Versatile Multiple Choice Learning and Its Application to Vision Computing

Kai Tian[1] Yi Xu[1] Shuigeng Zhou[1,2*] Jihong Guan[3]
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200433, China
[2]Shanghai Institute of Intelligent Electronics & Systems, Fudan University, Shanghai 200433, China
[3]Department of Computer Science & Technology, Tongji University, Shanghai 201804, China

Abstract

Most existing ensemble methods aim to train the underlying embedded models independently and simply aggregate their final outputs via averaging or weighted voting. As many prediction tasks contain uncertainty, most of these ensemble methods just reduce variance of the predictions without considering the collaborations among the ensembles. Different from these ensemble methods, multiple choice learning (MCL) methods exploit the cooperation among all the embedded models to generate multiple diverse hypotheses. In this paper, a new MCL method, called ***vMCL*** *(the abbreviation of* versatile Multiple Choice Learning*), is developed to extend the application scenarios of MCL methods by ensembling deep neural networks. Our vMCL method keeps the advantage of existing MCL methods while overcoming their major drawback, thus achieves better performance. The novelty of our vMCL lies in three aspects: (1) a* ***choice network*** *is designed to learn the confidence level of each specialist which can provide the best prediction base on multiple hypotheses; (2) a* ***hinge loss*** *is introduced to alleviate the overconfidence issue in MCL settings; (3) Easy to be implemented and can be trained in an end-to-end manner, which is a very attractive feature for many real-world applications. Experiments on image classification and image segmentation task show that vMCL outperforms the existing state-of-the-art MCL methods.*

1. Introduction

Machine learning tasks are often accompanied with ambiguity in many application areas, such as computer vision [16, 24], language understanding [11, 21], and recommendation systems [10]. Human beings interact with the world through various types of information flows. Sometimes it is hard to make a perceptron recognition in just one view. Due to the ambiguity, we cannot expect to get predictions from one model that are accurate for all data. Therefore, researchers suggested generating multiple plausible outputs [8]. This is important, especially for interactive intelligence systems, such as machine translation [1], image classification and denoising [7]. Generating multiple plausible predictions promotes the diversity of the solutions.

To generate multiple diverse predictions, two types of methods were developed. One is to train a model and generate multiple predictions during the inference process [2, 5, 12, 13]. Usually, graphical models are used to generate structured outputs. By optimizing the dissimilarity between different solutions, those methods can find a set of m-best configurations. Another is to train multiple models and aggregate their predictions to generate the final output. Such methods focus on the design of the learning process.

Among the second type of approaches, some methods ensemble many independent models and collect all the predictions into a candidate set. These methods, including Bayesian averaging [18], boosting [23] and bagging [3], often perform better than using a single model in many machine learning tasks, especially classification. As ensemble methods usually train all the embedded models independently, they may obtain low diversity in their predictions. Thus, multiple choice learning (MCL) [9] was proposed to overcome this defect by establishing cooperation among all the embedded models, each of which is trained to be a specialist on one particular data subset. The *oracle loss* concept was proposed, which focuses on one model that gives the most accurate prediction for each sample. And the *oracle error rate* was used to measure MCL performance, which means the ratio that none of the predictions are correct for the test examples.

Recently, Lee *et al*. [15] adopted deep neural networks into MCL and proposed stochastic Multiple Choice Learning (sMCL) to train diverse deep ensemble models. By minimizing the oracle loss directly, each model focuses on a subset of data to make high accuracy prediction. Although sMCL achieves high oracle performance and outperforms many existing baselines, it often fails to make satisfactory

*Correspondence author.

final decision since each network tends to be *overconfident* in its own prediction. Thus, simply aggregating these predictions by averaging or voting will result in bad final prediction. This leads sMCL to perform poorly in terms of top-1 accuracy measure and cannot be used for the scenarios that need one exact prediction. In other words, sMCL cannot take full advantage of the high oracle performance.

To solve the overconfidence problem, [14] developed the confident MCL (CMCL) algorithm that employs a new loss function named *confident oracle loss* to sMCL. The confident oracle loss adds a new term after the original oracle loss to minimize the Kullback-Leibler divergence between the predictive distribution of the non-specialized models and a uniform distribution. Although CMCL boosts the top-1 accuracy a lot, it does not take the diversity of the hypotheses into consideration which make it lose the merits of multiple choice learning. Consequently, it under-performs sMCL in terms of oracle performance.

In this paper, we argue that there exists a method that can extend MCL methods to general prediction scenario while keeps the merits of MCL settings. To this end, we develop a new MCL method, called *versatile* MCL (vMCL in short), which tries to harvest the advantages of existing MCL methods while overcoming their drawbacks effectively. Concretely, vMCL aims to maintain high diversity while restraining overconfidence. Thus, vMCL is good in terms of both *oracle* and top-1 metric which enables MCL more applicable in real-world. The major novelties and merits of vMCL are as follows: 1) A confident hinge loss is proposed to tackle the overconfidence problem, which can prevent non-specialized models from making inaccurate predictions with high confidence. 2) A choice network is adopted to learn the confidence level of each specialist, so that more reliable final decision can be obtained by aggregating the models' diverse predictions. 3) vMCL can be easily implemented and can be trained in an end-to-end manner, which is very attractive for many real-world applications.

