MoDE: A Mixture-of-Experts Model with Mutual Distillation among the Experts

Zhitian Xie[1*], Yinger Zhang[2*†], Chenyi Zhuang[1‡], Qitao Shi[1], Zhining Liu[1], Jinjie Gu[1], Guannan Zhang[1]

[1]Ant Group
[2]Zhejiang University
{xiezhitian.xzt, chenyi.zcy, qitao.sqt, eason.lzn, jinjie.gujj, zgn138592}@antgroup.com, zhangyinger@zju.edu.cn

Abstract

The application of mixture-of-experts (MoE) is gaining popularity due to its ability to improve model's performance. In an MoE structure, the gate layer plays a significant role in distinguishing and routing input features to different experts. This enables each expert to specialize in processing their corresponding sub-tasks. However, the gate's routing mechanism also gives rise to narrow vision: the individual MoE's expert fails to use more samples in learning the allocated sub-task, which in turn limits the MoE to further improve its generalization ability. To effectively address this, we propose a method called Mixture-of-Distilled-Expert (MoDE), which applies moderate mutual distillation among experts to enable each expert to pick up more features learned by other experts and gain more accurate perceptions on their original allocated sub-tasks. We conduct plenty experiments including tabular, NLP and CV datasets, which shows MoDE's effectiveness, universality and robustness. Furthermore, we develop a parallel study through innovatively constructing "expert probing", to experimentally prove why MoDE works: moderate distilling knowledge can improve each individual expert's test performances on their assigned tasks, leading to MoE's overall performance improvement.

Introduction

Datasets can be naturally divided into different subsets (such as those from different subdomains or with distinct sub-tasks) and attempting to learn these datasets with a single model may meet difficulties in fitting and generalization (Jacobs et al. 1991; Eigen, Ranzato, and Sutskever 2013; Shazeer et al. 2017). To address this, the Mixture of Experts (MoE) system has been proposed, which consists of several different experts and a gating network as the router. MoE have been applied in various domains, including multi-task learning (Ma et al. 2018), NLP (Shazeer et al. 2017; Lepikhin et al. 2020), and CV (Svetlana et al. 2022; Pham et al. 2021; Dosovitskiy et al. 2020; Riquelme et al. 2021) and proven as a promising architecture. Many studies (Jacobs et al. 1991; Eigen, Ranzato, and Sutskever 2013; Shazeer
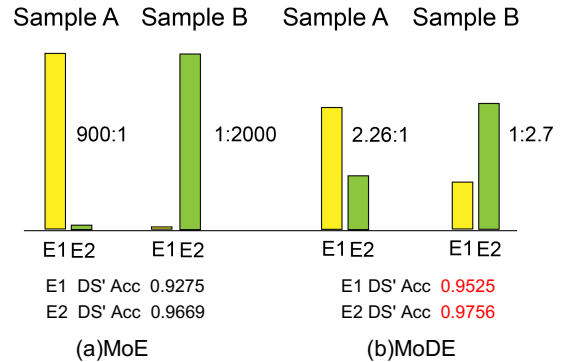

Figure 1: Narrow Vision. A and B represent two subsets of the training samples, while E1/2 represent expert 1/2. The normalized histogram schematically presents the ratio of gradients on each expert in learning the same subset. The narrow vision in MoE is indeed significant and MoDE alleviates it through distillation. As a result, it improves the accuracy of individual experts on their dominating sample-based task domain (referred as *DS* in the following section).

et al. 2017) have shown that in the MoE structure, each expert is specialized in processing a certain subset of samples.

The experts' specialization comes from the fact that they merely learn the limited sample features assigned by the gate during the training process. Figure 1(a) shows that during the training process of an MoE architecture (with totally two experts), different subsets of the training samples partitioned by the gate contribute their learning gradients in significantly different levels: Sample A has a significant gradient to update expert 1's parameters, but has negligible impact on expert 2. Therefore, certain samples contribute to each expert's specialization. However, is the mechanism of allocating limited samples to each expert the best way to construct MoE? Assigning limited samples to each expert in an MoE can lead to a "narrow vision" issue, where experts are not exposed to enough diverse data to develop a comprehensive understanding of their respective sub-tasks, potentially impairing the model's generalization performance.

Can we appropriately address narrow vision in MoE, while still maintaining each expert's specialization to improve the MoE's overall generalization ability? To answer

---

*These authors contributed equally.

†This author conducted this work when she was serving as an intern in Ant Group.

