)$. Since $\sigma_0 \leq d^{-0.01}$, we have that $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle^2 + \widetilde{O}(\sigma_0^{2.5}) = (1 + \widetilde{O}(d^{-0.005})) \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle^2$. Therefore, we have that

$$\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{v}_{k_m^*} \rangle = \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \frac{3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2} \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle^2, \qquad \text{(E.4)}$$

$$\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{v}_k \rangle = \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \frac{3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2} \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle^2. \qquad \text{(E.5)}$$

Applying Lemma F.2 by choosing $C_t = (3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005}))/(2KM^2 \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F)$, $S = 1 + \widetilde{O}(d^{-0.005})$, $G = 1/(3\log(d)M^2)$ and verifying $\langle \mathbf{w}_m^{(0)}, \mathbf{v}_{k_m^*} \rangle \geq S(1 + G^{-1}) \langle \mathbf{w}_m^{(0)}, \mathbf{v}_k \rangle$ (events in Section D hold), we have that $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle \leq O(G^{-1}\sigma_0) = \widetilde{O}(\sigma_0)$.

**Comparison with $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle$.** We want to get an upper bound of $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle$, so without loss of generality we can assume $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle = \Omega(\sigma_0)$. Because $\sigma_0 \leq d^{-0.01}$, one can easily show that

$$\begin{aligned}
\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{v}_{k_m^*} \rangle &= \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \frac{3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005})}{2KM^2} \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle^2, \\
\langle \mathbf{w}_{m,j}^{(t+1)}, \mathbf{c}_k \rangle &\leq \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \widetilde{O}(d^{-0.01}) \langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle^2.
\end{aligned}$$

Again, applying Lemma F.2 by choosing $C_t = (3\mathbb{E}[\alpha^3] + \widetilde{O}(d^{-0.005}))/(2KM^2 \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F)$, $S = \widetilde{O}(d^{-0.01})$, $G = 2$ and verifying $\langle \mathbf{w}_m^{(0)}, \mathbf{v}_{k_m^*} \rangle \geq S(1 + G^{-1}) \langle \mathbf{w}_m^{(0)}, \mathbf{c}_k \rangle$ (events in Section D hold), we have that $\langle \mathbf{w}^{(t)}, \mathbf{v}_k \rangle \leq O(G^{-1}\sigma_0) = \widetilde{O}(\sigma_0)$.

**Comparison with $\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle$.** The proof is exact the same as the one with $\mathbf{c}_k$. □

Denote the iteration $T^{(m)}$ as the first time that $\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F \geq \sigma_0^{1.8}$. Then Following lemma gives an upper bound of $T^{(m)}$ for all $m \in \mathcal{M}$.

**Lemma E.7.** For all $m \in [M]$, we have that $T^{(m)} = \widetilde{O}(\eta^{-1}\sigma_0^{0.8})$ and thus $T^{(m)} < 0.01T_1$. Besides, for all $T_m < t \leq T_1$ we have that

$$\langle \nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}, \mathbf{v}_{k_m^*} \rangle \geq (1 - \sigma_0^{0.1}) \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F.$$

*Proof.* Let projection matrix $B = \mathbf{v}_{k_m^*} \mathbf{v}_{k_m^*}^\top \in \mathbb{R}^{d \times d}$, then we can divide the gradient into two orthogonal part

$$\begin{aligned}
\|\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 &= \|B\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)} + (I - B)\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 \\
&\leq \|B\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 + \|(I - B)\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2
\end{aligned}$$

Recall that

$$\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)} = \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \mathbf{x}_i^{(p)},$$

So we have that

$$\begin{aligned}
\|(I - B)\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 &= \left\| \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{x}_i^{(p)} \rangle)(I - B)\mathbf{x}_i^{(p)} \right\|_2 \\
&\leq \frac{1}{n} \sum_{i,p} \left\| \sigma'(\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{x}_i^{(p)} \rangle)(I - B)\mathbf{x}_i^{(p)} \right\|_2 \\
&\leq \widetilde{O}(\sigma_0^2),
\end{aligned}$$

where the first inequality is by $|\ell'_{i,t}| \leq 1, \pi_m \in [0, 1]$ and the second equality is because

1. when $\mathbf{x}_i^{(p)}$ align with $\mathbf{v}_{k_m^*}$, $(I - B)\mathbf{x}_i^{(p)} = \mathbf{0}$.
2. when $\mathbf{x}_i^{(p)}$ doesn't align with $\mathbf{v}_{k_m^*}$, $\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{x}_i^{(p)} \rangle = \widetilde{O}(\sigma_0)$.

Therefore, we have that

$$\|\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 \leq \|B\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 + \widetilde{O}(\sigma_0^2) = \langle \nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \widetilde{O}(\sigma_0^2).$$

We next compute the gradient of the neuron $\mathbf{w}_{m,j}, j \neq j_m^*$,

$$\|\nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}\|_2 = \left\| \frac{1}{n} \sum_{i,p} \mathbb{1}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) y_i \sigma'(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \mathbf{x}_i^{(p)} \right\|_2 = \widetilde{O}(\sigma_0^2), \tag{E.6}$$

where the inequality is by $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle = \widetilde{O}(\sigma_0), \forall j \neq j_m^*$ which is due to Lemma E.6. Now we can upper bound the gradient norm,

$$\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F \leq \sum_{j \in [J]} \|\nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}\|_2 \leq \|\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 + \widetilde{O}(\sigma_0^2). \tag{E.7}$$

When $\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F \geq \sigma_0^{1.8}$, it is obviously that

$$\langle \nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}, \mathbf{v}_{k_m^*} \rangle \geq \|\nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}\|_2 - \widetilde{O}(\sigma_0^2) \geq \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F - \widetilde{O}(\sigma_0^2) \geq (1 - \sigma_0^{0.1}) \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F,$$

where the first inequality is by (E.6) and the second inequality is by (E.7). Now let us give an upper bound for $T^{(m)}$. During the period $t \leq T^{(m)}$, $\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F < \sigma_0^{1.8}$. On the one hand, by Lemma E.5 we have that

$$\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_2 \geq -\langle \nabla_{\mathbf{w}_{m,j}} \mathcal{L}^{(t)}, \mathbf{v}_{k_m^*} \rangle = \frac{3\mathbb{E}[\alpha^3] - \widetilde{O}(d^{-0.005})}{2KM^2} [\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle]^2 - \widetilde{O}(\sigma_0^{2.5})$$

which implies that the inner product $\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle \leq \widetilde{O}(\sigma_0^{0.9})$. On the other hand, by Lemma E.6 we have that

$$\begin{aligned}
\langle \mathbf{w}_{m,j_m^*}^{(t+1)}, \mathbf{v}_{k_m^*} \rangle &\geq \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \frac{\eta}{\|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F} \Theta(\frac{1}{KM^2}) \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle^2 \\
&\geq \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \Theta\Big(\frac{\eta}{KM^2 \sigma_0^{1.8}}\Big) \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle^2 \\
&\geq \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle + \Theta\Big(\frac{\eta}{KM^2 \sigma_0^{0.8}}\Big) \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle,
\end{aligned}$$

where last inequality is by $\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle \geq 0.1\sigma_0$. Therefore, we have that the inner product $\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle$ grows exponentially and will reach $\widetilde{O}(\sigma_0^{0.9})$ within $\widetilde{O}(\eta^{-1}\sigma_0^{0.8})$ iterations.

□

Recall that $T_1 = \lfloor \eta^{-1}\sigma_0^{0.5} \rfloor$, following Lemma shows that the expert $m \in [M]$ only learns one feature during the first stage,

**Lemma E.8.** For all $t \leq T_1, m \in [M]$, we have that

$$\begin{aligned}
\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle &= O(\sigma_0^{0.5}), \\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle &= \widetilde{O}(\sigma_0), \forall (j,k) \neq (j_m^*, k_m^*), \\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], k \in [K], \\
\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], i \in [n], p \geq 4.
\end{aligned}$$

Besides $\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle \geq (1 - \sigma_0^{0.1})\eta t$, for all $t \geq T_1/2$.

*Proof.* By Lemma E.7, we have $T^{(m)} = \widetilde{O}(\eta^{-1}\sigma_0^{0.8}) < \sigma_0^{0.2} \cdot T_1$. Notice that $\langle \nabla_{\mathbf{w}_{m,j_m^*}} \mathcal{L}^{(t)}, \mathbf{v}_{k^*} \rangle \geq (1 - \sigma_0^{0.1}) \|\nabla_{\mathbf{W}_m} \mathcal{L}^{(t)}\|_F$, for all $T_m \leq t \leq T_1$. Therefore, we have that

$$\langle \mathbf{w}_{m,j_m^*}^{(t+1)}, \mathbf{v}_{k_m^*} \rangle \geq \langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle + (1 - \sigma_0^{0.1})\eta, \forall T_m \leq t \leq T_1,$$

which implies $\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle \geq (1 - O(\sigma_0^{0.1}))\eta t, \forall t \geq T_1/2$. Finally, applying Lemma E.6 completes the proof.

□

### E.2 Router Learning Stage

Denote $T_2 = \lfloor \eta^{-1} M^{-2} \rfloor$, The second stage ends when $t = T_2$. Given $\mathbf{x} = [\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma \epsilon \mathbf{v}_{k'}, \boldsymbol{\xi}]$, we denote by $\bar{x} = [\mathbf{0}, \beta \mathbf{c}_k, \mathbf{0}, \ldots, \mathbf{0}]$ the one only keeps cluster-center signal and denote by $\widehat{x} = [\alpha y \mathbf{v}_k, \mathbf{0}, \gamma \epsilon \mathbf{v}_{k'}, \mathbf{0}]$ the one that only keeps feature signal and feature noise.