We evaluate vMCL on two vision computing tasks: image classification and segmentation. Experiments on four public datasets show that vMCL not only significantly lifts the oracle performance (compared to sMCL) but also outperforms the existing MCL methods in terms of top-1 accuracy. For a clear and quick understanding of the advantage of our vMCL over the existing MCL methods which based on deep learning, in Tab.1 we present a qualitative comparison among sMCL, CMCL and vMCL from five dimensions: overfitting, overconfidence, hypotheses diversity, top-1 error and oracle error. In summary, vMCL is better (or not worse) than sMCL and CMCL in all the five dimensions.

Table 1: A qualitative comparison with state of the art MCL methods. 'H', 'M' and 'L' indicate *high*, *medium* and *low* respectively. For *overfitting* and *overconfidence*, 'H' is bad and 'L' is good. For *hypotheses diversity*, 'H' is good and 'L' is bad. For *top-1 error* and *oracle error*, 'H' and 'L' means large and small, and the larger the worse. In all cases, 'M' means the state between 'H' and 'L'.

MCL method	Over-fitting	Over-confident	Hypotheses diversity	Top-1 error	Oracle error
sMCL	H	H	H	H	M
CMCL	M	M	M	M	H
vMCL	L	L	H	L	L

2. Related Work

Diversity is a good way to handle the uncertainty in AI tasks. Generally, there are two types of approaches to generate multiple diverse outputs. One is to infer m-best diverse predictions from a single model, the other is to treat diversity as a learning task by training multiple models.

Most methods of the first type are probabilistic graphical models that generate multiple predictions at the inference step. [2] proposed an algorithm to generate diverse m-best solutions. They approached the m best formulations in a sequential mode, where the next solution is searched by an integer programming optimization procedure. It is a greedy algorithm that enables each prediction to be the lowest energy state but different from the previously ones. However, due to the greedy nature, each solution is only influenced by the previously predictions but not the upcoming ones. To address this issue, [12] proposed a novel formulation to jointly construct the m-best-diverse solutions by solving an energy minimization in a specifically constructed graphical model. They claimed that the method of [2] can be seen as a greedy approximation of their algorithm. Recently, DiverseNet [5] learns to produce multiple hypotheses with a control variable and for each example its training diagram requires a set of labels rather than one label.

There are different ways to train multiple models, include the classical ensemble methods. In this work, we focus on multiple choice learning (MCL), which is a novel approach to generate multiple diverse solutions. The stochastic multiple choice learning (sMCL) algorithm [15] first introduces oracle scheme into deep neural networks, then minimizes the oracle loss of multiple deep networks. As a result, each network is able to handle a subset of classes of the classification task. However, due to the oracle scheme, each sample can be assigned only to one network and do backward on that network. Thus, it is prone to overfitting for each model when training data is not enough. Besides, sMCL focuses only on the oracle performance, it may fail in scenarios that need a deterministic output. Thus, sMCL performs poorly in terms of top-1 accuracy. [22] extends the idea of sMCL by providing a mathematical understanding why this formulation is beneficial.

Recently, [14] proposed the confident MCL (CMCL) algorithm that employs a new loss function named *confident oracle loss* to alleviate the overconfidence problem

