Abstract

The merit of ensemble learning lies in having different outputs from many individual models on a single input, i.e., the diversity of the base models. The high quality of diversity can be achieved when each model is specialized to different subsets of the whole dataset. Moreover, when each model explicitly knows to which subsets it is specialized, the diversity can be maximized. In this paper, we propose an advanced ensemble method, called Auxiliary class based Multiple Choice Learning (AMCL), to extremely specialize each model under the framework of multiple choice learning (MCL). The advancement of AMCL is originated from three novel techniques which control the framework from different directions: 1) the concept of auxiliary class to provide more distinct information about the model’s specialization through the labels, 2) the strategy, named memory-based assignment, to determine the association between the inputs and the models, and 3) the feature fusion module to achieve generalized features. To demonstrate the performance of our method compared to all variants of MCL methods, we conduct extensive experiments on the image classification and segmentation tasks. Overall, the performance of AMCL exceeds all others in most of the public datasets trained with various networks as members of the ensembles.

1 Introduction

Ensemble learning is the process to train multiple models and combine into a single result. The spotlights to ensemble methods, even until recently from numerous applications in recognition (Siqueira, Magg, and Wermt 2020; Zhu et al. 2019), detection (Zhao et al. 2019), and segmentation (Zheng et al. 2019), are derived from their superior predictive performances compared to that of a single model. This high performance of the ensemble results primarily from the diversity of the multiple models. If all individual models provide similar outputs, any strategic combination of them cannot reduce the error. Therefore, achieving high diversity while training an ensemble model, i.e., producing a set of various possible outputs, is where the quality of ensemble originates.

Diversity can usually be achieved by alternating either datasets or model configurations, such as hyperparameters and/or initializations, for training each individual model. Relying on randomization, many of these ensembles require a good number of individual models to fulfill the law of large numbers. However, rather than differentiating models by randomness, multiple choice learning (MCL) (Guzman-Rivera, Batra, and Kohli 2012) obtains high diversity by making each model specialize to a certain subset of the dataset. These specialized models from MCL are directly trained by minimizing a new concept of loss function called oracle loss, which assigns each example to one or more, depending on the setting of the constraints, models with the highest accuracy.

Inheriting the concept of the oracle loss, Lee et al. (2016) applied MCL to ensembles of deep neural networks with a trainable algorithm and named stochastic multiple choice learning (sMCL). Since each specialized model takes the loss only from a certain set of examples, overconfident predictions to unassigned examples occur by nature. To address this overconfidence issue, a series of progressive methods are introduced. Lee et al. (2017) added a regularization term to the oracle loss in order to coerce the predictive probability of an unassigned example to be a uniform distribution. Tian et al. (2019) modified the regularization term to set a fixed margin between the aggregated probability of the ground truth and the others. In addition, a choice network is appended to the architecture to combine the predictions from all specialized models with different weights. Both methods substantially improved top-1 error rates while maintaining as good or better oracle errors compared to those of the sMCL method.

Although the accuracies of the ensembles are boosted, all previous methods, utilizing the concept of the oracle loss, do not provide explicit information about the unassigned examples to each specialized model. Setting the distribution (Lee et al. 2017) or the value range (Tian et al. 2019) of the predictive probabilities of unassigned examples is not a direct technique to encourage model specialization. The best way to lead the model specialization is to set labels of the unassigned examples with low entropies by directly indicating that the examples are not from any subsets of the model’s specialties. This clear indication leads every model in the ensemble to an extreme specialization, resulting in highly diversified outputs with lower prediction errors even when simply combined using an averaging scheme.
Inspired by this extreme model specialization, we propose an advanced method, called Auxiliary class based Multiple Choice Learning (AMCL), that expressly aims to train each base model of an ensemble to become the best specialist. As the main contributions of AMCL, we firstly develop the auxiliary class, an extra element appended to the ground truth label. This class indicates whether or not the input example belongs to a part of the model’s specialties. From a perspective of training, the auxiliary class helps to exclude the examples from the non-specialized classes. Secondly, unlike all previous MCL methods that designate each input to a particular model based on the loss value, i.e., loss-based assignment, we suggest a principled technique, i.e., memory-based assignment, based on the connections between the input examples and the models in the early iterations. Once the assignment of the inputs to the models are fixed after the first few iterations, the input examples are directly forwarded in the later iterations that helps each model to solely focus on its specialization. Finally, we propose a module to fuse low-level features from each model and train the most informative feature through the attention mechanism. This module, named feature fusion module, supports to overcome weak generalization which typically occurs under the framework of MCL (Mun et al. 2018).