For all $T_1 \leq t \leq T_2$, we will show that the router only focuses on the cluster-center signals and the experts only focus on the feature signals, i.e., we will prove that $|f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})|$ and $\|\mathbf{h}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \mathbf{h}(\bar{\mathbf{x}}_i, \boldsymbol{\Theta}^{(t)})\|_\infty$ are small. In particular, We claim that for all $T_1 \leq t \leq T_2$, following proposition holds.

**Proposition E.9.** For all $T_1 \leq t \leq T_2$, following inequalities hold,

$$|f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})| \leq O(d^{-0.001}), \forall m \in [M], i \in [n], \tag{E.8}$$

$$\|\mathbf{h}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \mathbf{h}(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)})\|_\infty \leq O(d^{-0.001}), \forall i \in [n], \tag{E.9}$$

$$\mathbb{P}(m_{i,t} = m), \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) = \Omega(1/M), \forall m \in [M], i \in \Omega_{k_m^*}. \tag{E.10}$$

Proposition E.9 implies that expert will only focus on the label signal and router will only focus on the cluster-center signal. We will prove Proposition E.9 by induction. Before we move into the detailed proof of Proposition E.9, we will first prove some important lemmas.

**Lemma E.10.** For all $T_1 \leq t \leq T_2$, the neural network parameter maintains following property.

- $|f_m(\mathbf{x}_i; \mathbf{W}^{(t)})| = O(1), \forall m \in [M]$,
- $\pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) = \Omega(1/M), \forall i \in [n]$.

*Proof.* Because we use normalized gradient descent, the first bullet would be quite straight forward.

$$|f_m(\mathbf{x}_i, \mathbf{W}^{(t)})| = \sum_{j\in[J]} \sum_{p\in[P]} \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{x}_i^{(p)} \rangle) \overset{(i)}{=} O(1),$$

where (i) is by $\|\mathbf{w}_{m,j}^{(t)} - \mathbf{w}_{m,j}^{(0)}\|_2 = O(\eta T_2) = O(M^{-2})$ and $\mathbf{x}_i^{(p)} = O(1)$.

Now we prove the second bullet. By Lemma C.4, we have that $h_{m_{i,t}}(\mathbf{x}; \mathbf{\Theta}) \geq \max_m h_m(\mathbf{x}; \mathbf{\Theta}) - 1$, which implies that

$$\pi_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) = \frac{\exp(h_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}))}{\sum_m \exp(h_m(\mathbf{x}; \mathbf{\Theta}^{(t)}))} \geq \frac{\exp(h_{m_{i,t}}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}))}{M \max_m \exp(h_m(\mathbf{x}; \mathbf{\Theta}^{(t)}))} \geq \frac{1}{eM}.$$

□

**Lemma E.11.** Denote $\delta_{\mathbf{\Theta}} = \max_i \|\mathbf{h}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) - \mathbf{h}(\mathbf{x}_i; \mathbf{\Theta})\|_\infty$ and let the random variable $\bar{m}_{i,t}$ be expert that get routed if we use the gating network output $\mathbf{h}(\bar{\mathbf{x}}_i; \mathbf{\Theta}^{(t)})$ instead. Then we have following inequalities,

$$|\pi_m(\mathbf{x}_i; \mathbf{\Theta}) - \pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta})| = O(\delta_{\mathbf{\Theta}}), \forall m \in [M], i \in [n], . \tag{E.11}$$

$$|\mathbb{P}(m_{i,t} = m) - \mathbb{P}(\bar{m}_{i,t} = m)| = O(M^2 \delta_{\mathbf{\Theta}}), \forall m \in [M], i \in [n]. \tag{E.12}$$

*Proof.* By definition of $\delta_{\mathbf{\Theta}}$, we have that $\|\mathbf{h}(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) - \mathbf{h}(\bar{\mathbf{x}}_i; \mathbf{\Theta}^{(t)})\|_\infty \leq \delta_{\mathbf{\Theta}}$. Then applying Lemma 5.1 gives $|\mathbb{P}(m_{i,t} = m) - \mathbb{P}(\bar{m}_{k,t} = m)| = \widetilde{O}(\delta_{\mathbf{\Theta}}), \forall m \in [M], i \in [n]$, which completes the proof for (E.12).

Next we prove (E.11), which needs more effort. For all $i \in [n]$, we have

$$\pi_m(\mathbf{x}_i; \mathbf{\Theta}) = \frac{\pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \exp(h_m(\mathbf{x}_i; \mathbf{\Theta}) - h_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}))}{\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \exp(h_{m'}(\mathbf{x}_i; \mathbf{\Theta}) - h_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}))}.$$

Let $\delta_{m'} = \exp(h_{m'}(\mathbf{x}_i; \mathbf{\Theta}) - h_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta})) = 1 + O(\delta_{\mathbf{\Theta}})$. Then for sufficiently small $\delta_{\mathbf{\Theta}}$, we have that $\delta_{m'} \geq 0.5$ . Then we can further compute

$$\begin{aligned}
|\pi_m(\mathbf{x}_i; \mathbf{\Theta}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta})| &= \pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \left| \frac{\delta_m}{\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \delta_{m'}} - 1 \right| \\
&= \pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \frac{|\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta})(\delta_{m'} - \delta_m)|}{\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \delta_{m'}} \\
&\leq \pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \frac{\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) |\delta_{m'} - \delta_m|}{\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \delta_{m'}} \\
&\leq O(\delta_{\mathbf{\Theta}}),
\end{aligned}$$

where the last inequality is by $|\delta_{m'} - \delta_m| \leq O(\delta_{\mathbf{\Theta}})$, $\pi_m(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \leq 1$ and $\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta}) \delta_{m'} \geq [\sum_{m'} \pi_{m'}(\bar{\mathbf{x}}_i; \mathbf{\Theta})]/2 = 0.5$. □

Following Lemma implies that the pattern learned by experts during the first stage won't change in the second stage.

**Lemma E.12.** Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, then we have following inequalities hold for all $t \in [T_1, T + 1]$,

$$\begin{aligned}
&\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle \geq (1 - O(\sigma_0^{0.1}))\eta t, \\
&\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle = \widetilde{O}(\sigma_0), \forall (j,k) \neq (j_m^*, k_m^*), \\
&\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle = \widetilde{O}(\sigma_0), \forall j \in [J], k \in [K], \\
&\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle = \widetilde{O}(\sigma_0), \forall j \in [J], k \in [K], i \in [n], p \geq 4.
\end{aligned}$$

*Proof.* Most of the proof exactly follows the proof in the first stage, so we only list some key steps here. Recall that

$$\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)} = \frac{1}{n}\sum_{i,p}\mathbb{1}(m_{i,t}=m)\ell'_{i,t}\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)})y_i\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\mathbf{x}_i^{(p)}\rangle)\mathbf{x}_i^{(p)}.$$

In the proof of Lemma E.5, we do Taylor expansion at the zero point. Now we will do Taylor expansion at $f_m(\widehat{\mathbf{x}}_i;\mathbf{W})$ and $\pi(\bar{\mathbf{x}}_i;\mathbf{\Theta})$ as follows,

$$\begin{aligned}
&|\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)})f_m(\mathbf{x}_i;\mathbf{W}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i;\mathbf{\Theta}^{(t)})f_m(\widehat{\mathbf{x}}_i;\mathbf{W}^{(t)})|\\
&\le |\pi_m(\bar{\mathbf{x}}_i;\mathbf{\Theta}^{(t)})[f_m(\mathbf{x}_i;\mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i;\mathbf{W}^{(t)})]| + |[\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i;\mathbf{\Theta}^{(t)})]f_m(\mathbf{x}_i;\mathbf{W}^{(t)})|\\
&\le |f_m(\mathbf{x}_i;\mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i;\mathbf{W}^{(t)})| + O(|\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i;\mathbf{\Theta}^{(t)})|)\\
&\le O(d^{-0.001}),
\end{aligned}$$

where the first inequality is by triangle inequality, the second inequality is by $\pi_m(\bar{\mathbf{x}}_i;\mathbf{\Theta}^{(t)}) \le 1$ and $|f_m(\mathbf{x}_i;\mathbf{W}^{(t)})| = O(1)$ in Lemma E.10, the third inequality is by (E.8), (E.9) and (E.11).

Then follow the proof of Lemma E.5, we have that

$$\begin{aligned}
\mathbb{E}[\langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)},\mathbf{v}_{k_m^*}\rangle] &= -\frac{1}{n}\sum_{i\in\Omega_{k_m^*}}\mathbb{P}(m_{i,t}=m)\ell'_{i,t}\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)})\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle)\alpha_i^3\|\mathbf{v}_{k_m^*}\|_2^2\\
&\quad - \frac{1}{n}\sum_{i\in\Omega_{k',k_m^*}}\mathbb{P}(m_{i,t}=m)\ell'_{i,t}\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)})\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle)\gamma_i^3 y_i\epsilon_i\|\mathbf{v}_{k_m^*}\|_2^2\\
&\quad - \frac{1}{n}\sum_{i,p}\mathbb{P}(m_{i,t}=m)\ell'_{i,t}\pi_m(\mathbf{x}_i;\mathbf{\Theta}^{(t)})\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\boldsymbol{\xi}_{i,p}\rangle)y_i\langle\mathbf{v}_{k_m^*},\boldsymbol{\xi}_{i,p}\rangle\\
&= \Bigg[-\widetilde{\Theta}\Big(\frac{1}{n}\Big)\sum_{i\in\Omega_{k_m^*}}\mathbb{P}(m_{i,t}=m)\alpha_i^3 - \widetilde{\Theta}\Big(\frac{1}{n}\Big)\sum_{i\in\Omega_{k',k_m^*}}\mathbb{P}(m_{i,t}=m)\gamma_i^3 y_i\epsilon_i\\
&\quad + O(d^{-0.001})\Bigg]\cdot\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle) + \widetilde{O}(d^{-1/2})\\
&\overset{(i)}{=} -\widetilde{\Theta}(1)\sigma'(\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle),
\end{aligned}$$

where (i) is due to (E.10): $\mathbb{P}(m_{i,t}=m) \ge \Theta(1/M), \forall i\in\Omega_{k_m^*}, m\in[M]$. Again follow Lemma E.5 and Lemma E.6, we further have that

$$\begin{aligned}
\langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)},\mathbf{v}_k\rangle &= -\widetilde{\Theta}(1)[\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_k\rangle]^2,\\
\langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)},\mathbf{c}_k\rangle &= \widetilde{O}(1)[\langle\mathbf{w}_{m,j}^{(t)},\mathbf{c}_k\rangle]^2,\\
\langle\nabla_{\mathbf{w}_{m,j}}\mathcal{L}^{(t)},\boldsymbol{\xi}_{i,p}\rangle &= \widetilde{O}(1)[\langle\mathbf{w}_{m,j}^{(t)},\boldsymbol{\xi}_{i,p}\rangle]^2.
\end{aligned}$$