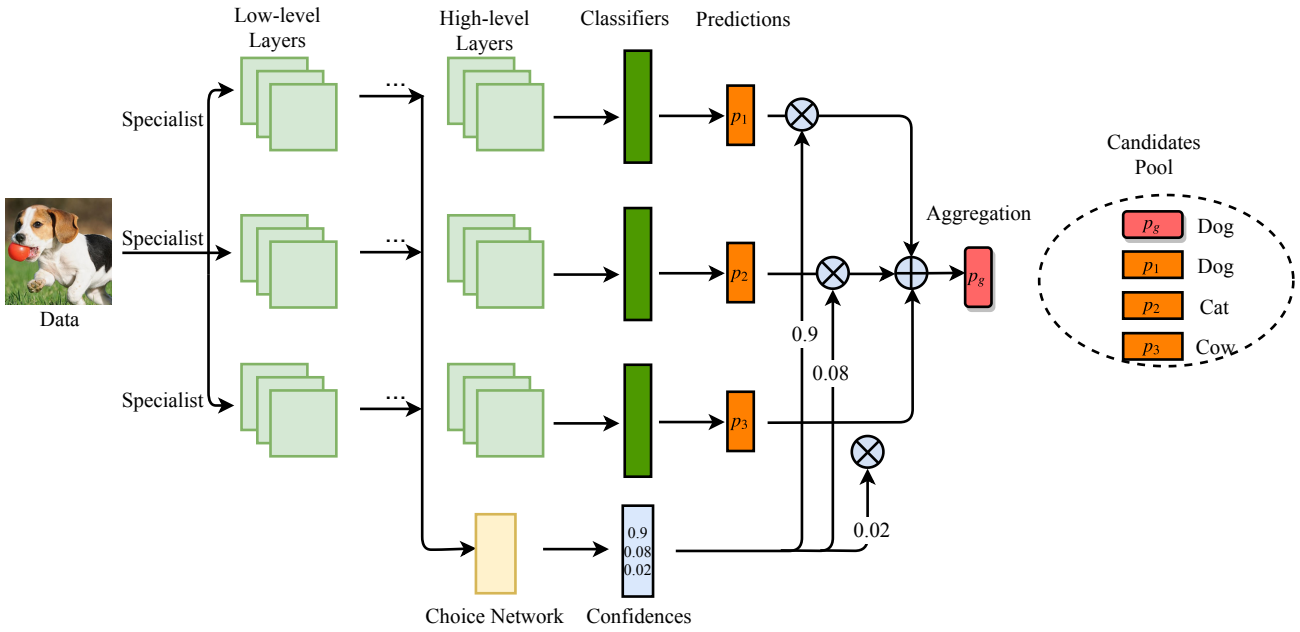


Figure 1: The architecture of our vMCL method with three networks, each of them is referred to as a specialist, where the choice network takes the concatenation of the features produced by the low-level layers as the input and generates a confidence distribution, which can be used to aggregate the diverse predictions from all the networks (specialists) for generating a high-quality final prediction. The candidates pool is used to evaluate the diversity of those hypotheses as well as the accuracy of the ensemble.

of sMCL. The confident oracle loss adds a new term after the original oracle loss to minimize the Kullback-Leibler divergence between the predictive distribution of the non-specialized models and a uniform distribution. Though CMCL boosts the top-1 accuracy a lot, the diversity of the hypotheses are much less than sMCL in many cases. This indicates that KL-divergence is a double-edged sword, it can help to avoid the overconfidence issue, meanwhile it also reduce the diversity.

3. Preliminaries

Let $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ be a dataset where each instance $\boldsymbol{x}_i$ is a training example and y_i is the label, $f_m(\boldsymbol{x})$ (m=1, ..., M) be an individual model, and M is the ensemble size (*i.e.* the number of embedded individual models). Traditional independent ensemble (IE) methods train each model over the whole dataset by adopting the following objective:

$$\min \quad \mathcal{L}_{IE} = \sum_{i=1}^{N}\sum_{m=1}^{M} \ell\Big(y_i, f_m(\boldsymbol{x}_i)\Big). \tag{1}$$

Above, $\ell(\cdot,\cdot)$ indicates a loss function. The predictions of IE methods often have low variance.

Different from traditional ensemble methods, multiple choice learning (MCL) [8] aims to specialize each individual model on a subset of the data, by minimizing the *oracle loss* as follows:

$$\min \quad \mathcal{L}_{oracle} = \sum_{i=1}^{N} \min_{m\in\{1,\dots,M\}} \ell\Big(y_i, f_m(\boldsymbol{x}_i)\Big) \tag{2}$$

As the oracle loss is a non-continuous function, an iterative block coordinate decent algorithm is used to optimize this objective function.

Stochastic multiple choice learning (sMCL) [15] implements multiple choice learning by deep neural networks with the following objective:

$$\begin{aligned} \min \quad & \mathcal{L}_{sMCL} = \sum_{i=1}^{N}\sum_{m=1}^{M} v_i^m \ell\Big(y_i, \boldsymbol{p}_m(\boldsymbol{x}_i)\Big) \\ \text{s.t.} \quad & \sum_{m=1}^{M} v_i^m = 1, \qquad v_i^m \in \{0,1\}. \end{aligned} \tag{3}$$

where $\boldsymbol{p}_m(\boldsymbol{x}_i)$ is the prediction of the m-th network, and v_i^m is an indicator variable that takes only 0 or 1. In each iteration of the training procedure, sMCL feeds the training data to each neural network and gets the output by doing forward propagation over these networks respectively. After the computation of oracle loss, it chooses the most accurate network for i-th sample, say the m-th network, and sets $v_i^m = 1$. Then, the i-th training example does backward propagation only on the m-th network. Consequently, each network performs better on some classes than the other

networks, *i.e*. each network becomes a specialist on some particular classes.

4. Versatile Multiple Choice Learning

Fig.1 shows the architecture of the vMCL method, which consists of two major parts: (a) multiple *specialist* networks; and (b) a *choice network*. Specialist networks aim to provide diverse outputs and the choice network makes the final decision of those specialists if necessary. There is no restriction on network architecture selection for these specialists, which makes our method general and flexible.

4.1. Choice Network

As we mentioned before, existing MCL algorithms perform poorly in terms of top-1 error rate because they lack a scheme to aggregate the diverse predictions from all the embedded models. These MCL algorithms are not suitable for the scenarios that do not need human interference (*e.g*. to select the best prediction). Here, we adopt a *choice network* to predict the confidence of each model. As shown in Fig.1, the choice network is deployed just after a few low-level layers of the specialists and it learns to generate the confidence of each specialist. Generally, the choice network is a neural network whose input is the concatenation of all specialist features and the output size is the ensemble size M. The target labels of the choice network are dynamically generated according to the MCL mechanism for each iteration in the training phase.