‡This author is the corresponding author.

this question, we propose a training methodology called Mixture-of-Distilled-Experts (MoDE), which involves a moderate level of mutual distillation among experts. For each expert, the knowledge obtained from other experts, which is derived from their allocated samples, may provide potentially useful features that are not present in the limited samples allocated to that particular expert. Consequently, this exchange of knowledge can enhance the expert's perception of its sub-task and the MoE's overall performance. Figure 1 illustrates the mitigation of narrow vision in MoDE, as evidenced by the increased accuracy of the expert's allocated sub-task (referred as *DS* in the following section). Moreover, plenty of experiments and analysis have been conducted to illustrate the effectiveness and mechanism of MoDE. Innovatively, we conduct a novel parallel study utilizing "expert probing" as an evaluation method to approximate the performance of individual experts within the MoE. We observe that excessive knowledge distillation pushes the experts to present overly similar opinions, undermining the experts' specialization, which deviates from the original motivation of introducing the MoE structure: encourage each expert to be specialized and do what they are good at, resulting in the failure to improve MoE's generalization ability. However, by employing an appropriate distillation strength, each individual expert does not only maintain its specialization but also achieves improved test performance in its allocated sub-task, consequently enhancing the MoE model's generalization ability. With plenty of experiments on the datasets of tabular, NLP and CV, MoDE proves its effectiveness, universality and robustness in solving narrow vision and provide us a valuable exploring space to increase MoE's generalization ability.

Related Work

In our work, we apply mutual distillation to the MoE to improve its generalization ability. The most recent related works are introduced in the following subsections.

Mixture-of-Experts

MoE was first introduced by Jacob et al. (Jacobs et al. 1991) to combine multiple experts, each trained on a different subset of the data, to form a single powerful model. Eigen et al. (Eigen, Ranzato, and Sutskever 2013) extends the MoE to a layer in neural network, which consists of a set of experts (neural networks) and a trainable gate. The gate assigns weights to the experts on a per-example basis, which enables the MoE to output a weighted combination of the experts' outputs. As for the gate routing mechanism, dense gate MoE (DMoE) was firstly introduced, which assigns continuous weight to employ all the experts for each input (Eigen, Ranzato, and Sutskever 2013; Ma et al. 2018; Jacobs et al. 1991; Jordan and Jacobs 1994; Chen, Xu, and Chi 1999; Yuksel, Wilson, and Gader 2012). Recently, the sparse gate MoE (SMoE) have been proposed, to reduce the tremendous computational cost because of the enormous parameters, through activating partial experts or subsets of a network for each input (Shazeer et al. 2017; Zuo et al. 2021; Lewis et al. 2021; Fedus, Zoph, and Shazeer 2022).
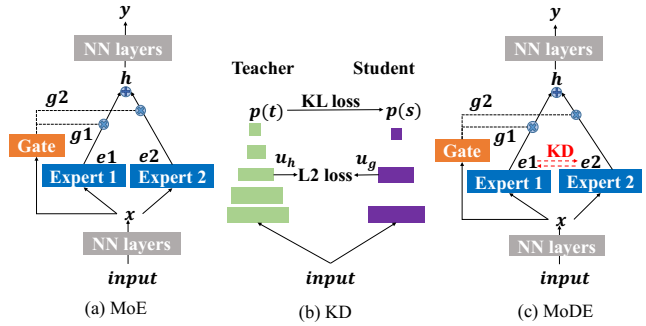

Figure 2: Overview of MoDE.

Although there are many studies on MoE's gate routing strategy, few works on exploring and utilizing the interactions among individual experts, espeicaly narrow vision, have been conducted. In this study, we discover that the mutual distillation among the experts can facilitate the communication of the knowledge acquired in their distinctive feature learning process, and this exchange yields positive effects on the MoE's overall performance.

Knowledge Distillation

Knowledge Distillation (KD) is originally proposed by Hinton et al. (Hinton, Vinyals, and Dean 2015) to transfer the knowledge from a high-capacity teacher model to a compact student model, aiming at a more efficient deployment of machine learning models on resource-constrained devices. In (Hinton, Vinyals, and Dean 2015), a small model is trained to directly match the output predictions of the teacher model, while in recent years, many methods have been proposed to excavate more information from the teacher, such as intermediate representations, additional attention information (Zagoruyko and Komodakis 2016), relations of layers (Yim et al. 2017) and mutual relations of data examples (Park et al. 2019). These methods have broadened the scope of knowledge distillation beyond just directly learning the teacher model's output. Recent advancements in online KD (Guo et al. 2020; Zhu, Gong et al. 2018; Kim et al. 2021; Wu and Gong 2021) have enabled updates of the teacher and student models simultaneously, significantly simplifying the training process. Deep mutual learning (Zhang et al. 2018) allows peer students to learn from each other by computing cross-entropy loss between each pair of students. Zhu et al. (Allen-Zhu and Li 2020) investigate the mechanism of ensemble and knowledge distillation. Their work theoretically proves that "multi-view" data structure leads to ensembles provably improving the test performance through learning most of the features.