Thus for all $T_1 \le t \le T$, the update rule of every expert could be written as,

$$\begin{aligned}
\langle\mathbf{w}_{m,j}^{(t+1)},\mathbf{v}_{k_m^*}\rangle &= \langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle + \widetilde{\Theta}(1)\frac{\eta}{\|\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}\|_F}\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_{k_m^*}\rangle^2\\
\langle\mathbf{w}_{m,j}^{(t+1)},\mathbf{v}_k\rangle &= \langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_k\rangle + \widetilde{O}(1)\frac{\eta}{\|\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}\|_F}\langle\mathbf{w}_{m,j}^{(t)},\mathbf{v}_k\rangle^2\\
\langle\mathbf{w}^{(t+1)},\boldsymbol{\xi}_{i,p}\rangle &= \langle\mathbf{w}^{(t)},\boldsymbol{\xi}_{i,p}\rangle + \widetilde{O}(1)\frac{\eta}{\|\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}\|_F}\langle\mathbf{w}^{(t)},\boldsymbol{\xi}_{i,p}\rangle^2\\
\langle\mathbf{w}_{m,j}^{(t+1)},\mathbf{c}_k\rangle &= \langle\mathbf{w}_{m,j}^{(t)},\mathbf{c}_k\rangle + \widetilde{O}(1)\frac{\eta}{\|\nabla_{\mathbf{W}_m}\mathcal{L}^{(t)}\|_F}\langle\mathbf{w}_{m,j}^{(t)},\mathbf{c}_k\rangle^2.
\end{aligned}$$

By the first stage of training we have that $\langle \mathbf{w}_{m,j}^{(T_1)}, \mathbf{v}_{k_m^*} \rangle = \Theta(\sigma_0^{0.5})$, while others remains $\widetilde{O}(\sigma_0)$. Then we can use Lemma F.2, by choosing $S = \widetilde{\Theta}(1)$ and $G = 2$, then we have that

$$\begin{aligned}
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k_m^*} \rangle &= O(1).\\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k} \rangle &= \widetilde{O}(\sigma_0), \forall k \neq k_m^*.\\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_{k} \rangle &= \widetilde{O}(\sigma_0).\\
\langle \mathbf{w}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle &= \widetilde{O}(\sigma_0).
\end{aligned}$$

Then following Lemma E.7 and E.8, we can prove that for all $T_1 \leq t \leq T+1$, $m \in [M]$,

$$\begin{aligned}
\langle \mathbf{w}_{m,j_m^*}^{(t)}, \mathbf{v}_{k_m^*} \rangle &\geq (1 - O(\sigma_0^{0.1}))\eta t,\\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k} \rangle &= \widetilde{O}(\sigma_0), \forall (j,k) \neq (j_m^*, k_m^*),\\
\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_{k} \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], k \in [K],\\
\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle &= \widetilde{O}(\sigma_0), \forall j \in [J], i \in [n], p \geq 4.
\end{aligned}$$

□

By the result of expert training we have following results

**Lemma E.13.** Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, then we have that $|f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})| = \widetilde{O}(\sigma_0^3)$ for all $m \in [M]$ and $i \in [n]$, $t \in [T_1, T+1]$. Besides,

$$\begin{aligned}
y_i f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)}) &= \sum_{j \in [J]} \Big[\alpha_i^3 \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) + \gamma_i^3 \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k'} \rangle)\Big], \forall i \in \Omega_{k,k'}^+, m \in [M],\\
y_i f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)}) &= \sum_{j \in [J]} \Big[\alpha_i^3 \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_k \rangle) - \gamma_i^3 \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{v}_{k'} \rangle)\Big], \forall i \in \Omega_{k,k'}^-, m \in [M].
\end{aligned}$$

*Proof.* For all $i \in \Omega_k$, we have that

$$\begin{aligned}
\big|f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})\big| &\leq \Big|\sum_{j \in [J]} \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle)\Big| + \Big|\sum_{j \in [J], p \geq 4} \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle)\Big|\\
&\leq O(J) \cdot \max_{k,j} \sigma(\langle \mathbf{w}_{m,j}^{(t)}, \mathbf{c}_k \rangle) + O(J) \cdot \max_{i,j,p} |\sigma(\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle)|\\
&= \widetilde{O}(\sigma_0^3),
\end{aligned}$$

where the first inequality is by triangle inequality and the last equality is by Lemma E.12. □

Next we will show that router only focus on the cluster-center signal rather than the label signal during the router training.

**Lemma E.14.** Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, then we have that $\|\mathbf{h}(\bar{\mathbf{x}}_i, \boldsymbol{\Theta}^{(t)}) - \mathbf{h}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})\|_\infty = \widetilde{O}(d^{-0.005})$ hold for all $i \in [n]$ and $t \in [T_1, T+1]$. Besides, we have that $\max_{m,k} |\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{v}_k \rangle|, \max_{m,i,p} |\langle \boldsymbol{\theta}_m^{(t)}, \boldsymbol{\xi}_{i,p} \rangle| = \widetilde{O}(d^{-0.005})$ for all $t \in [T_1, T+1]$.

*Proof.* Recall the definition of $\delta_{\boldsymbol{\Theta}}$ in Lemma E.11, we need to show that $\delta_{\boldsymbol{\Theta}^{(t)}} = \widetilde{O}(d^{-0.005})$ for all $t \in [T_1, T+1]$. We first prove following router parameter update rules,

$$\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v}_k \rangle = O(\delta_{\boldsymbol{\Theta}^{(t)}} K^2) + \widetilde{O}(d^{-0.005}), \langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \boldsymbol{\xi}_{i,p} \rangle = \widetilde{O}(d^{-0.005}), \tag{E.13}$$

for all $T_1 \leq t \leq T$, $m \in [M]$, $k \in [K]$, $i \in [n]$ and $p \geq 4$.

Consider the inner product of the router gradient and the feature vector and we have

$$
\begin{aligned}
&\mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{v}_k \rangle] \\
&= \underbrace{\frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) y_i \alpha_i}_{I_1} \\
&\quad + \underbrace{\frac{1}{n} \sum_{i \in \Omega_{k',k}} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \epsilon_i \gamma_i}_{I_2} \\
&\quad - \underbrace{\frac{1}{n} \sum_{i \in \Omega_k, m' \in [M]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} y_i \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)}) y_i \alpha_i}_{I_3} \\
&\quad - \underbrace{\frac{1}{n} \sum_{i \in \Omega_{k',k}, m' \in [M]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)}) \epsilon_i \gamma_i}_{I_4} \\
&\quad + \underbrace{\frac{1}{n} \sum_{i \in [n], p \geq 4} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{v}_k \rangle}_{I_5} \\
&\quad - \underbrace{\frac{1}{n} \sum_{i \in [n], p \geq 4, m' \in [M]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} y_i \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_{m'}(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{v}_k \rangle}_{I_6}.
\end{aligned} \tag{E.14}
$$

Denote $y_i \pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)}) f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)}), \forall i \in \Omega^+_{k,k'}$ by $\bar{F}^+_{k,k'}$. We next show that the output of the MoE multiplied by label: $y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}), \forall i \in \Omega^+_{k,k'}$ can be approximated by $\bar{F}^+_{k,k'}$.

$$
\begin{aligned}
&|\pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)}) f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})| \\
&\leq |[\pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)})] f_m(\mathbf{x}_i; \mathbf{W}^{(t)})| + |\pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)})[f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})]| \\
&\leq O(|\pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)})|) + |f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) - f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)})| \\
&\leq O(\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(\sigma_0^3),
\end{aligned}
$$

where the first inequality is by triangle inequality, the second inequality is by $\pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)}) \leq 1$ and $|f_m(\mathbf{x}_i; \mathbf{W}^{(t)})| = O(1)$ in Lemma E.10, the third inequality is by (E.11) and Lemma E.13.