We evaluate AMCL on two representative computer vision tasks: image classification and segmentation. With image classification, extensive experiments are conducted with three large-scale neural networks, VGGNet (Simonyan and Zisserman 2015), ResNet (He et al. 2016a), and GoogLeNet (Szegedy et al. 2015), on three public image datasets, including CIFAR-10/100 (Krizhevsky and Hinton 2009) and TinyImageNet (Le and Yang 2015). Compared to the existing MCL methods, AMCL significantly outperforms in both the oracle and the top-1 errors. We also provide general comparisons with the recent ensemble models, such as the method by Opitz, Possegger, and Bischof (2016) (abbreviated as EMADNN), ONE (Lan, Zhu, and Gong 2018), and KDCL (Guo et al. 2020), in which AMCL performs better than any of those baselines. Additionally, the performance on the foreground-background segmentation is evaluated with the iCoseg (Batra et al. 2010) dataset. The overall results show that our proposed method achieves impressive performance in most of the experiments, surpassing the top-1 accuracy as well as the oracle accuracy compared to the other MCL extended methods.

2 Preliminaries

In this section, we describe the formulation of the oracle loss from multiple choice learning (Guzman-Rivera, Batra, and Kohli 2012), on which all MCL related algorithms are based. Let \( D = \{ (x_j, y_j) \}_{j=1}^N \) be a dataset, where \( x_j \) represents a training example, and \( y_j \) is the ground truth label, consisting of \( N \) i.i.d. observations. Each individual model is denoted by \( f_m(x), m = 1, \cdots, M \), where \( M \) implies the ensemble size. Formally, a traditional independent ensemble (IE) method trains each model with the following objective:

\[
L_{IE}(D) = \sum_{j=1}^N \sum_{m=1}^M \ell(y_j, f_m(x_j)).
\]

where \( \ell(\cdot, \cdot) \) implies the task-specific loss function. Differently from the traditional IE method, the objective of MCL is to minimize the oracle loss, which accounts only the lowest loss from the predictions of the \( M \) models, defined by:

\[
L_{oracle}(D) = \sum_{j=1}^N \min_{m \in \{1, \cdots, M\}} \ell(y_j, f_m(x_j)).
\]

Owing to this loss function, the most accurate model for an individual example \( x_j \) is determined, and thus, each model in the ensemble is trained to become a specialist to certain subsets or classes of the dataset.

Since the oracle loss (2) is a non-continuous function, an iterative block coordinate decent algorithm is used to optimize the function. However, directly applying MCL to an ensemble of deep neural networks is not efficient due to its training costs. As a solution, Stochastic Multiple Choice Learning (sMCL) (Lee et al. 2016) proposed a training scheme based on the stochastic gradient descent algorithm to concurrently train ensembles of deep neural networks. Specifically, sMCL relaxed the objective function to the following integer programming problem:

\[
L_{sMCL}(D) = \min_{v^m} \sum_{j=1}^N \sum_{m=1}^M v^m \ell(y_j, P_m(x_j)),
\]

subject to \( \sum_{m=1}^M v^m = K \),

where \( P_m(x_j) \) is the prediction of \( m \)-th network, and \( v^m \in \{0, 1\} \) is an indicator variable representing the assignment of the input \( x_j \) to the \( m \)-th model. \( K \in \{1, \cdots, M\} \) is the number of specialized models per each input example. We note that if \( K = 1 \), the algorithm chooses the single most accurate network for every input and back-propagates only in the corresponding network. Consequently, each network is trained and specialized to a certain subset of the dataset. When \( K = M \), sMCL becomes identical to the IE method.

3 Auxiliary class based Multiple Choice Learning

In this section, we describe the technical details and novelties of our proposed training method, consisting of three major parts: (a) auxiliary class, (b) memory-based assignment, and (c) feature fusion module. These techniques combined enhance the specialization of each model in the ensembles which, in turn, ultimately promotes the performance in both oracle and top-1 error rates. We remark that the three newly introduced parts have no restriction on the architecture of each individual models, which lead our method applicable to general fields. The overall learning framework of AMCL is illustrated in Figure 1.