Preliminary

Mixture-of-Experts

Figure 2 (a) illustrates a general MoE layer, which can be inserted into any neural network. The MoE layer accepts the output vector x of the previous layer as input, and outputs a

vector h to a successive layer. y is the output of the whole model. The output of MoE layer can be formulated as:

$$h = \sum_{i \in \psi} g_i(x) e_i(x), \qquad (1)$$

where $e_i(x)$ is the output of the ith expert belonging to the expert set ψ, $g_i(x)$ indicates the weight on the ith expert distributed by the gate. Then the gradient of h with respect to the ith expert's parameters can be written as below:

$$\frac{\alpha h}{\alpha \theta_i} = g_i(x) \frac{\alpha e_i(x)}{\alpha \theta_i}. \qquad (2)$$

It indicates that during the training process of each expert, the backpropagation not only depends on the gradient of this expert w.r.t its parameters, but also the gate's weight distribution: a negligible distributed gate weight negligibly updates this expert's parameter using gradient descent.

As introduced in the previous section, MoE can be divided into DMoE and SMoE. DMoE employs all the experts N for each input, and the gating network $g(x)$ simply comes from a linear transformation $f(.)$ using a softmax layer:

$$g(x) = Softmax(f(x)), |\psi| = N. \qquad (3)$$

For SMoE, only a part of experts K are selected by the routing strategy. Considering that the routing strategy is not our discussion's topic, so we choose the Topk routing mechanism in (Lepikhin et al. 2020; Fedus, Zoph, and Shazeer 2022; Riquelme et al. 2021) as the representative of SMoE, following (Riquelme et al. 2021), $g(x)$ can be written as:

$$g(x) = TopK(Softmax(f(x)) + \epsilon), |\psi| = K, K \ll N, \qquad (4)$$

where $\varepsilon \sim N(0, \frac{1}{E^2})$ is the Gaussian noise for exploration of expert routing. When $K \ll N$, most elements of $g(x)$ would be zero so that sparse conditional computation is achieved.

Various MoE structures applied in our experiments will be discussed later.

Knowledge Distillation

Figure 2 (b) illustrates two kinds of knowledge distillation methods. In traditional knowledge distillation, the knowledge of distillation comes from the logit output of the model:

$$p_i(z_i, T) = \frac{exp(z_i/T)}{\sum_{i=0}^{k} exp(z_i/T)}, \qquad (5)$$

$$L_{KD}(p(t,T), p(s,T)) = \sum_{i=0}^{k} -p_i(t_i, T)log(p_i(s_i, T)), \qquad (6)$$

Where T is the temperature coefficient used to control the softening degree of output logit. $p(t,T)$ and $p(s,T)$ represents the soft output logit of the teacher model and student model, respectively.

Moreover, some studies propose to utilize the features extracted from the teacher model's layer as a guide for the output of the student model's middle layer. We also adopted this approach as the distillation scheme. The L2 loss of the embeddings from the hidden layer is defined as:

$$L_{KD}(W_t, W_s) = \frac{1}{2} \|u_h(x; W_t) - u_g(x; W_s)\|^2. \qquad (7)$$

W_t is the weight of the first h layers of the teacher model, W_s is the weight of the first g layers of the student model. u is the output vector of the embedding.

Methodology

In our work, we propose a methodology called Mixture-of-Distilled-Expert (MoDE), which applies mutual distillation among MoE's experts to encourage each expert to learn more effective features learned by other experts, to gain more accurate perceptions on its learning samples, which in turn increases the overall MoE model's generalization ability. We start from an MoDE with 2 experts as a simple example, then we extend it to more experts and sparse gate.

Mixture-of-Distilled-Experts

Our method establishes a loss function to encourage the knowledge distillation among the experts, denoted as L_{KD}. Therefore, the overall loss function L is defined as:

$$L = L_{task} + \alpha L_{KD}, \qquad (8)$$

where L_{task} represents the loss function of the network itself, which is related to specific tasks, α is the distillation strength and will be discussed in the following section.

When designing the distillation loss L_{KD}, we have replaced the conventional teacher-student model with a collaborative learning approach, where each expert functions as a peer student learning from one another. Moreover, we expands the distillation approach from traditionally learning output predictions to intermediate representations. This provides a wider range of potential applications for the proposed MoDE layer, as it can now be easily integrated into any neural network as an independent layer.