Similarly, denote $y_i \pi_m(\bar{\mathbf{x}}_i; \boldsymbol{\Theta}^{(t)}) f_m(\widehat{\mathbf{x}}_i; \mathbf{W}^{(t)}), i \in \Omega^-_{k,k'}$ by $\bar{F}^-_{k,k'}$ and we can show that value $y_i \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) f_m(\mathbf{x}_i; \mathbf{W}^{(t)}), \forall i \in \Omega^-_{k,k'}$ can be approximated by $\bar{F}^-_{k,k'}$. Now we can bound $I_1$ as

follows,

$$
\begin{aligned}
I_1 &= \sum_{k'\neq k} \frac{\ell'(\bar{F}_{k,k'+})\bar{F}^+_{k,k'}}{n} \sum_{i\in\Omega^+_{k,k'}} \big[\mathbb{P}(m_{i,t}=m)y_i\alpha_i + O(\delta_{\boldsymbol{\Theta}^{(t)}})\big] + \widetilde{O}(\sigma_0^3) \\
&\quad + \sum_{k'\neq k} \frac{\ell'(\bar{F}_{k,k'-})\bar{F}^-_{k,k'}}{n} \sum_{i\in\Omega^-_{k,k'}} \big[\mathbb{P}(m_{i,t}=m)y_i\alpha_i + O(\delta_{\boldsymbol{\Theta}^{(t)}})\big] + \widetilde{O}(\sigma_0^3) \\
&\overset{(i)}{=} \sum_{k'\neq k} \frac{\ell'(\bar{F}_{k,k'+})\bar{F}^+_{k,k'}}{n} \sum_{i\in\Omega^+_{k,k'}} \big[\mathbb{P}(\bar{m}_{i,t}=m)y_i\alpha_i + O(M^2\delta_{\boldsymbol{\Theta}^{(t)}})\big] + \widetilde{O}(\sigma_0^3) \\
&\quad + \sum_{k'\neq k} \frac{\ell'(\bar{F}_{k,k'-})\bar{F}^-_{k,k'}}{n} \sum_{i\in\Omega^-_{k,k'}} \big[\mathbb{P}(\bar{m}_{i,t}=m)y_i\alpha_i + O(M^2\delta_{\boldsymbol{\Theta}^{(t)}})\big] + \widetilde{O}(\sigma_0^3) \\
&\overset{(ii)}{=} O(M^2\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(n^{-1/2}+\sigma_0^3) \\
&= O(M^2\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(d^{-0.005})
\end{aligned}
$$

where (i) is due to (E.12) and (ii) is by $\sum_{i\in\Omega^+_{k,k'}} y_i\alpha = \widetilde{O}(\sqrt{n})$ and $\sum_{i\in\Omega^-_{k,k'}} y_i\alpha = \widetilde{O}(\sqrt{n})$ in Lemma D.1. Similarly we can prove that $I_2, I_3, I_4 = O(M^2\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(d^{-0.005})$. Since $\langle \mathbf{x}_i^{(p)}, \mathbf{v}_i\rangle = \widetilde{O}(d^{-1/2}), \forall p \geq 4$, $\pi_m, \pi_{m_{i,t}} \leq 1$ and $f_{m_{i,t}} = O(1)$, we can upper bound $I_5, I_6$ by $\widetilde{O}(d^{-1/2})$. Plugging those bounds into the gradient computation (E.14) gives

$$
\mathbb{E}[\big\langle \nabla_{\boldsymbol{\theta}_m}\mathcal{L}^{(t)}, \mathbf{v}_k\big\rangle] = O(M^2\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(d^{-0.005}).
$$

We finally consider the alignment between router gradient and noise

$$
\begin{aligned}
\big\langle \nabla_{\boldsymbol{\theta}_m}\mathcal{L}^{(t)}, \boldsymbol{\xi}_{i',p'}\big\rangle &= \frac{1}{n}\sum_{i\in[n],p\geq 4} \mathbb{1}(m_{i,t}=m)\ell'_{i,t}y_i\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\langle \mathbf{x}_i^{(p)}, \boldsymbol{\xi}_{i',p'}\rangle \\
&\quad - \frac{1}{n}\sum_{i\in[n],p\geq 4} \ell'_{i,t}y_i\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\langle \mathbf{x}_i^{(p)}, \boldsymbol{\xi}_{i',p'}\rangle. \\
&\overset{(i)}{=} \widetilde{O}\bigg(\frac{1}{n}\bigg) + \widetilde{O}(d^{-1/2}) \\
&\overset{(ii)}{=} \widetilde{O}(d^{-1/2}),
\end{aligned}
$$

where the (i) is by considering the cases $(i',p') = \boldsymbol{\xi}_{i,p}$ and $\boldsymbol{\xi}_{i',p'} \neq \boldsymbol{\xi}_{i,p}$ respectively and (ii) is due to our choice of $n$. Now, we have completed the proof of (E.13).

Plugging the gradient estimation (E.13) in to the gradient update rule for the gating network (3.5) gives

$$
\max_{m,k} |\langle \boldsymbol{\theta}_m^{(t+1)}, \mathbf{v}_k\rangle| \leq \max_{m,k} |\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{v}_k\rangle| + O(\eta_r M^2\delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(\eta_r d^{-0.005}) \tag{E.15}
$$

$$
\max_{m,i,p} |\langle \boldsymbol{\theta}_m^{(t+1)}, \boldsymbol{\xi}_{i,p}\rangle| \leq \max_{m,i,p} |\langle \boldsymbol{\theta}_m^{(t)}, \boldsymbol{\xi}_{i,p}\rangle| + \widetilde{O}(\eta_r d^{-0.005}) \tag{E.16}
$$

Combining (E.15) and (E.16), we have that there exist $C_1 = O(M^2)$ and $C_2 = \widetilde{O}(d^{-0.005})$ such that $\delta_{\boldsymbol{\Theta}^{(t+1)}} \leq \delta_{\boldsymbol{\Theta}^{(t)}} + C_1\eta_r\delta_{\boldsymbol{\Theta}^{(t)}} + C_2\eta_r$. Therefore, we have that

$$
\begin{aligned}
\delta_{\boldsymbol{\Theta}^{(t+1)}} + C_1^{-1}C_2 &\leq (1+C_1\eta_r)[\delta_{\boldsymbol{\Theta}^{(t)}} + C_1^{-1}C_2] \\
&\leq \exp(C_1\eta_r)[\delta_{\boldsymbol{\Theta}^{(t)}} + C_1^{-1}C_2],
\end{aligned}
$$

where the last inequality is due to $\exp(z) \geq 1+z$ for all $z\in\mathbb{R}$. Then we further have that

$$
\delta_{\boldsymbol{\Theta}^{(t)}} \leq \exp(C_1\eta_r t)[\delta_{\boldsymbol{\Theta}^{(0)}} + C_1^{-1}C_2] \leq \exp(C_1\eta_r\eta^{-1}M^{-2})[\delta_{\boldsymbol{\Theta}^{(0)}} + C_1^{-1}C_2] = \widetilde{O}(d^{-0.005}),
$$

where the last equality is by $\eta_r = \Theta(M^2)\eta$. □

Define $\Delta_{\boldsymbol{\Theta}} := \max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} \max_{(\mathbf{x}_i,y_i)\in\Omega_k} |h_m(\mathbf{x}_i;\boldsymbol{\Theta}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta})|$, which measures the bias of the router towards different experts in the same $\mathcal{M}_k$. Following Lemma shows that the router will treats professional experts equally when $\Delta_{\boldsymbol{\theta}}$ is small.

**Lemma E.15.** For all $t \geq 0$, we have that following inequality holds,

$$\max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} \max_{(\mathbf{x}_i,y_i)\in\Omega_k} |\pi_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - \pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})| \leq 2\Delta_{\boldsymbol{\Theta}^{(t)}},$$
$$\max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} \max_{(\mathbf{x}_i,y_i)\in\Omega_k} |\mathbb{P}(m_{i,t} = m) - \mathbb{P}(m_{i,t} = m')| = O(M^2)\Delta_{\boldsymbol{\Theta}^{(t)}}.$$

*Proof.* By Lemma C.3, we directly have that

$$|\mathbb{P}(m_{i,t} = m) - \mathbb{P}(m_{i,t} = m')| \leq O(M^2)|h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})|.$$

Then, we prove that

$$|\pi_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}) - \pi_m(\mathbf{x}_i;\boldsymbol{\Theta})| \leq 2|h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})|. \tag{E.17}$$

When $|h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})| \geq 1$, it is obvious that (E.17) is true. When $|h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})| \leq 1$ we have that

$$\begin{aligned}
|\pi_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}) - \pi_m(\mathbf{x}_i;\boldsymbol{\Theta})| &= \left|\frac{\exp(h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})) - \exp(h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}))}{\sum_{m''}\exp\left(h_{m''}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\right)}\right| \\
&= \left|\frac{\exp(h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}))}{\sum_{m''}\exp\left(h_{m''}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\right)}\right| \cdot |\exp(h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})) - 1| \\
&\leq 2|h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})|,
\end{aligned}$$

which completes the proof of (E.17). □

Notice that the gating network is initialized to be zero, so we have $\Delta_{\boldsymbol{\Theta}} = 0$ at initialization. We can further show that $\Delta_{\boldsymbol{\Theta}} = O\big(1/\mathrm{poly}(d)\big)$ during the training up to time $T = \widetilde{O}(\eta^{-1})$.

**Lemma E.16.** Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, then we have that $\Delta_{\boldsymbol{\Theta}^{(t)}} \leq \widetilde{O}(d^{-0.001})$ holds for all $t \in [T_1, T + 1]$.

*Proof.* One of the key observation is the similarity of the m-th and the $m'$-th expert in the same expert class $\mathcal{M}_k$. Lemma E.12 implies that $\max_{i\in\Omega_k} |f_m(\mathbf{x}_i, \mathbf{W}^{(t)}) - f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)})| = \widetilde{O}(\sigma_0^{0.1}) \leq \widetilde{O}(d^{-0.001})$.