Specifically, suppose we have M networks (specialists) and $\{\boldsymbol{\theta}_m\}_{m=1}^M$ are the parameters of those specialists. The parameters of the choice network are denoted as $\boldsymbol{\vartheta}$. Let $P_{\theta_m}(y|\boldsymbol{x}_i)$ be the prediction distribution of the m-th specialist on example $\boldsymbol{x}_i$. The output of choice network for $\boldsymbol{x}_i$ can be denoted as $[w_i^i, ..., w_i^M]$, which is calculated by a *softmax* layer on the logits and thus $\sum_{m=1}^M w_i^m = 1$. $P_{opt}(c|\boldsymbol{x}_i)$ is the aggregated prediction over all the specialists that output the probability of $\boldsymbol{x}_i$ belonging to class c. Formally,

$$P_{opt}(c|\boldsymbol{x}_i) = \sum_{m=1}^{M} w_i^m P_{\theta_m}(y = c|\boldsymbol{x}_i). \tag{4}$$

It superficially looks like the mixture of experts (MoE) method [26, 19]. They all provide a way to decide how much a model can be relied on. The main difference lies in that our method has explicit target labels for the choice network, while MoE does not provide ground truth labels for the gating neural network [6] because it does not need to know which model is the specialist for a specific example. MoE considers only the correctness of aggregated output, thus it may not capable to provide multiple diverse outputs. Though there are some approaches to estimate the labels for the gating network, they are usually time-consuming.

4.2. Confident Hinge Loss

Overconfidence can be seen as a generalization issue in machine learning. It happens in many scenarios, such as imbalanced classification [4]. This problem also exists in deep learning, especially when the training dataset is not large enough. It is quite normal that a deep neural network classifies a never-seen example to some particular classes with high confidence. Recently, some solutions were proposed to deal with the overconfidence problem. For example, penalizing the confidence of outputs that have low entropy distributions [20].

Here, we propose a confident hinge loss to tackle the overconfident problem in MCL. The objective of vMCL is defined as follows:

$$\begin{aligned}\min_{\boldsymbol{v},\boldsymbol{\theta}_m,\boldsymbol{\vartheta}} \quad & \mathcal{L}(\mathcal{D}) = \sum_{i=1}^{N}\Big[\sum_{m=1}^{M} v_i^m \ell\Big(y_i, P_{\theta_m}(y|\boldsymbol{x}_i)\Big) + \ell(\boldsymbol{v}_i, \boldsymbol{w}_i) \\ & + \alpha \sum_{c\neq y_i}^{C} \max\Big(P_{opt}(c|\boldsymbol{x}_i) - P_{opt}(y_i|\boldsymbol{x}_i) + \beta, 0\Big)\Big] \\ s.t. \quad & \sum_{m=1}^{M} v_i^m = 1, \forall i \\ & v_i^m \in \{0,1\}, \forall i, m\end{aligned} \tag{5}$$

where v_i^m is the indicator variable as defined in Eq.3, and $v_i^m = 1$ means that the m-th model is the best one for the i-th example. In classification tasks, $\ell(\cdot,\cdot)$ is often selected as cross entropy function. α is a hyper-parameter for balancing the importance of the margin-based loss and β is a hyper-parameter indicating the confidence margin.

This objective function has three parts. The first part is the oracle loss, which aims to minimize the loss of the most accurate model. The second part is the choice network loss that enables vMCL to generate an accurate prediction by learning the confidence of each specialist. The choice network learns to produce the best prediction by selection or aggregation on these diverse outputs. The third part is our confident hinge loss, which aims to address the overconfidence issue. The hinge loss is set up for the aggregated prediction probability so that the correct class of each image has a higher probability than the incorrect classes by a fixed margin β.

Formally, $P_{opt}(y_i|\boldsymbol{x}_i) - P_{opt}(c|\boldsymbol{x}_i) \geq \beta$ for $c \neq y_i$. The idea is to depress the predictive probability of the non-specialists when they make wrong predictions with high confidence. Meanwhile, it promotes the specialist to predict the true labels with high probability. Compared to the confident oracle loss in [14], which desires each non-specialist to output a uniform distribution that is meaningless for the candidates pool, our new loss is a sparse regularization term, which only rectifies the incorrect predic-

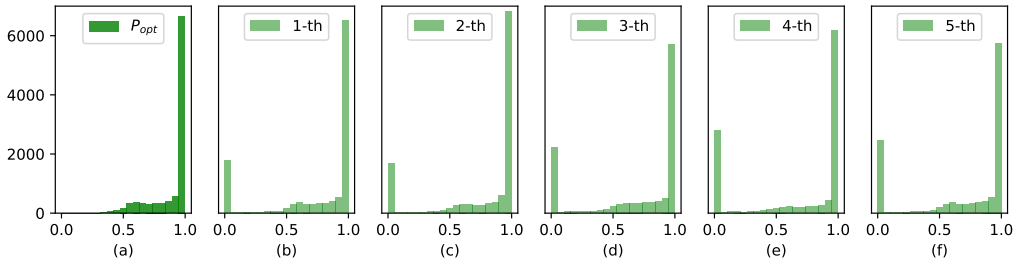


Figure 2: Histogram of the predictive distribution of our model (5 networks) tested on CIFAR-10 dataset. (a) Histogram of P_{opt} which is aggregated from all models. (b)-(f) show the probability residual between P_{opt} and P_{θ_m} for m=1, 2, 3, 4 and 5 respectively. The probability residuals near zero indicate the m-th model is specialized on these samples, while the probability residuals near 1 indicate the m-th model is a non-specialist for those data.

tions of high confidence. In other word, even if the maximal probability in some hypotheses is very high (say 0.9), it will not be punished if it does not affect the final optimal prediction. This property promotes the diversity among multiple hypotheses.