When the expert number in the MoE is $K = 2$, the knowledge distillation loss L_{KD} is defined as the squared mean error between the experts' output e_1 and e_2. $mean(\cdot)$ operator represents the mean for all dimensions of the vector:

$$L_{KD} = mean((e_1 - e_2)^2). \qquad (9)$$

When the MoE layer is located at the end of the network, e_1 and e_2 correspond to the output predictions. While $K > 2$, to address the computational complexity, we apply the average of K experts as a single teacher e_{avg} to provide an averaged learning experience and each expert only distills from this single teacher:

$$L_{KD} = \frac{1}{K} \sum_{i=1}^{K} mean((e_i - e_{avg})^2), e_{avg} = \frac{1}{K} \sum_{i=1}^{K} e_i. \qquad (10)$$

Extension to Sparse-Gated MoE

For SMoE, only the activated K experts will be involved in the distillation process, and the method for calculating the distillation loss is the same as in the previous subsection. It

Dataset	N	#Dim	#Classes
Isolet	7797	617	26
Mfeat-karhunen	2000	64	10
Mfeat-factors	2000	216	10
Optdigits	5620	64	10
Satimage	6430	36	6
First-order-theorem-proving	6118	52	6
Artificial-characters	10218	8	10

Table 1: Statistics of tabular datasets.

Name	Baseline Architecture	Gate Type	Data Type
T-DMoE	DNN	Dense	Tabular Data
N-DMoE	Transformer	Dense	NLP Data
C-DMoE	CNN	Dense	CV Data
T-SMoE	DNN	Sparse	Tabular Data

Table 2: MoE's Architectures on Different Datasets.

is important to note that in DMoE, the outputs e_i and e_j from expert i and expert j include all samples, whereas in SMoE, a sample only passes through a subset of the experts, and only outputs from the same input can be mutually distilled.

Experiments

In this section, we answer the following questions through plenty experiments and discussions:

- *Does MoDE work universally?*
- *How and Why MoDE works?*
- *Is MoDE robust?*

Datasets

Tabular Datasets 7 tabular benchmark data sets of classification task from the OpenML[1] are used. Table 1 is the basic statistics of the data sets, where N, $\#Dim$ and $\#Classes$ are the number of samples, features and classes respectively.

Natural Language Datasets We evaluated our approach on the task of translation, which is widely recognized in the natural language processing. For the low-resource scenario, we used datasets from the IWSLT competitions[2], specifically the IWSLT14 $English \leftrightarrow German$ ($En \leftrightarrow De$) and IWSLT17 $English \leftrightarrow Arabic$ ($En \leftrightarrow Ar$) translations. The size of the IWSLT datasets are 175k and 241k for $En \leftrightarrow De$ and $En \leftrightarrow Ar$, respectively. For the rich-resource scenario, we used the WMT14 $English \leftrightarrow German$ dataset, which contains approximately 4.5M sentence pairs.

Computer Vision Datasets We apply a variety of datasets: both MNIST (LeCun et al. 1998) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) consist of 60,000/10,000 examples of size 28x28 pixels for the training/test set, associated with 10 classes. CIFAR10/100 (Krizhevsky, Hinton et al. 2009) has a training set with 50,000 images of size 32x32 pixels belonging to 10/100 classes.

Models and Settings

Models To illustrate various MoE's structures, Table 2 presents MoE's variations and their corresponding data types. It should be noted that T-DMoE, N-DMoE, and C-DMoE represent three DMoE variations, with the basic DNN structures, Transformer (Vaswani et al. 2017), and Convolutional Neural Network (CNN) networks, respectively applicable to tabular data, NLP data, and CV data.

In this study, Equation 3 is used for DMoE based on previous research (Eigen, Ranzato, and Sutskever 2013; Ma et al. 2018; Jacobs et al. 1991; Jordan and Jacobs 1994; Chen, Xu, and Chi 1999), while the TopK routing mechanism (Equation 4) is applied for SMoE (Lepikhin et al. 2020; Fedus, Zoph, and Shazeer 2022; Riquelme et al. 2021).

T-DMoE and T-SMoE share the same expert structure, with a 2-layer fully-connected (fc) neural network, from input dim to 16 to 10 (equal to class number), where the experts mixture occurs. We take the most popular Transformer network as the backbone architecture for N-DMoE. For the WMT experiments, the transformer _vaswani_wmt_en_de_big setting is used. Each transformer block consists of a self-attention layer, followed by a feed-forward network and a ReLU non-linearity, namely, FFN. To incorporate MoE models, we replace the FFN layer with MoE models in the 1st encoder and decoder block. Implementation is developed on Fairseq [3].

The design of C-DMoE, modified from (Svetlana et al. 2022), utilizes a convolution neural network (CNN) followed by a fc layer to output a 128-dimensional embedding, where expert mixture occurs. Subsequently, the MoE layer's output h is transmitted through a fc layer with a dimension of (128, class). In the case of MNIST and Fashion-MNIST, a 2-layer CNN is utilized, whereas for CIFAR10 and CIFAR100, a ResNet18 (He et al. 2016) is used.