Another key observe is that, we only need to focus on the $k-th$ cluster-center signal. Lemma E.14 implies that,

$$\begin{aligned}
\Delta_{\boldsymbol{\Theta}^{(t)}} &= \max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} \max_{(\mathbf{x}_i,y_i)\in\Omega_k} |h_m(\mathbf{x}_i;\boldsymbol{\Theta}) - h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})| \\
&\leq \max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} \max_{(\mathbf{x}_i,y_i)\in\Omega_k} |h_m(\bar{\mathbf{x}}_i;\boldsymbol{\Theta}^{(t)}) - h_{m'}(\bar{\mathbf{x}}_i;\boldsymbol{\Theta}^{(t)})| + 2\delta_{\boldsymbol{\Theta}^{(t)}} \\
&= \max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} |\langle\boldsymbol{\theta}_m - \boldsymbol{\theta}_{m'}, \beta_i\mathbf{c}_k\rangle| + 2\delta_{\boldsymbol{\Theta}^{(t)}} \\
&\leq C_2 \max_{k\in[K]} \max_{m,m'\in\mathcal{M}_k} |\langle\boldsymbol{\theta}_m - \boldsymbol{\theta}_{m'}, \mathbf{c}_k\rangle| + 2\delta_{\boldsymbol{\Theta}^{(t)}},
\end{aligned}$$

where the first inequality is by Lemma E.14 and the second inequality is by $\beta_i \leq C_2$. We now prove that following gradient difference is small

$$
\begin{aligned}
&\big\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)} - \nabla_{\boldsymbol{\theta}_{m'}} \mathcal{L}^{(t)}, \mathbf{c}_k \big\rangle \\
&\overset{(i)}{=} \frac{1}{n} \sum_{i\in[n]} \sum_{p\in[P]} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{c}_k \rangle \\
&\quad - \frac{1}{n} \sum_{i\in[n]} \sum_{p\in[P]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_{m'}(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{c}_k \rangle \\
&\quad + \frac{1}{n} \sum_{i\in\Omega_k} \sum_{p\in[P]} \sum_{m''\in[M]} [\pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})] \mathbb{P}(m_{i,t} = m'') \ell'_{i,t} \pi_{m''}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \cdot \\
&\quad y_i f_{m''}(\mathbf{x}_i, \mathbf{W}) \langle \mathbf{x}_i^{(p)}, \mathbf{c}_k \rangle + \widetilde{O}(d^{-0.001}) \\
&= O\Big(\frac{1}{n}\Big) \sum_{i\in\Omega_k} [\mathbb{P}(m_{i,t} = m') - \mathbb{P}(m_{i,t} = m)] |\ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}) \beta_i y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)})| + \widetilde{O}(d^{-0.001}) \\
&\quad + O(1) \max_{i\in\Omega_k} |\pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})| + O(1) \max_{i\in\Omega_k} |f_m(\mathbf{x}_i, \mathbf{W}^{(t)}) - f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)})| \\
&= O(1) |\mathbb{P}(m_{i,t} = m') - \mathbb{P}(m_{i,t} = m)]| + O(1) \max_{i\in\Omega_k} |\pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})| \\
&\quad + O(1) \max_{i\in\Omega_k} |f_m(\mathbf{x}_i, \mathbf{W}^{(t)}) - f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)})| + \widetilde{O}(d^{-0.001}) \\
&\overset{(ii)}{=} O(M^2 \Delta_{\boldsymbol{\Theta}^{(t)}}) + \widetilde{O}(d^{-0.001}),
\end{aligned}
$$

where the (i) is by Lemma E.2 and (ii) is by Lemma E.15. It further implies that $\Delta_{\boldsymbol{\Theta}^{(t+1)}} \leq O(\eta_r M^2) \Delta_{\boldsymbol{\Theta}^{(t)}} + \widetilde{O}(\eta_r d^{-0.001})$. Following previous proof of $\delta_{\boldsymbol{\Theta}}$, we have that $\Delta_{\boldsymbol{\Theta}^{(T+1)}} = \widetilde{O}(d^{-0.001})$. □

Together with the key technique 1, we can infer that each expert $m \in \mathcal{M}_k$ will get nearly the same load as other experts in $\mathcal{M}_k$. Since $\Delta_{\boldsymbol{\Theta}}$ keeps increasing during the training, it cannot be bounded if we allow the total number of iterations goes to infinity in Algorithm 1. This is the reason that we require early stopping in Theorem 4.2, which we believe can be waived by adding load balancing loss (Eigen et al., 2013; Shazeer et al., 2017; Fedus et al., 2021), or advanced MoE layer structure such as BASE Layers (Lewis et al., 2021; Dua et al., 2021) and Hash Layers (Roller et al., 2021).

**Lemma E.17.** Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, then for $m \notin \mathcal{M}_k$ and $t \in [T_1, T]$ , if $\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{c}_k \rangle \geq \max_{m'} \langle \boldsymbol{\theta}_{m'}^{(t)}, \mathbf{c}_k \rangle - 1$ we have that

$$
\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle \geq \Omega\bigg( \frac{\eta^3 t^3}{K M^3} \bigg) + \widetilde{O}(d^{-0.005}).
$$

*Proof.* The expectation of the inner product $\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle$ can be computed as follows,

$$
\begin{aligned}
\mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle] &= \frac{1}{n} \sum_{i,p} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{c}_k \rangle \\
&\quad - \frac{1}{n} \sum_{i,p,m'} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)}) \langle \mathbf{x}_i^{(p)}, \mathbf{c}_k \rangle \\
&\overset{(i)}{=} \frac{1}{n} \sum_{i\in\Omega_k} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) + \widetilde{O}(d^{-0.005}) \\
&\quad - \frac{1}{n} \sum_{i\in\Omega_k} \sum_{m'\in[M]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_{m'}(\mathbf{x}_i, \mathbf{W}).
\end{aligned} \tag{E.18}
$$

where (i) is due to $|\langle \boldsymbol{\xi}_{i,p}, \mathbf{c}_k \rangle| = \widetilde{O}(d^{-0.5})$.

We can rewrite the inner product (E.18) as follows,

$$\begin{aligned}
\mathbb{E}[\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{c}_k \rangle] &= \frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) + \widetilde{O}(d^{-0.005}) \\
&\quad - \frac{1}{n} \sum_{i \in \Omega_k} \sum_{m' \in [M]} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_{m'}(\mathbf{x}_i, \mathbf{W}) \\
&= \underbrace{\frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \beta_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)})}_{I_1} + \widetilde{O}(d^{-0.005}) \\
&\quad \underbrace{-\frac{1}{n} \sum_{i \in \Omega_k, m' \in \mathcal{M}_k} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)})}_{I_2} \qquad \text{(E.19)} \\
&\quad \underbrace{-\frac{1}{n} \sum_{i \in \Omega_k, m' \notin \mathcal{M}_k} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)})}_{I_3} . \qquad \text{(E.20)}
\end{aligned}$$

To calculate $I_1, I_2, I_3$, let's first lower bound $I_2$. We now consider the case that $m \notin \mathcal{M}_k, m' \in \mathcal{M}_k$. Because $\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{c}_k \rangle \geq \max_{m'} \langle \boldsymbol{\theta}_m^{(t)}, \mathbf{c}_k \rangle - 1$, we can easily prove that $\pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) = \Omega(1/M), \forall i \in \Omega_k$. Then we have that

$$\begin{aligned}
I_2 &= -\frac{1}{n} \sum_{i \in \Omega_k, m' \in \mathcal{M}_k} \mathbb{P}(m_{i,t} = m') \ell'_{i,t} \pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \beta_i y_i f_{m'}(\mathbf{x}_i, \mathbf{W}^{(t)}) \\
&\geq \Omega\Big(\frac{\eta^3 t^3}{nM^3}\Big) \sum_{i \in \Omega_k, m' \in \mathcal{M}_k} \beta_i \\
&\geq \Omega\Big(\frac{\eta^3 t^3}{KM^3}\Big),
\end{aligned}$$

where the first inequality is by $\pi_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) = \Omega(1/M)$, $\mathbb{P}(m_{i,t} = m') \geq \Theta(1/M)$, $\forall i \in \Omega_{k_m^*}, m \in [M]$, $y_i f_{m'}(\mathbf{x}_i; \mathbf{W}^{(t)}) = \eta^3 t^3 (1 - O(\sigma_0^{0.1}))$ and $\ell' = -\Theta(1)$ for all $i \in \Omega_k, m' \in \mathcal{M}_k$ due to Proposition E.9 and Lemma E.12, and the last inequality is by $|\mathcal{M}_k| \geq 1$ in Lemma D.4 and $\sum_{i \in \Omega_k} \beta_i = \Omega(n/K)$ in Lemma D.1.

Then we consider the case that $m, m' \notin \mathcal{M}_k$. Applying Taylor expansion of $\ell'_{i,t} = 1/2 + O(J\eta^3 t^3)$ gives

$$\begin{aligned}
&\frac{1}{n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \ell'_{i,t} \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \beta_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \\
&= \frac{1}{2n} \sum_{i \in \Omega_k} \mathbb{P}(m_{i,t} = m) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \beta_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) + O\big(J^2 \eta^6 t^6\big) \\
&= \frac{1}{2n} \sum_{k'} \sum_{i \in \Omega_{k,k'}^+} \mathbb{P}(m_{i,t} = m) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \beta_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) + O\big(J^2 \eta^6 t^6\big) \\
&\quad + \frac{1}{2n} \sum_{k'} \sum_{i \in \Omega_{k,k'}^-} \mathbb{P}(m_{i,t} = m) \pi_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) y_i \beta_i f_m(\mathbf{x}_i; \mathbf{W}^{(t)}) \\
&= O(J^2 \eta^6 t^6) + \widetilde{O}(d^{-0.005}). \qquad \text{(E.21)}
\end{aligned}$$

where the last inequality is by the technique we have used before in Lemma E.16. By (E.21), we can get upper bound $|I_1|, |I_3|$ by $O(J^2 \eta^6 t^6) + \widetilde{O}(d^{-0.005})$.

Plugging the bound of $I_1, I_2, I_3$ into (E.20) gives,

$$\begin{aligned}\left\langle \nabla_{\boldsymbol{\theta}_m} \mathcal{L}^{(t)}, \mathbf{c}_k \right\rangle &\geq \Omega\bigg(\frac{\eta^3 t^3}{KM^3}\bigg) + O(J^2\eta^6 t^6) + \widetilde{O}(d^{-0.005}) \\ &\leq \Omega\bigg(\frac{\eta^3 t^3}{KM^3}\bigg) + \widetilde{O}(d^{-0.005}),\end{aligned}$$

where the last inequality is by $t \leq T_2 = \lfloor \eta^{-1} M^{-2} \rfloor$. □

Now we can claim that Proposition E.9 is true and we summarize the results as follow lemma.