We investigate the specialization of models by analyzing their probability residuals. Here, the probability residual of the m-th model is defined as $r_m(y|\boldsymbol{x})$=$P_{opt}(y|\boldsymbol{x})$ - $P_{\theta_m}(y|\boldsymbol{x})$=$\sum_{j\neq m} w^j P_{\theta_j}(y|\boldsymbol{x})$. Some empirical results are shown in Fig.2. As shown in Fig.2(a), most optimal probabilities are near 1, which indicates that our choice network gives the optimal prediction with high confidence. Fig.2(b)-(f) show that each model only specializes a part of samples/classes. The reason is that when the residual $r_m(y|\boldsymbol{x}) \approx 0$, the optimal prediction P_{opt} is dominated by the m-th model. On the contrary, $r_m(y|\boldsymbol{x}) \approx 1$ means that the m-th model has no contribution to P_{opt}.

4.3. Training and Inference

Training: We modify the optimization algorithm of sMCL to solve Eq. (5). Considering that $\ell(\boldsymbol{v}_i, \boldsymbol{w}_i)$ is differentiable for the network parameters, and we can get the target labels $\boldsymbol{v}_i = [v_i^1, v_i^2, ..., v_i^M]$ of the choice network from the indicator variable v_i^m. The predictive distribution $\boldsymbol{w}_i$ is obtained by a softmax function on the output of the choice network. Thus, vMCL can be trained in an end to end manner. Alg.1 presents the training procedure of vMCL, which is based on stochastic gradient decent (SGD). Note that this algorithm can be easily adopted to batch-wise SGD while here we present sample-wise SGD for clarity.

Inference: For a test example $\boldsymbol{x}_i$, vMCL generates M diverse outputs $P_{\theta_m}(y|\boldsymbol{x}_i)$ (m=1,..,M) that are used to evaluate the oracle performance, and the aggregation of diverse outputs $P_{opt}(\boldsymbol{x}_i)$ is used for generating a final decision.

4.4. Feature Sharing

To cure the overfitting problem in MCL, we share the weights of a few foremost convolutional layers among the specialists, which is referred to as *shared layers*. It is different from the feature sharing method in CMCL, where the features of some specific layers are shared randomly. As previous works [25] have proven that the first a few layers learn common patterns in deep CNNs, shared layers will learn more general features on the whole dataset.

Algorithm 1 Training Algorithm of vMCL

Input: Input dataset $\mathcal{D}$, hyperparameters α, β
Output: the well-trained vMCL model.

Initialize parameters of specialists $\{\boldsymbol{\theta}_m\}_{m=1}^M$ and choice network $\boldsymbol{\vartheta}$
repeat
  Sample a batch $\mathcal{S} \in \mathcal{D}$
  for $m = 1 \rightarrow M$ **do**
    // Compute outputs for batch for each specialists
    $y_{m,1}, ..., y_{m,|\mathcal{S}|} \leftarrow P_{\theta_m}(\mathcal{S})$
  end for
  for $i = 1 \rightarrow |\mathcal{S}|$ **do**
    $v_i^m = 0, m = 1, ..., M$
    // Select lowest error model per example
    $m^* \leftarrow \arg\min_{m\in[1...M]} \ell(y_i, y_{m,i})$
    $v_i^{m^*} = 1$
    // Set the target of choice network
    $\boldsymbol{v}_i = [v_i^1, ..., v_i^M]$
    // Compute the optimal prediction
    $P_{opt}(y|\boldsymbol{x}_i) = \sum_{m=1}^M w_i^m P_{\theta_m}(y|\boldsymbol{x}_i)$
    // Compute the gradient of $\boldsymbol{\vartheta}$
    $\partial\mathcal{L}(\boldsymbol{x}_i)/\partial\boldsymbol{\vartheta}$
    // Compute the gradient of $\boldsymbol{\theta}_m, \forall m$
    $\partial\mathcal{L}(\boldsymbol{x}_i)/\partial\boldsymbol{\theta}_m$
  end for
  Update the model parameters
until convergence

5. Performance Evaluation

We evaluate vMCL on two tasks: image classification and segmentation. Three real-world image datasets includ-

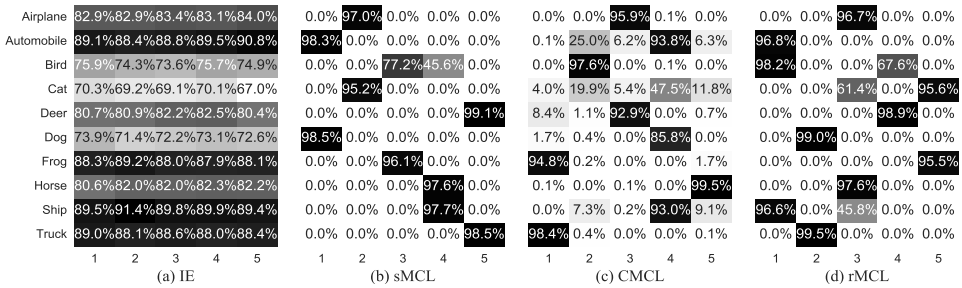


Figure 3: Class-wise accuracy of different ensemble methods on CIFAR-10. Horizontal axis indicates the model's indexes (labeled from 1 to 5) in each ensemble classifier with 5 models, and vertical axis indicates the classes in the classification task. The more concentration of each column indicates a better specialization of the corresponding model.