Settings In this work, the number of experts N in all T-DMoE, N-DMoE and C-DMoE is set to 2, and the total number of experts N in T-SMoE is set to 10, while the number of activated experts $K = 2$. The distillation factor α is set to 0.01 or 0.1 in the tablular data sets, 1 in the NLP data sets and 10 in the CV data sets. Experiments on each dataset are repeated 10 times to calculate the mean and standard variance. All the experiments are conducted on NVIDIA A100 GPUs.

Does MoDE Work Universally?

To verify the effectiveness and universality of MoDE, plenty of experiments has been conducted including the tabular, NLP and CV datasets, where MoE has been widely applied.

Application to Tabular Datasets For each tabular data set, we sample a random 60%, 20% and 20% of the samples as the training, validation and test set, respectively.

In Table 3, we present the comparison of various models over 7 multi-category classification tasks. On each dataset, base DMoE and SMoE structures present their advantages over the single model who is identical to the individual expert's architecture. Besides, it can be observed that MoDE

---

[1]https://www.openml.org
[2]https://iwslt.org/

[3]https://github.com/facebookresearch/fairseq

Dataset	Single	Dense-Gated MoE (DMoE)		Sparse-Gated MoE (SMoE)	
		MoE	MoDE	MoE	MoDE
Isolet	0.9228±0.0064	0.9305±0.0065	0.9450±0.0074	0.9426±0.0035	0.9546±0.0034
Mfeat-karhunen	0.9388±0.0103	0.9380±0.0136	0.9525±0.0079	0.9498±0.0126	0.9603±0.0086
Mfeat-factors	0.9617±0.0080	0.9660±0.0099	0.9735±0.0091	0.9662±0.0090	0.9745±0.0048
Optdigits	0.9658±0.0081	0.9712±0.0041	0.9758±0.0038	0.9760±0.0035	0.9798±0.0040
Satimage	0.8772±0.0090	0.8872±0.0097	0.8951±0.0102	0.8945±0.0094	0.8970±0.0085
First-order-theorem-proving	0.5202±0.0156	0.5269±0.0209	0.5366±0.0210	0.5429±0.01645	0.5611±0.01499
Artificial-characters	0.5994±0.0076	0.6296±0.0113	0.6425±0.0110	0.6619±0.01372	0.6658±0.01354

Table 3: Comparisons on the benchmark tabular datasets. Top-1 Accuracy is reported (Higher is better).

Dataset	Single	MoE	MoDE
iwslt 14 de-en	34.60	34.88	**35.14**
iwslt 14 en-de	28.78	28.66	**28.91**
iwslt 17-ar-en	32.92	29.81	**33.17**
iwslt 17-en-ar	14.42	14.46	**15.40**
wmt14 en-de	27.9	29.10	**29.48**

Table 4: Comparisons on the benchmark NLP datasets. BLEU is reported (Higher is better).

Dataset	Single	MoE	MoDE
Mnist	0.9889	0.9886	**0.9918**
Fashion-M	0.8973	0.9057	**0.9083**
Cifar 10	0.9462	0.9444	**0.9519**
Cifar 100	0.7594	0.7545	**0.7824**

Table 5: Comparisons on the benchmark CV datasets. Top-1 Accuracy is reported (Higher is better).

with both gate types can give a significantly improved test accuracy than the base models, on all the tabular datasets.

Application to Natural Language Datasets All pre-processing steps follow the Fairseq (Ott et al. 2019) implementation. We calculate the BLEU scores on these tasks for evaluation (Post 2018) and follow the work (Vaswani et al. 2017) for inference. It can be observed in Table 4 that our MoDE model achieves a slight improvement in BLEU score for both low-resource and rich-resource translation tasks, indicating the effectiveness of our proposed method.

Application to Computer Vision Datasets In Table 5, the test accuracy of MoE and MoDE structures (dense gate) have been compared over 4 classic computer vision tasks. On each dataset, although base MoE structures fail to present their advantages over the single model, MoDE can still present a significantly improved test accuracy than both the single model and the corresponding MoE base models.

Through the empirical results, it can be concluded that MoDE proves its effectiveness in enhancing MoE's generalization ability and universality on the tabular, NLP and CV datasets, where MoE architecture has been widely applied.

How and Why MoDE Works?

In this subsection, a simplified parallel study has been conducted to illustrate the mechanism and the benefits of MoDE. We use the experiment of DMoE (with two experts) on the Mfeat-karhunen dataset, whose comparison has been demonstrated previously with other datasets.

Definition 1 When the gate allocates a sample to one expert with more weight (over 0.5 if there are totally two experts) than the other one, then this expert is this sample's *dominating expert, DE*. This sample is in this expert's *dominating sample-based task domain, DS*.

We propose a method called *expert probing*, in approximating each expert's test performance in its *DS*.

Methodology: Expert Probing For each test sample *t*, there exists a *DE*. By masking this *DE*'s allocated weight to 1, the MoE's prediction output can be compared against *t*'s label. Then the test performance of the *DE* on this *t* can be approximated. Therefore, this individual expert's performance on its *DS* can be evaluated.