**Lemma E.18.** For all $T_1 \leq t \leq T_2$, we have Proposition E.9 holds. Besides, we have that $\langle \boldsymbol{\theta}_m^{(T_2)}, \mathbf{c}_k \rangle \leq \max_{m' \in [M]} \langle \boldsymbol{\theta}_{m'}^{(T_2)}, \mathbf{c}_k \rangle - \Omega(K^{-1}M^{-9})$ for all $m \notin \mathcal{M}_k$..

*Proof.* We will first use induction to prove Proposition E.9. It is worth noting that proposition E.9 is true at the beginning of the second stage $t = T_1$. Suppose (E.8), (E.9), (E.10) hold for all $t \in [T_1, T] \subseteq [T_1, T_2 - 1]$, we next verify that they also hold for $t \in [T_1, T+1]$. Lemma E.13 shows that (E.8) holds for $t \in [T_1, T+1]$. Lemma E.14 further shows that (E.8) holds for $t \in [T_1, T+1]$. Therefore, we only need to verify whether (E.10) holds for $t \in [T_1, T+1]$. Therefore, for each pair $i \in \Omega_k$, $m \in \mathcal{M}_k$, we need to estimate the gap between expert $m$ and the expert with best performance $h_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \max_{m'} h_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})$. By Lemma E.17 and Lemma E.14, we can induce that $h_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)})$ is small therefore cannot be the largest one. Thus $h_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \max_{m'} h_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) = h_m(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) - \max_{m'} h_{m'}(\mathbf{x}_i; \boldsymbol{\Theta}^{(t)}) \leq \Delta_{\boldsymbol{\Theta}^{(t)}} \leq \widetilde{O}(d^{-0.001})$. Therefore, by Lemma C.3 we have (E.10) holds. Now we have verified that (E.10) also holds for $t \in [T_1, T+1]$, which completes the induction for Lemma E.9.

Finally, we carefully characterize the value of $\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{c}_k \rangle$, for $\eta_r \eta^{-1} = \Theta(M^2)$ and $m \notin \mathcal{M}_k$. If $\langle \boldsymbol{\theta}_m^{(t)}, \mathbf{c}_k \rangle \geq \max_{m'} \langle \boldsymbol{\theta}_{m'}^{(t)}, \mathbf{c}_k \rangle - 1$, by Lemma E.17 we have that

$$\langle \boldsymbol{\theta}_m^{(t+1)}, c_k \rangle \leq \langle \boldsymbol{\theta}_m^{(t)}, c_k \rangle - \Theta\bigg(\frac{\eta_r \eta^3 t^3}{KM^3}\bigg) + \widetilde{O}(\eta_r d^{-0.005}) \leq 0. \tag{E.22}$$

If there exists $t \leq T_2 - 1$ such that $\langle \boldsymbol{\theta}_m^{(t+1)}, c_k \rangle \leq \max_{m'} \langle \boldsymbol{\theta}_{m'}^{(t)}, \mathbf{c}_k \rangle - 1$, clearly we have that $\langle \boldsymbol{\theta}_m^{(T_2)}, c_k \rangle \leq -\Omega(K^{-1}M^{-9})$ since $\langle \boldsymbol{\theta}_m^{(t)}, c_k \rangle$ will keep decreasing as long as $\langle \boldsymbol{\theta}_m^{(t+1)}, c_k \rangle \geq -1$ and our step size $\eta_r = \Theta(M^2)\eta$ is small enough. If $\langle \boldsymbol{\theta}_m^{(t+1)}, c_k \rangle \geq \max_{m'} \langle \boldsymbol{\theta}_{m'}^{(t)}, \mathbf{c}_k \rangle - 1$ holds for all $t \leq T_2 - 1$, take telescope sum of (E.22) from $t = 0$ to $t = T_2 - 1$ gives that

$$\begin{aligned}\langle \boldsymbol{\theta}_m^{(T_2)}, c_k \rangle &\leq \langle \boldsymbol{\theta}_m^{(0)}, c_k \rangle - \sum_{s=0}^{T_2-1} \Theta\bigg(\frac{\eta_r \eta^3 s^3}{KM^3}\bigg) + \widetilde{O}(d^{-0.005}) \\ &\overset{(i)}{=} - \sum_{s=0}^{T_2-1} \Theta\bigg(\frac{\eta_r \eta^3 s^3}{KM^3}\bigg) + \widetilde{O}(d^{-0.005}) \\ &\overset{(ii)}{=} -\Theta\bigg(\frac{\eta_r \eta^3 T_2^4}{KM^3}\bigg) + \widetilde{O}(d^{-0.005}) \\ &\leq -\Omega(K^{-1}M^{-9}),\end{aligned}$$

where the (i) is by $\boldsymbol{\theta}_m^{(0)} = 0$ and (ii) is by $\sum_{i=0}^{n-1} i^3 = n^2(n-1)^2/4$ and the last inequality is due to $T_2 = \lfloor \eta^{-1} M^{-2} \rfloor$ and $\eta_r = \Theta(M^2)\eta$. Now we have proved that $\langle \boldsymbol{\theta}_m^{(T_2)}, c_k \rangle \leq -\Omega(K^{-1}M^{-9})$ for all $m \notin \mathcal{M}_k$. Finally, by Lemma E.1 we have that

$$\max_{m' \in [M]} \langle \boldsymbol{\theta}_{m'}^{(T_2)}, \mathbf{c}_k \rangle \geq \frac{1}{m} \sum_{m' \in [M]} \langle \boldsymbol{\theta}_{m'}^{(T_2)}, \mathbf{c}_k \rangle = 0.$$

Therefore, we have that $\langle \boldsymbol{\theta}_m^{(T_2)}, \mathbf{c}_k \rangle \leq -\Omega(K^{-1}M^{-9}) \leq \max_{m' \in [M]} \langle \boldsymbol{\theta}_{m'}^{(T_2)}, \mathbf{c}_k \rangle - \Omega(K^{-1}M^{-9})$, which completes the proof.

□

### E.3 Generalization Results

In this section, we will present the detailed proof of Lemma 5.2 and Theorem 4.2 based on analysis in the previous stages.

*Proof of Lemma 5.2.* We consider the $m$-th expert in the MoE layer, suppose that $m \in \mathcal{M}_k$. Then if we draw a new sample $(\mathbf{x}, y) \in \Omega_k$. Without loss of generality, we assume $\mathbf{x} = [\alpha y \mathbf{v}_k, \beta \mathbf{c}_k, \gamma \epsilon \mathbf{v}_{k'}, \boldsymbol{\xi}]$. By Lemma E.8, we have already get the bound for inner product between weights and feature signal, cluster-center signal and feature noise. However, we need to recalculate the bound of the inner product between weights and random noises because we have fresh random noises i.i.d drawn from $\mathcal{N}(0, (\sigma_p^2/d) \cdot I_d)$. Notice that we use normalized gradient descent for expert with step size $\eta$, so we have that

$$\|\mathbf{w}_{m,j}^{(T_1)} - \mathbf{w}_{m,j}^{(0)}\|_2 \leq \eta T_1 = O(\sigma_0^{0.5}).$$

Therefore, by triangle inequality we have that $\|\mathbf{w}_{m,j}^{(T_1)}\|_2 \leq \|\mathbf{w}_{m,j}^{(0)}\|_2 + O(\sigma_0^{0.5}) \leq \widetilde{O}(\sigma_0\sqrt{d})$. Because the inner product $\langle \mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_p \rangle$ follows the distribution $\mathcal{N}(0, (\sigma_p^2/d) \cdot \|\mathbf{w}_{m,j}^{(T_1)}\|_2^2)$, we have that with probability at least $1 - 1/(dPMJ)$,

$$|\langle \mathbf{w}_{m,j}^{(T_1)}, \boldsymbol{\xi}_p \rangle| = O(\sigma_p d^{-1/2} \|\mathbf{w}_{m,j}^{(t)}\|_2 \log(dPMJ)) \leq \widetilde{O}(\sigma_0).$$

Applying Union bound for $m \in [M], j \in [J], p \geq 4$ gives that, with probability at least $1 - 1/d$,

$$|\langle \mathbf{w}_{m,j}^{(T_1)}, \boldsymbol{\xi}_p \rangle| = \widetilde{O}(\sigma_0), \forall m \in [M], j \in [J], p \geq 4. \tag{E.23}$$

Now under the event that (E.23) holds, we have that

$$\begin{aligned}
y f_m(\mathbf{x}, \mathbf{W}^{(t)}) &= y \sum_{j \in [J]} \sum_{p \in [P]} \sigma(\langle \mathbf{w}_{m,j}, \mathbf{x}^{(p)} \rangle) \\
&= y\sigma(\langle \mathbf{w}_{m,j_m^*}, \alpha y \mathbf{v}_k \rangle) + y \sum_{(j,p) \neq (j_m^*, 1)} \sigma(\langle \mathbf{w}_{m,j}, \mathbf{x}^{(p)} \rangle) \\
&\geq C_1^3 (1 - \sigma_0^{0.1})^3 \sigma_0^{1.5} - \widetilde{O}(\sigma_0^3) \\
&\geq \Omega(\sigma_0^{1.5}),
\end{aligned}$$

where the first inequality is due to (E.3). Because (E.23) holds holds with probability at least $1 - 1/d$, so we have prove that

$$\mathbb{P}_{(\mathbf{x},y) \sim \mathcal{D}}\big(y f_m(\mathbf{x}; \mathbf{W}^{(T_1)}) \leq 0 \big| (\mathbf{x}, y) \in \Omega_k\big) \leq 1/d.$$