Table 2: Classification error rates on CIFAR-10 when different techniques are optionally used in vMCL. '$\surd$' means selected.

Ensemble Method	Shared Layers	Choice Network	Oracle Error Rate	Top-1 Error Rate
IE	-	-	7.2%	15.74%
sMCL	-	-	2.43%	54.95%
vMCL	-	-	1.79%	15.14%
	$\surd$	-	1.55%	13.74%
	-	$\surd$	1.14%	13.64%
	$\surd$	$\surd$	**1.37%**	**12.03%**

ing CIFAR-10, SVHN and CIFAR-100 are used for classification, and the image dataset iCoseg is used for segmentation. In all experiments, we compare vMCL with the traditional independent ensemble (IE), stochastic MCL (sMCL) and confident MCL (CMCL). For fairness, similar network architecture and training strategy are used for all methods.

5.1. Datasets

- **CIFAR-10** consists of 50,000 training examples and 10,000 test examples. Each image is of 32×32 pixel size and the category number is 10.
- **CIFAR-100** has the same basic statistics as CIFAR-10, except that it contains 100 classes.
- **SVHN** is a digital image dataset that consists of 73,257 training images and 26,032 test images. It has the same image size and category number as CIFAR-10. Following [14] and [27], we preprocess the images with global contrast normalization and ZCA whitening.
- **iCoseg** consists of 38 groups of images with pixel-level ground truth of foreground-background segmentation of each image. We preprocess this dataset as suggested in [14], i.e., randomly splitting the training and test sets and resizing the images.

5.2. Image Classification

Performance measures. We use the oracle and top-1 error rates to measure classification performance. The top-1 error rate is evaluated by the weighted sum of all the predictions of vMCL, and by averaging the output probabilities of all models for the other methods. The oracle error rate indicates the ratio of test images that are not correctly predicted by any specialist, which can be formulated as follows:

$$e_{oracle} = \frac{1}{N}\sum_{i=1}^{N}\prod_{m=1}^{M}\mathbb{1}(y_{m,i} \neq y_i). \tag{6}$$

$$\mathbb{1}(\mathrm{x}) = \begin{cases} 0 & \mathrm{x} = \text{False}, \\ 1 & \mathrm{x} = \text{True}. \end{cases} \tag{7}$$

Above, $\mathbb{1}(\cdot)$ is an indicator function, $y_{m,i}$ is the m-th network's prediction on the i-th sample.

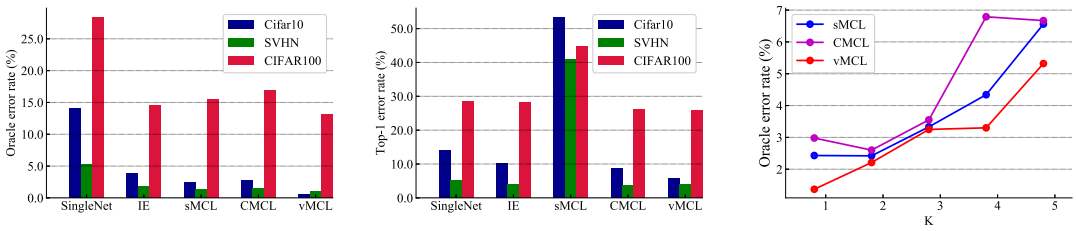
Training settings. We evaluate vMCL on a small network with 3 conv layers and a large-scale ResNet. The ensemble size of all methods is 5. The choice network is deployed just after the last convolutional layer of the specialists. All methods are optimized by SGD with an initial learning rate of 0.1, which is reduced linearly after a few epochs. We use the Nesterov momentum that is set to 0.9. The weight decay and the minibatch size are set to 0.0005 and 128, respectively. For each method, we run 5 times and average the results.

Specialization comparison. Fig.3 gives the empirical class-wise accuracy results of the four ensemble methods on the test set of CIFAR-10. For each model in these methods, the distribution of accuracy over different classes shows its specialization. The more uniform the distribution is, the less specialized the model is. We can see that IE lacks diversity as each model performs similarly and has nearly uniform distribution. sMCL and vMCL have higher specialization than the models of CMCL, as sMCL and vMCL focus on fewer classes with high accuracy than CMCL.

Ablation analysis. We check the effectiveness of major techniques (shared layers and choice network) in vMCL by conducting experiments with or without these techniques on CIFAR-10. The results are shown in Tab.2, which are compared with that of IE and sMCL.

Table 3: Performance comparison of different methods on CIFAR-10, SVHN and CIFAR100. Best results are in **bold**.

Method	CIFAR10		SVHN		CIFAR100	
	Top-1 error	Oracle error	Top-1 error	Oracle error	Top-1 error	Oracle error
IE	15.74%	7.20%	**5.64%**	3.02%	41.95%	26.49%
sMCL	58.56%	2.43%	35.74%	1.55%	52.86%	24.38%
CMCL	13.82%	2.98%	6.43%	1.62%	41.25%	26.76%
vMCL	**12.03%**	**1.37%**	5.88%	**1.22%**	**38.07%**	**19.32%**



(a) Oracle error rate of ResNet20 (b) Top1 error rate of ResNet20 (c) Overlapping K on CIFAR-10

Figure 4: (a) and (b) are classification error rates of ResNet-20 on three datasets. (c) Oracle error rate vs. K overlapping.