3.1 Auxiliary Class

The oracle loss (2) used for MCL is useful for producing diverse solutions in high quality because only the most accurate network corresponding to each input example is trained, and thus, each model eventually focuses on a certain subset of the dataset. However, this property yields to the over-confidence issue. To address this issue, confident multiple
choice learning (CMCL) (Lee et al. 2017) proposed a new loss function named confident oracle loss. The confident oracle loss adds a regularization term to the original oracle loss to minimize Kullback-Leibler divergence between the predictive distribution of an unassigned example and a uniform distribution.

To understand better about the effect of this regularization term, assume that we have an image dataset with two classes: dog and airplane. If model ‘A’ and ‘B’ are specialized in each class, dog and airplane, respectively, the predictive probabilities from model ‘A’ for any airplane images will be uniform, \([1/2, 1/2]^T\). Thus, CMCL forces each model to be less confident about non-specialized classes, and in terms of performance, it notably reduces the top-1 error while keeping the oracle error similar to that of MCL. Nevertheless, the interpretation of the probability is ambiguous since model ‘A’ returns a dog class with the confidence level of 50% to an airplane image.

From a human perspective, it makes more sense when model ‘A’ rejects an image of airplane with the confidence level of 100% because model ‘A’ is specialized in dog, not airplane. To this end, we add an auxiliary class to efficiently handle the non-specialized classes of each model. When model ‘A’ is fully trained along with this auxiliary class, the predictive probability of model ‘A’ for an input of airplane will be [0, 0, 1]^T, where the last component is the auxiliary class. This probability explicitly indicates that the input is definitely not a dog image. At the inference stage, model ‘A’ regards the probabilities of airplane images as [0, 0]^T through a simple post-process, directly removing the auxiliary class. Since model ‘B’ is specialized in the airplane class, the predictive probability to an example of airplane will be [0, 1]^T after the post-processing, and thus, the ensemble prediction via averaging scheme becomes [0, 1/2]^T.

When the prediction is normalized to make the sum equal to 1, AMCL results in [0, 1]^T while CMCL’s prediction is \([1/4, 3/4]^T\) by calculating the average between \([1/2, 1/2]^T\) from model ‘A’ and \([0, 1]^T\) from model ‘B’. Through this simple yet effective technique, AMCL produces a more intuitive and plausible solution compared to CMCL.

To demonstrate the example mentioned above, we trained an ensemble model with a simple CNN using only two classes, dog and airplane, from CIFAR-10 and compared the confidence level between CMCL and AMCL. The simple CNN consists of three convolutional layers with 32, 64 and 128 filters followed by a fully-connected layer. As shown in Figure 2, the results indeed show that the confidence levels of both classes in CMCL lie on near 0.75 while those of AMCL lie on near 1. Additionally, AMCL shows the or-
cle / top-1 error rates of 0.03% / 3.12%, which is superior to those of CMCL, 0.21% / 3.21%. AMCL achieves up to 85.71% relative reduction in the oracle error rate from the corresponding model trained with the CMCL scheme.

3.2 Memory-based Assignment

Following the constraints of the oracle loss (2), only the networks with the lowest loss for the corresponding input examples are trained with the ground truth labels. Since all the variants of MCL adopt the same oracle loss, this procedure occurs at each training step. With this strategy, loss-based assignment, the examples which do not belong to the specialized class of a model might yield to the lowest loss, and as a consequence, these wrongly assigned examples diminish the purity of the model specialization.

To tackle this problem, we introduce a simple technique to raise the purity of input to model assignment, named memory-based assignment. The memory-based assignment counts, by online manner, the number of examples per class that belongs to each model within the first few epochs of the training. The numbers of examples in some classes becomes incrementally larger than those of the other classes, so after couple epochs, the specialized classes for each model can be naturally decided. After a set number of epochs, the specialized classes are hard-coded according to the counts of examples that results in the “Fixed specialization” matrix in Figure 1. This strategy helps the model to focus more on a certain subset of the dataset for the rest of the training without any inputs that interrupt the specialization.

To describe in more detail with Figure 1, whether to assign the label to the auxiliary class or not is initially decided based on the loss values. After the set number of epochs, the specialization of each model is fixed according to the history of the past assignments, which results in the “Fixed specialization”. Therefore, as shown in the right side of the figure, even when the probability of cat turns out to be the lowest given one of the cat images, the assignment of the label to the second model is still cat, and this individual model will be trained to specialize in the images of cats.