Although this mandatory masked output is not exactly the MoE structure's opinion, to some extent, this *expert probing* method can reflect the ability of each individual expert. Moreover, considering that in MoE/MoDE the gate routes a sample to its *DE* with sloping weights (nearly 1) as shown in Table 6 and masking the gate brings a negligible effect on the MoE/MoDE's performance, the proposed expert probing method can be regarded as a reliable methodology in estimating each expert's performance.

While analyzing the performance of gate routing, we define a statistic metric *Recognition Accuracy* .

Definition 2 For one sample, if there exists at least one expert who could correctly predict it (evaluated through *expert probing*), *recognition accuracy* indicates the gate's possibility of correctly picking the expert as the *DE*.

Based on our observation on the test performance of each individual expert, the gate's *recognition accuracy* and their effects on the overall MoE's prediction performance, there are generally two types of errors.

Definition 3 Type 1 Error indicates the scenario where the sample cannot be predicted correctly by either individual expert (evaluated through *expert probing*) and the overall MoE architecture cannot thus predict the label correctly;

Definition 4 Type 2 Error indicates the scenario where the sample can be predicted correctly by at least one individual expert (evaluated through expert probing), however the gate fails to recognize the sample correctly and select the *DE* that cannot predict correctly.

It should be noted that the MoDE model significantly enhances the accuracy over the test dataset with totally 9 more correctly predicted samples. Among them, 6 and 3 samples belong to *Type 1 Error* and *Type 2 Error* observed in the base MoE model, respectively, indicating that MoDE's benefits consists of multiple factors. Table 6 has been presented

Metric	MoE	MoDE a=0.01	MoDE a=100
MoE's Overall Performance	0.9450	**0.9625**	0.9450
MoE's Overall Performance with Gate Masking	0.9400	**0.9600**	0.9425
Individual Expert's Performance on its DS	0.9275 (Exp1) 0.9669 (Exp2)	**0.9525 (Exp1)** **0.9756 (Exp2)**	0.9600 (Exp1) 0.9590 (Exp2)
Gate's Average Inclination	0.9496	**0.9661**	0.9108
Gate's Recognition Accuracy	0.9768(380/389)	**0.9795(384/392)**	1.000(377/377)
Expert Consistency	0.0150	**0.6125**	1.000

Table 6: How MoDE works? (analysis on Mfeat-karhunen dataset)

Dataset	2 Experts		4 Experts		8 Experts	
	MoE	MoDE	MoE	MoDE	MoE	MoDE
Isolet	0.9305(0.0065)	**0.9450(0.0074)**	0.9352(0.0068)	**0.9483(0.0059)**	0.9410(0.0076)	**0.9522(0.0046)**
Mfeat-karhunen	0.9380(0.0136)	**0.9525(0.0079)**	0.9432(0.0109)	**0.9543(0.0109)**	0.9433(0.0096)	**0.9602(0.0083)**
Mfeat-factors	0.9660(0.0099)	**0.9735(0.0091)**	0.9620(0.0057)	**0.9742(0.0040)**	0.9617(0.0060)	**0.9755(0.0058)**
Optdigits	0.9712(0.0041)	**0.9758(0.0038)**	0.9711(0.0025)	**0.9762(0.0042)**	0.9737(0.0060)	**0.9781(0.0043)**
Satimage	0.8872(0.0097)	**0.8951(0.0102)**	0.8925(0.0117)	**0.8950(0.0093)**	0.8990(0.0079)	**0.9005(0.0088)**
1st-order-theorem	0.5269(0.0209)	**0.5366(0.0210)**	0.5315(0.0193)	**0.5388(0.0158)**	0.5370(0.0151)	**0.5450(0.0163)**
Artificial-characters	0.6296(0.0113)	**0.6425(0.0110)**	0.6477(0.0079)	**0.6568(0.0163)**	0.6540(0.0167)	**0.6644(0.0110)**

Table 7: MoDE's robustness w.r.t more individual experts.

to briefly illustrate each expert's and the gate's performance.

The Expert Performs Better in Its *DS*. During the training process of an MoE model, the cross-entropy loss via gradient descent from random initialization will put a relatively large punishment while the gate allocates a relatively large weight on an expert who however drags the MoE's overall output away from the true label. The gate can thus learn and distinguish the coming samples according to their features and route them to the appropriate experts. Benefiting from this, each expert sees a different subset of training samples and thus obtains its advantage over these samples through training, compared with other experts, which leads to a virtuous coupling effect with the gate's routing mechanism.