We can see that both shared layers and choice network can boost the performance of multiple choice learning obviously. And when both techniques are used, vMCL achieves the best performance. In the following experiments, vMCL is referred to as the one with shared layers by default.

Results on a small network. We first compare vMCL with other methods on a small network with only 3 convolutional layers and 2 fully connected layers. The results are in Tab.3. On CIFAR-10, vMCL achieves the best oracle error rate, which is nearly 43% lower than sMCL. In terms of top-1 error rate, vMCL is relatively 12.95% better than CMCL, though CMCL performs better than sMCL and IE.

Generally, images in SVHN contain relatively simpler patterns than images in CIFAR-10, they are relatively easier to be classified than that in CIFAR-10. So it is not surprised to see that IE achieves a little better top-1 error rate than vMCL. But vMCL is still better than sMCL and CMCL in top-1 error rate. And vMCL outperforms the other methods in oracle error rate, with the improvement up to 27.05%.

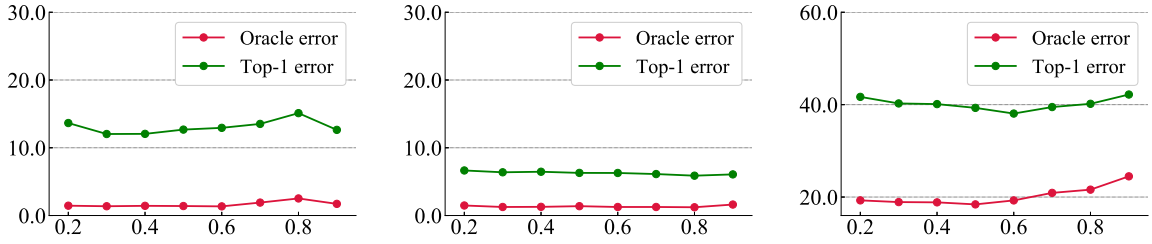
For CIFAR-100 that has a relatively large number of classes, vMCL still achieves better oracle error rate than sMCL, surprisingly with about 20% improvement. What is more, vMCL has the lowest top-1 error rate, which is about 7.27% better than that of the CMCL method.

Results on a large network. We then compare vMCL with the other methods on a large convolutional network ResNet-20, whose architecture is the same as that in [14]. We use a *single* ResNet as the baseline, denoted by *SingleNet*, and set ensemble size to 5 for the four methods. The results are shown in Fig.4(a), (b). Obviously, vMCL outperforms the other methods.

As IE lacks diversity, it performs worse than the MCL methods in oracle error rate. Each model in sMCL is designed to be specialized on some subset of the data, the overconfidence and overfitting problems make it perform poorly in top-1 measure. Although CMCL improves top-1 error rate significantly, it fails to reduce oracle error. This is because its confident oracle loss impacts the specialization of each network, by minimizing the KL-divergence between the predictive distribution on non-specialized data and the uniform distribution. Thanks to the confident hinge loss, vMCL achieves much better oracle measure than sMCL. With the shared layers and choice network, vMCL achieves the best performance in terms of oracle error rate on the three datasets. Moreover, vMCL is much better than CMCL in terms of top-1 error rate.

Overlapping effect. Here we check the effect of picking top-K best specialists at the training stage, which was also investigated in previous MCL works. By overlapping, we mean $\sum_{m=1}^{M} v_i^m = K$, where K is the overlapping size. The results are shown in Fig.4(c). As K increases, the performance of all methods turns better. However, when K approaches to the ensemble size M, the performance becomes worse, because sMCL is degenerated to IE when $K = M$.

Effect of hyperparameters. We also investigate the sensitivity of hyperparameters in vMCL. Due to the limit of space, here we present only the results of β that indicates the confidence margin of the margin loss. If we do not use the choice network, the final prediction for each test example is the average of all specialists' predictions. Thus, β is suggested to be larger than $\frac{1}{M}$, where M is the number of the specialists. As shown in Fig.5, given the dataset, the performance is quite stable when β varies. The best values of β for CIFAR-10, SVHN and CIFAR-100 are 0.3, 0.8 and



(a) CIFAR-10 (b) SVHN (c) CIFAR-100

Figure 5: Sensitivity of β on three image datasets.

Table 4: Foreground-background segmentation results on iCoseg. The model size M is varied from 1 to 5. Best results are in **bold**.

Method	IE		sMCL		CMCL		vMCL	
M	Top-1 error	Oracle error	Top-1 error	Oracle error	Top-1 error	Oracle error	Top-1 error	Oracle error
1	15.41%	15.41%	15.41%	15.41%	15.41%	15.41%	15.41%	15.41%
2	14.79%	11.60%	16.65%	10.59%	11.60%	10.82%	**10.98%**	**9.37%**
3	12.09%	10.85%	16.54%	**7.00%**	11.39%	8.26%	**10.57%**	7.02%
4	11.69%	8.57%	15.58%	6.35%	10.99%	7.77%	**9.99%**	**3.52%**
5	11.42%	7.41%	14.96%	6.35%	10.36%	7.8%	**10.28%**	**3.07%**

0.6 respectively. We also check the sensitivity of α, and find that the performance is insensitive to α value. So we set $\alpha = 1$ in all experiments.