On the other hand, if we draw a new sample $(\mathbf{x}, y) \in \Omega_{k'}, k' \neq k$. Then we consider the special set $\Omega_{k',k}^- \subseteq \Omega_{k'}$ where feature noise is $\mathbf{v}_k$ and the sign of the feature noise $\epsilon$ is not equal to the label $y$. Without loss of generality, we assume it as $\mathbf{x} = [\alpha y \mathbf{v}_{k'}, \beta \mathbf{c}_{k'}, -\gamma y \mathbf{v}_k, \boldsymbol{\xi}]$. Then under the event that (E.23) holds, we have that

$$\begin{aligned}
y f_m(\mathbf{x}, \mathbf{W}^{(t)}) &= y \sum_{j \in [J]} \sum_{p \in [P]} \sigma(\langle \mathbf{w}_{m,j}, \mathbf{x}^{(p)} \rangle) \\
&= y\sigma(\langle \mathbf{w}_{m,j_m^*}, -\gamma y \mathbf{v}_k \rangle) + y \sum_{(j,p) \neq (j_m^*, 3)} \sigma(\langle \mathbf{w}_{m,j}, \mathbf{x}^{(p)} \rangle) \\
&\leq -C_1^3 (1 - \sigma_0^{0.1})^3 \sigma_0^{1.5} + \widetilde{O}(\sigma_0^3) \\
&\leq -\Omega(\sigma_0^{1.5}),
\end{aligned}$$

where the first inequality is due to (E.3). Because (E.23) holds holds with probability at least $1 - 1/d$, so we have prove that

$$\mathbb{P}_{(\mathbf{x},y) \sim \mathcal{D}}\big(y f_m(\mathbf{x}; \mathbf{W}^{(T_1)}) \leq 0 \big| (\mathbf{x}, y) \in \Omega_{k',k}^-\big) \geq 1 - 1/d.$$

Then we further have that

$$\begin{aligned}
&\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T_1)})\leq 0\big|(\mathbf{x},y)\in\Omega_{k'}\big)\\
&\geq \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T_1)})\leq 0\big|(\mathbf{x},y)\in\Omega^-_{k',k}\big)\cdot\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big((\mathbf{x},y)\in\Omega^-_{k',k}\big|(\mathbf{x},y)\in\Omega_{k'}\big)\\
&\geq \Omega(1/K),
\end{aligned}$$

which completes the proof.

□

*Proof of Theorem 4.2.* We will give the prove for $T = T_2$, i.e., at the end of the second stage.

**Test Error is small.** We first prove the following result for the experts. For all expert $m \in \mathcal{M}_k$, we have that

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T)})\leq 0\big|(\mathbf{x},y)\in\Omega_k\big) = o(1). \tag{E.24}$$

The proof of is similar to the proof of Lemma 5.2. We consider the $m$-th expert in the MoE layer, suppose that $m \in \mathcal{M}_k$. Then if we draw a new sample $(\mathbf{x}, y) \in \Omega_k$. Without loss of generality, we assume $\mathbf{x} = [\alpha y\mathbf{v}_k, \beta\mathbf{c}_k, \gamma\epsilon\mathbf{v}_{k'}, \boldsymbol{\xi}]$. By Lemma E.8, we have already get the bound for inner product between weights and feature signal, cluster-center signal and feature noise. However, we need to recalculate the bound of the inner product between weights and random noises because we have fresh random noises i.i.d drawn from $\mathcal{N}(0, (\sigma_p^2/d)\cdot I_d)$. Notice that we use normalized gradient descent with step size $\eta$, so we have that

$$\|\mathbf{w}_{m,j}^{(T)} - \mathbf{w}_{m,j}^{(0)}\|_2 \leq \eta T = \widetilde{O}(1).$$

Therefore, by triangle inequality we have that $\|\mathbf{w}_{m,j}^{(T)}\|_2 \leq \|\mathbf{w}_{m,j}^{(0)}\|_2 + \widetilde{O}(1) \leq \widetilde{O}(\sigma_0\sqrt{d})$. Because the inner product $\langle\mathbf{w}_{m,j}^{(t)}, \boldsymbol{\xi}_p\rangle$ follows the distribution $\mathcal{N}(0, (\sigma_p^2/d)\cdot\|\mathbf{w}_{m,j}^{(T)}\|_2^2)$, with probability at least $1 - 1/(dPMJ)$ we have that ,

$$|\langle\mathbf{w}_{m,j}^{(T)}, \boldsymbol{\xi}_p\rangle| = O(\sigma_p d^{-1/2}\|\mathbf{w}_{m,j}^{(t)}\|_2\log(dPMJ)) \leq \widetilde{O}(\sigma_0).$$

Applying Union bound for $m \in [M], j \in [J], p \geq 4$ gives that, with probability at least $1 - 1/d$,

$$|\langle\mathbf{w}_{m,j}^{(T)}, \boldsymbol{\xi}_p\rangle| = \widetilde{O}(\sigma_0), \forall m \in [M], j \in [J], p \geq 4. \tag{E.25}$$

Now, under the event that (E.25) holds, we have that

$$\begin{aligned}
yf_m(\mathbf{x},\mathbf{W}^{(T)}) &= y\sum_{j\in[J]}\sum_{p\in[P]}\sigma(\langle\mathbf{w}_{m,j}^{(T)}, \mathbf{x}^{(p)}\rangle)\\
&= y\sigma(\langle\mathbf{w}_{m,j_m^*}^{(T)}, \alpha y\mathbf{v}_k\rangle) + y\sum_{(j,p)\neq(j_m^*,1)}\sigma(\langle\mathbf{w}_{m,j}^{(T)}, \mathbf{x}^{(p)}\rangle)\\
&\geq C_1^3(1-\sigma_0^{0.1})^3M^{-4} - \widetilde{O}(\sigma_0^3)\\
&= \widetilde{\Omega}(1),
\end{aligned}$$

where the first inequality is by Lemma E.12. Because (E.25) holds with probability at least $1 - 1/d$, so we have prove that

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big(yf_m(\mathbf{x};\mathbf{W}^{(T)})\leq 0\big|(\mathbf{x},y)\in\Omega_k\big) \leq 1/d.$$

We then prove that, with probability at least $1 - o(1)$, an example $\mathbf{x} \in \Omega_k$ will be routed to one of the experts in $\mathcal{M}_k$. For $\mathbf{x} = [\alpha y\mathbf{v}_k, \beta\mathbf{c}_k, \gamma\epsilon\mathbf{v}_{k'}, \boldsymbol{\xi}]$, we need to check that $h_m(\mathbf{x};\boldsymbol{\Theta}^{(T)}) < \max_{m'} h_{m'}(\mathbf{x};\boldsymbol{\Theta}^{(T)}), \forall m \notin \mathcal{M}_k$. By Lemma E.18, we know that $\langle\boldsymbol{\theta}_m^{(T)}, \mathbf{c}_k\rangle \leq \max_{m'}\langle\boldsymbol{\theta}_{m'}^{(T)}, \mathbf{c}_k\rangle -$ $-\Omega(K^{-1}M^{-9})$. Further by Lemma E.14, we have that $\max_{m,k}|\langle\boldsymbol{\theta}_m^{(T)}, \mathbf{v}_k\rangle| = O(d^{-0.001})$. Again to calculate test error, we need to give an upper bound $\langle\boldsymbol{\theta}_m^{(T)}, \boldsymbol{\xi}_p\rangle$, where $\boldsymbol{\xi}_p$ is a fresh noise drawn from $\mathcal{N}(0, (\sigma_p^2/d)\cdot I_d)$. We can upper bound the gradient of the gating network by

$$\|\nabla_{\boldsymbol{\theta}_m}\mathcal{L}^{(t)}\|_2 = \bigg\|\frac{1}{n}\sum_{i,p}\mathbb{1}(m_{i,t}=m)\ell'_{i,t}\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})y_i f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\mathbf{x}_i^{(p)} - \frac{1}{n}\sum_{i,p}\ell'_{i,t}\pi_{m_{i,t}}(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})\pi_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(t)})y_i f_{m_{i,t}}(\mathbf{x}_i;\mathbf{W}^{(t)})\mathbf{x}_i^{(p)}\bigg\|_2.$$
$$= \widetilde{O}(1),$$

where the last inequality is due to $|\ell'_{i,t}| \leq 1$, $\pi_m, \pi_{m_{i,t}} \in [0,1]$ and $\|\mathbf{x}_i^{(p)}\|_2 = O(1)$. This further implies that

$$\|\boldsymbol{\theta}_m^{(T)}\|_2 = \|\boldsymbol{\theta}_m^{(T)} - \boldsymbol{\theta}_m^{(0)}\|_2 \leq \widetilde{O}(t\eta_r) \leq \widetilde{O}(\eta^{-1}\eta_r) = \widetilde{O}(1),$$

where the last inequality is by $\eta_r = \Theta(M^2)\eta$. Because the inner product $\langle\boldsymbol{\theta}_m^{(T)}, \boldsymbol{\xi}_p\rangle$ follows the distribution $\mathcal{N}(0, (\sigma_p^2/d)\cdot\|\boldsymbol{\theta}_m^{(T)}\|_2^2)$, we have that with probability at least $1 - 1/(dPM)$,

$$|\langle\boldsymbol{\theta}_m^{(T)}, \boldsymbol{\xi}_p\rangle| = O(\sigma_p d^{-1/2}\|\boldsymbol{\theta}_m^{(T)}\|_2 \log(dPM)) \leq \widetilde{O}(d^{-1/2}).$$

Applying Union bound for $m \in [M], p \geq 4$ gives that, with probability at least $1 - 1/d$,

$$|\langle\boldsymbol{\theta}_m^{(T)}, \boldsymbol{\xi}_p\rangle| = \widetilde{O}(d^{-1/2}), \forall m \in [M], p \geq 4. \tag{E.26}$$

Now, under the event that (E.26) holds, we have that

$$\begin{aligned}
&h_m(\mathbf{x};\boldsymbol{\Theta}^{(T)}) - \max_{m'} h_{m'}(\mathbf{x};\boldsymbol{\Theta}^{(T)}) \\
&\leq \langle\boldsymbol{\theta}_m^{(T)}, \mathbf{c}_k\rangle - \max_{m'}\langle\boldsymbol{\theta}_{m'}^{(T)}, \mathbf{c}_k\rangle + 4\max_{m,k}|\langle\boldsymbol{\theta}_m^{(T)}, \mathbf{v}_k\rangle| + 4P\max_{m,p}|\langle\boldsymbol{\theta}_m^{(T)}, \boldsymbol{\xi}_p\rangle| \\
&\leq -\Omega(K^{-1}M^{-9}) + \widetilde{O}(d^{-0.001}) \\
&< 0.
\end{aligned}$$

Because (E.26) holds holds with probability at least $1 - 1/d$, so we have prove that with probability at least $1 - 1/d$, an example $\mathbf{x} \in \Omega_k$ will be routed to one of the experts in $\mathcal{M}_k$.