Figure 3 shows that each expert of the base MoE obtains its *DS* after training: one expert is specific in the samples generally labeled with overall 7 categories and the other one is specific in the remaining 3 categories. Through *Expert Probing*, the base MoE's individual experts achieve the accuracy of 0.9275 and 0.9669 in their *DS*, respectively. In MoDE with appropriate distillation strength ($\alpha = 0.01$), each expert does not only maintain its specialization but also significantly raises its accuracy in its *DS* (through *Expert Probing*) to 0.9525 and 0.9756, respectively.

As mentioned, 6 *Type 1 Error* samples from the base MoE's individual experts are classified correctly by at least one expert in MoDE and thus predicted accurately by the overall MoDE model. Moreover, it should be necessarily noted that each of these 6 samples belongs to its correct *DE*'s original *DS*, according to the labels. In other words, with moderate mutual knowledge distillation, each expert maintains its specialization in its *DS* with increased performance.

The Gate Knows the Experts Better. Table 6 shows that in the base MoE model, the gate's *recognition accuracy* reaches 0.9769. As for MoDE ($\alpha = 0.01$), it raises to 0.9796.

Moreover, it can be found in Table 6 that the gate allocates its weights between the experts with enhanced inclination $|g_1 - g_2|$ towards each sample, averaging from 0.9496 to 0.9661, which means the gate tends to choose the appropriate expert to handle the current sample more confidently. Due to the enhancement of the gate's *recognition accuracy* in MoDE, 3 *Type 2 Error* samples in the base MoE model can be correctly routed to the correct *DE* by the gate.

Why Does MoDE Work? As observed in Figure 3, the gate distinguishes the samples' features that implicitly correspond to the labels and then make the routing decision during training. As a result, each expert meets and learns certain features preferred by the gate, towards understanding its *DS*.

As introduced and proved in Zhu's work (Allen-Zhu and Li 2020), each *DS* consists of "multi-view" data structure, where multiple features exist and can be used to classify them correctly and "single-view" data structure, where partial features for the correct labels are missing. During the training process, each expert picks up partial features corresponding to its *DS*, where most multi-view data and merely partial single-view data can be classified correctly and then contribute negligibly, because of the nature of the cross-entropy loss gradient. Due to insufficient amount of left samples, the expert just memorizes instead of learning the remaining single-view training data that the learned features cannot classify correctly. As a result, the individual expert cannot achieve the testing performance as good as training.

By applying the moderate mutual knowledge distillation to match other experts' output, the training individual expert is prompted to pay attention to other neglected features contained in the remaining single-view samples, although it has already perfectly classified the training data. Therefore, each individual expert presents improved test performance.

Will the Experts Degenerate into the Same Network? There is a concern regarding whether MoDE forces the experts to be similar and deviates the original motivation of in-
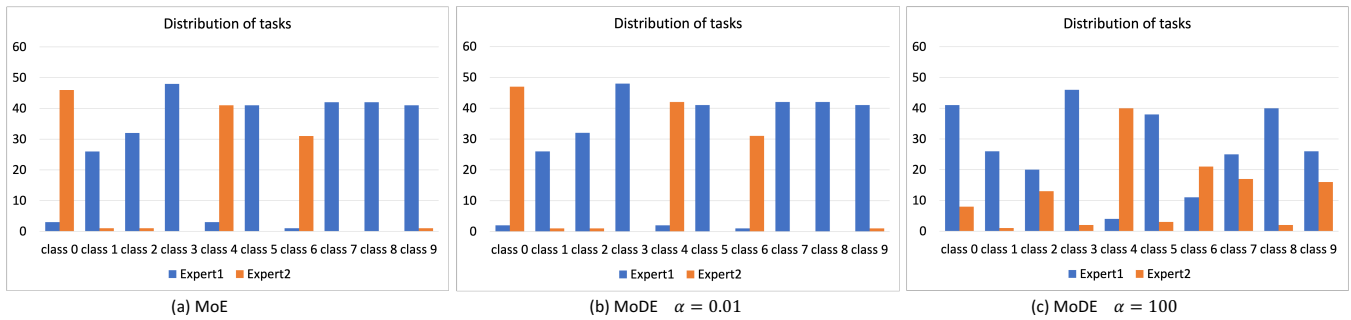
Figure 3: Task Domain Distribution of MoE, MoDE (α=0.01), MoDE (α=100).

troducing MoE: encourage each expert to be specialized and do what they are good at. To explore this, an overly heavy knowledge distillation strength ($\alpha = 100$) is placed to force both experts to express the exact opinion towards each sample. Figure 3 shows that there is a significant shift in these individual experts' **DS** and both experts presents the totally same opinion towards each sample (consistency equals to 1) as shown in Table 6. Although the gate's ***recognition accuracy*** is raised to 1, MoDE fails to achieve a higher test accuracy, because each individual expert fails to maintain its **DS** and gains better test performance in it. While with a moderate knowledge distillation strength (for example, $\alpha = 0.01$), although the individual experts' consistency is 0.6125, much higher that of base MoE (0.015), both experts still maintains their corresponding **DS**, which means each individual expert is still specialized. Furthermore, each expert also gains better perception in its **DS** as shown in Table 6. As a result, with improved gate's ***recognition accuracy***, MoDE with moderate knowledge distillation strength shows its superior generalization. Further discussions about the knowledge distillation strength are presented in the next subsection.