5.3. Image Segmentation

Here, we evaluate vMCL on the segmentation task. As iCoseg is a foreground-background segmentation dataset, this task is formulated as a pixel-level classification problem with 2 classes. We select the images larger than 300×500 pixels, and randomly split the selected images into training and test datasets for each class by a ratio of 80% (training) : 20% (test). As suggested in [14], we resize the images into 75×125 using bicubic interpolation, and design a Fully Convolutional Network (FCN) [17] to do the segmentation task. For each method, we change the ensemble size from 1 to 5 and train the network up to 300 epochs.

Different from the classification task, here the prediction error rate is defined as the percentage of incorrectly labeled pixels [8]. For IE, sCML and CMCL, the top-1 error rate is measured by selecting the prediction that has a lower pixel-wise entropy among the outputs. For vMCL, the top-1 error rate is measured by using the final aggregated prediction. This is understandable as we choose the most confident prediction from a candidate set. For all methods, the oracle error rate is calculated as the lowest error rate over all outputs. We change the ensemble size from 1 to 5 and record the results of both oracle and top-1 measures. The results are shown in Tab.4.

As in the classification task, compared with sMCL, CMCL reduces the top-1 error rate significantly. However, it performs worse than sMCL in terms of oracle error rate. vMCL not only outperforms sMCL in oracle error rate but also has lower top-1 error than all the other methods. In summary, vMCL shows high specialization on the segmentation task and handles the overconfidence problem well.

6. Conclusion

This paper develops a new MCL approach vMCL for learning deep ensemble networks. vMCL aims to extend the application scenarios of deep learning based MCL methods include sMCL. By introducing some important techniques, vMCL is able to maintain the diversity among multiple hypotheses and it can aggregate a better final prediction that is better than CMCL or independent ensemble (IE) method. vMCL distinguishes itself from the existing MCL methods in four aspects: 1) using a novel confident hinge loss to address the overconfidence issue; 2) employing a choice network to aggregate the diverse predictions; 3) exploring the feature sharing technique to avoid overfitting; 4) can be easily implemented and can be trained in an end-to-end fashion. Extensive experiments on image classification and segmentation show that vMCL significantly outperforms the state-of-the-art MCL methods.

Acknowledgement

This work was partially supported by National Natural Science Foundation of China under grants No. U1636205 and No. 61772367.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Computer Science*, 2014. 1

[2] Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *European Conference on Computer Vision*, pages 1–16. Springer, 2012. 1, 2

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 1

[4] David A Cieslak and Nitesh V Chawla. Learning decision trees for unbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 241–256. Springer, 2008. 4

[5] Michael Firman, Neill DF Campbell, Lourdes Agapito, and Gabriel J Brostow. Diversenet: When one right answer is not enough. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5598–5607, 2018. 1, 2

[6] ZongYuan Ge, Alex Bewley, Christopher McCool, Peter Corke, Ben Upcroft, and Conrad Sanderson. Fine-grained classification via mixture of deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–6. IEEE, 2016. 4

[7] Gabriela Ghimpeţeanu, Thomas Batard, Marcelo Bertalmío, and Stacey Levine. A decomposition framework for image denoising algorithms. *IEEE transactions on Image Processing*, 25(1):388–399, 2016. 1

[8] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2012. 1, 3, 8

[9] Abner Guzman-Rivera, Pushmeet Kohli, Dhruv Batra, and Rob Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. In *Artificial Intelligence and Statistics*, pages 284–292, 2014. 1

[10] Hubert Kadima and Maria Malek. Toward ontology-based personalization of a recommender system in social network. In *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, pages 119–122. IEEE, 2010. 1

[11] Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1037–1045. ACM, 2011. 1

[12] Alexander Kirillov, Bogdan Savchynskyy, Dmitrij Schlesinger, Dmitry Vetrov, and Carsten Rother. Inferring m-best diverse labelings in a single one. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1814–1822, 2015. 1, 2

[13] Alexander Kirillov, Dmytro Shlezinger, Dmitry P Vetrov, Carsten Rother, and Bogdan Savchynskyy. M-best-diverse labelings for submodular energies and beyond. In *Advances in Neural Information Processing Systems*, pages 613–621, 2015. 1

[14] Kimin Lee, Changho Hwang, Kyoung Soo Park, and Jinwoo Shin. Confident multiple choice learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2014–2023. JMLR. org, 2017. 2, 4, 6, 7, 8

[15] Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pages 2119–2127, 2016. 1, 2, 3

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 8

[18] David Madigan, Adrian E Raftery, C Volinsky, and J Hoeting. Bayesian model averaging. In *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models, Portland, OR*, pages 77–83, 1996. 1

[19] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, pages 1–19, 2014. 4

[20] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 4

[21] Dan Roth. Learning to resolve natural language ambiguities: A unified approach. In *AAAI/IAAI*, pages 806–813, 1998. 1

[22] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[23] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990. 1

[24] V. Sharmanska, D. Hernández-Lobato, J. M. Hernández-Lobato, and N. Quadrianto. Ambiguity helps: Classification with disagreements in crowdsourced annotations. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2194–2202, June 2016. 1

[25] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 5

[26] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012. 4

[27] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6