**Training Error is zero.** The prove for training error is much easier, because we no longer need to deal with the fresh noises and we no longer need to use high probability bound for those inner products with fresh noises. That's the reason we can get exactly zero training error. We first prove the following result for the experts. For all expert $m \in \mathcal{M}_k$, we have that

$$y_i f_m(\mathbf{x}_i;\mathbf{W}^{(T)}) \leq 0, \forall i \in \Omega_k.$$

Without loss of generality, we assume that the feature patch appears in $\mathbf{x}_i^{(1)}$. By Lemma E.12, we have that for all $i \in \Omega_k$

$$\begin{aligned}
y_i f_m(\mathbf{x}_i, \mathbf{W}^{(T)}) &= y_i \sum_{j\in[J]}\sum_{p\in[P]}\sigma(\langle\mathbf{w}_{m,j}^{(T)}, \mathbf{x}_i^{(p)}\rangle) \\
&= y_i\sigma(\langle\mathbf{w}_{m,j_m^*}^{(T)}, \alpha y_i\mathbf{v}_k\rangle) + y_i\sum_{(j,p)\neq(j_m^*,1)}\sigma(\langle\mathbf{w}_{m,j}^{(T)}, \mathbf{x}^{(p)}\rangle) \\
&\geq C_1^3(1-\sigma_0^{0.1})^3 M^{-4} - \widetilde{O}(\sigma_0^3) \\
&> 0,
\end{aligned}$$

where the first inequality is Lemma E.12. We then prove that, and example $(\mathbf{x}_i, y_i) \in \Omega$ will be routed to one of the experts in $\mathcal{M}_k$. Suppose the $m$-th expert is not in $\mathcal{M}_k$. We only need to check the value of $h_m(\mathbf{x}_i;\boldsymbol{\Theta}^{(T)}) < \max_{m'} h_{m'}(\mathbf{x}_i;\boldsymbol{\Theta}^{(T)})$, which is straight forward by Lemma E.18 and Lemma E.14.

□

## F Auxiliary Lemmas

**Lemma F.1.** Let $\{a_m\}_{m=1}^M$ are the random variable i.i.d. drawn from $\mathcal{N}(0,1)$. Define the non-increasing sequence of $\{a_m\}_{m=1}^M$ as $a^{(1)} \geq \ldots \geq a^{(M)}$. Then we have that

$$\mathbb{P}(a^{(2)} \geq (1-G)a^{(1)}) \leq GM^2$$

*Proof.* Let $\Psi$ be the CDF of $\mathcal{N}(0,1)$ and let $\rho$ be the PDF of $\mathcal{N}(0,\sigma_0^2)$. Then we have that,

$$\begin{aligned}
&\mathbb{P}(a^{(2)} \geq (1-G)a^{(1)}) \\
&= \int_{a^{(1)} \geq \ldots \geq a^{(M)}} \mathbb{1}(a^{(2)} \geq (1-G)a^{(1)}) M! \Pi_m \rho(a^{(m)}) d\mathbf{a} \\
&= \int_{a^{(1)} \geq a^{(2)}} \mathbb{1}(a^{(2)} \geq (1-G)a^{(1)}) M(M-1)\rho(a^{(1)})\rho(a^{(2)})\Psi(a^{(2)})^{M-2} da^{(1)} da^{(2)} \\
&\leq \int_{a^{(1)} \geq a^{(2)}} \mathbb{1}(a^{(2)} \geq (1-G)a^{(1)}) M(M-1)\rho(a^{(1)}) \frac{1}{\sqrt{2\pi}} da^{(1)} da^{(2)} \\
&= \int_{a^{(1)} \geq 0} \frac{GM(M-1)}{\sqrt{2\pi}} a^{(1)} \rho(a^{(1)}) da^{(1)} \\
&\leq GM^2.
\end{aligned}$$

□

For normalized gradient descent we have following lemma,

**Lemma F.2** (Lemma C.19 Allen-Zhu and Li 2020c)**.** Let $\{x_t, y_t\}_{t=1,..}$ be two positive sequences that satisfy

$$\begin{aligned}
x_{t+1} &\geq x_t + \eta \cdot C_t x_t^2 \\
y_{t+1} &\leq y_t + S\eta \cdot C_t y_t^2,
\end{aligned}$$

and $|x_{t+1} - x_t|^2 + |y_{t+1} - y_t|^2 \leq \eta^2$. Suppose $x_0, y_0 = o(1), x_0 \geq y_0 S(1+G)$,

$$\eta \leq \min\{\frac{G^2 x_0}{\log(A/x_0)}, \frac{G^2 y_0}{\log(1/G)}\}.$$

Then we have for all $A > x_0$, let $T_x$ be the first iteration such that $x_t \geq A$, then we have $y_{T_x} \leq O(y_0 G^{-1})$.

*Proof.* We only need to replace $O(\eta A^{q-1})$ in the proof of Lemma C.19 by $O(\eta)$, because we use normalized gradient descent, i.e, $C_t \mathbf{x}_t^2 \leq 1$. For completeness, we present the whole poof here.

for all $g = 0, 1, 2, \ldots,$ let $\mathcal{T}_g$ be the first iteration such that $x_t \geq (1+\delta)^g x_0$, let $b$ be the smallest integer such that $(1+\delta)^b x_0 \geq A$. For simplicity of notation, we replace $x_t$ with $A$ whenever $x_t \geq A$. Then by the definition of $\mathcal{T}_g$, we have that

$$\sum_{t \in [\mathcal{T}_g, \mathcal{T}_{g+1})} \eta C_t [(1+\delta)^g x_0]^2 \leq x_{\mathcal{T}_{g+1}} - x_{\mathcal{T}_g} \leq \delta(1+\delta)^g x_0 + O(\eta),$$

where the last inequality holds because we are using normalized gradient descent, i.e., $\max_t |x_{t+1} - x_t| \leq \eta$. This implies that

$$\sum_{t \in [\mathcal{T}_g, \mathcal{T}_{g+1})} \eta C_t \leq \frac{\delta}{(1+\delta)^g} \frac{1}{x_0} + \frac{O(\eta)}{x_0^2}.$$

Recall that $b$ is the smallest integer such that $(1+\delta)^b x_0 \geq A$, so we can calculate

$$\sum_{t \geq 0, x_t \leq A} \eta C_t \leq \left[\sum_{g=0}^{b-1} \frac{\delta}{(1+\delta)^g} \frac{1}{x_0}\right] + \frac{O(\eta)}{x_0^2} b = \frac{1+\delta}{x_0} + \frac{O(\eta) b}{x_0^2} \leq \frac{1+\delta}{x_0} + \frac{O(\eta)\log(A/x_0)}{x_0^2 \log(1+\delta)}$$

Let $T_x$ be the first iteration $t$ in which $x_t \geq A$. Then we have that

$$\sum_{t=0}^{T_x} \eta C_t \leq \frac{1+\delta}{x_0} + \frac{O(\eta)\log(A/x_0)}{\delta x_0^2}. \tag{F.1}$$

On the other hand, let $A' = G^{-1}y_0$ and b' be the smallest integer such that $(1+\delta)b'x_0 \geq A'$. For simplicity of notation, we replace $y_t$ with $A'$ when $y_t \geq A'$. Then let $\mathcal{T}'_g$ be the first iteration such that $y_t \geq (1+\delta)^g y_0$, then we have that

$$\sum_{t\in[\mathcal{T}'_g,\mathcal{T}'_{g+1})} \eta S C_t [(1+\delta)^{g+1}x_0]^{(q-1)} \geq y_{\mathcal{T}'_{g+1}} - y_{\mathcal{T}'_g} \geq \delta(1+\delta)^g y_0 - O(\eta).$$

Therefore, we have that

$$\sum_{t\in[\mathcal{T}'_g,\mathcal{T}'_{g+1})} S\eta C_t \geq \frac{\delta}{(1+\delta)^g(1+\delta)^2}\frac{1}{y_0} - \frac{O(\eta)}{y_0^2}.$$

Recall that $b'$ is the smallest integer such that $(1+\delta)^{b'}y_0 \geq A'$. wo we have that

$$\sum_{t\geq 0, x_t \leq A} \eta S C_t \geq \sum_{g=0}^{b'-2} \frac{\delta}{(1+\delta)^g(1+\delta)^2}\frac{1}{y_0} - \frac{O(\eta)b'}{y_0^2}$$

Let $T_y$ be the first iteration $t$ in which $y_t \geq A'$, so we can calculate

$$\sum_{t=0}^{T_y} \eta S C_t \geq \frac{1-O(\delta+G)}{y_0} - \frac{O(\eta)\log(A'/y_0)}{y_0^2\delta}. \tag{F.2}$$

Compare (F.1) and (F.2). Choosing $\delta = G$ and $\eta \leq \min\{\frac{G^2x_0}{\log(A/x_0)}, \frac{G^2y_0}{\log(1/G)}\}$, together with $x_0 \geq y_0 S(1+G)$

□