Ablation Study

More Individual Experts Employed by MoDE Table 7 selects tabular datasets as the benchmark to analyze the effect of the expert's amount and shows that both MoE and MoDE benefit from the increasing expert's amount with improved test performance. Moreover, the MoDE still maintains a higher accuracy than the base MoE employing the same number of experts, which means the mechanism of mutual knowledge distillation among experts works, regardless of the number of sub-networks employed.

Distillation Strength To investigate the effect of the distillation strength α, we run experiments on the tabular dataset (Isolet and Mfeat-karhunen, considering the distillation strength's scope consistency), which can be found in Figure 4. α equivalent to zero stands for the test performance of the base MoE model. As α keeps increasing and surpasses a certain point, it tends to push the experts to express overly similar opinions that have been discussed previously and fails to improve the MoE's test accuracy. The relatively wide range that the appropriate distillation strength α lays in maintains MoDE's advantage over the base MoE model
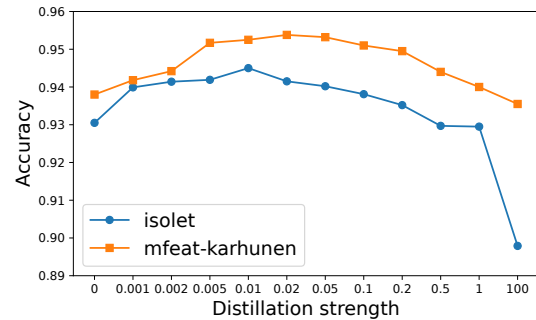
Figure 4: MoDE w.r.t the distillation strength α.

and presents its robustness to provide us a valuable exploring space in increasing MoE's generalization ability.

Conclusions

In this work, we introduce narrow vision, where each individual MoE's expert fails to use more samples in learning the allocated sub-task and thus limits the overall MoE's generalization. To address this, we propose Mixture-of-Distilled-Expert (MoDE), which applies moderate mutual distillation among the experts to encourage them to gain more accurate perceptions on their corresponding distributed tasks.

Through "expert probing", an innovative evaluation method proposed by us, we find that excessive distillation pushes the experts to presents overly similar opinions, which deviates the original motivation of MoE's structure and thus fails to improve its generalization ability. However, with moderate distillation strength, each individual expert does not only maintain its specialization in its **DS** but also gains improved test performance along with the gate, where an analysis has been experimentally and analytically conducted. Moreover, this moderate distillation strength lays in a relatively wide range, presenting the robustness.

With plenty of experiments on the datasets of tabular, NLP and CV, MoDE proves its effectiveness, universality and robustness in solving narrow vision and provide us a valuable exploring space in increasing MoE's generalization. In our future work, we plan to implement our proposed methodology in the most recent industrial scenarios, such as Large Language Model.

References

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

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

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

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

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1): 5232–5270.

Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; and Luo, P. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11020–11029.

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*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

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

Kim, J.; Hyun, M.; Chung, I.; and Kwak, N. 2021. Feature fusion for online mutual knowledge distillation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 4619–4625. IEEE.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

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

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*, 6265–6274. PMLR.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3967–3976.

Pham, L.; Phan, H.; Palaniappan, R.; Mertins, A.; and McLoughlin, I. 2021. CNN-MoE based framework for classification of respiratory anomalies and lung disease detection. *IEEE journal of biomedical and health informatics*, 25(8): 2938–2947.

Post, M. 2018. A call for clarity in reporting BLEU scores. *arXiv preprint arXiv:1804.08771*.

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

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*.

Svetlana, P.; Christian, H.; Lukas, S.; and J. Marius, Z. 2022. Sparsely-gated Mixture-of-Expert Layers for CNN Interpretability. *arXiv preprint arXiv.2204.10598*.

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

Wu, G.; and Gong, S. 2021. Peer collaborative learning for online knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, 10302–10310.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4133–4141.

Yuksel, S. E.; Wilson, J. N.; and Gader, P. D. 2012. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8): 1177–1193.

Zagoruyko, S.; and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4320–4328.

Zhu, X.; Gong, S.; et al. 2018. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31.

Zuo, S.; Liu, X.; Jiao, J.; Kim, Y. J.; Hassan, H.; Zhang, R.; Zhao, T.; and Gao, J. 2021. Taming sparsely activated transformer with stochastic experts. *arXiv preprint arXiv:2110.04260*.