Originated from the objective function of sMCL (3), a modified loss for our algorithm implements both of the auxiliary class and the memory-based assignment. Let \( P_{\theta_m}(\tilde{y} | x) \) be the predictive distribution where \( \tilde{y} \) is the output with the auxiliary class appended, and \( \theta_m \) denotes the \( m \)-th model parameters. \( A(\tilde{y}) \) is the one-hot encoded vector with only the auxiliary class being 1. Also, \( t_s \) is the current epoch number, and \( T_r \) denotes the threshold of epochs from loss-based to memory-based. Then, the objective of AMCL is defined as follows:

\[
L_{AMCL}(D) = \begin{cases} 
L_{LBA}(D) & \text{if } t_s \leq T_r, \\
L_{MBA}(D) & \text{otherwise}, 
\end{cases}
\]

where \( L_{LBA}(D) \) and \( L_{MBA}(D) \) are the loss functions from loss-based assignment (LBA) and memory-based assignment (MBA), which are presented below.

\[
L_{LBA}(D) = \min_{v_j} \sum_{j=1}^{N} \sum_{m=1}^{M} v_j^m \ell \left( \tilde{y}_j, P_{\theta_m}(\tilde{y} | x_j) \right) + \beta (1 - v_j^m) D_{KL} \left( A(\tilde{y}) || P_{\theta_m}(\tilde{y} | x_j) \right) \\
\text{subject to } \sum_{m=1}^{M} v_j^m = K,
\]

and

\[
L_{MBA}(D) = \sum_{j=1}^{N} \sum_{m=1}^{M} w_{c(j)}^m \ell \left( \tilde{y}_j, P_{\theta_m}(\tilde{y} | x_j) \right) + \gamma (1 - w_{c(j)}^m) D_{KL} \left( A(\tilde{y}) || P_{\theta_m}(\tilde{y} | x_j) \right),
\]

where \( D_{KL} \) denotes the KL divergence, \( \beta \) and \( \gamma \) are the penalty parameters, and \( v_j^m \) is the indicator variable as defined in (3). \( w_{c(j)}^m \) is an element of binary flag matrix of the dimension \( N_c \times M \) where \( c(j) \) denotes the class index for each input example \( x_j \), and \( N_c \) is the total number of classes. This matrix is generated by the cumulative sum of \( v_j^m \) until the epoch reaches the threshold, \( T_r \). Each element of the matrix is converted to a binary flag by choosing the top-\( K \) counts, row-wise, where \( K \) means the set number of models for each class. Therefore, \( w_{c(j)}^m \) implies that the \( m \)-th model is specialized in the class of the input example \( x_j \), and this specific class will be assigned to the \( m \)-th model until the end of the training. Therefore, AMCL trains each model to become a specialist for certain classes with the purity of high intensity compared to the other MCL methods.

3.3 Feature Fusion Module

Only with the auxiliary class and the memory-based assignment, it is challenging to overcome the weak generalization problem from each model under the MCL framework as mentioned by Mun et al. (2018). When the model specialization is completed with only the auxiliary class and the memory-based assignment, each model is trained without the general information about the whole dataset because all the examples belonging to the non-specialized classes are labeled to a single auxiliary class. Therefore, to further address this generalization issue, we suggest an additional module, called feature fusion module, in the lower level of the network with the architecture similar to the attention networks (Park et al. 2018; Woo et al. 2018). As each model in the ensemble is forced to learn low-level features differently, these features are concatenated and fed into the feature fusion module to generate a more comprehensive but concise feature, which is uniformly provided back to each model for further training. The fusion module is deployed just after a few layers of each model since the first couple layers tend to learn more common features (Yosinski et al. 2014). Owing to our feature fusion module, the different perspectives about a single input from each ensemble members are combined in the early layer, and all model can learn their own specialized expertise with more generalized features.

In the previous methods, similar techniques to enhance the low-level features are applied as well. CMCL proposes
the feature sharing scheme, which stochastically shuffles the features among the ensemble models. Meanwhile, versatile multiple choice learning (vMCL) (Tian et al. 2019) suggests to share the weights of the couple foremost convolutional layers among the models. However, in contrast to these previous techniques, feature fusion module approaches this problem in terms of learning the optimal knowledge from all models. The effect of this module compared to the other techniques is experimentally examined in Section 4.1.

4 Experiments

We evaluate the experiments on classification and foreground-background segmentation tasks to verify the effectiveness of our algorithm, AMCL. The three public image datasets, CIFAR-10/100 (Krizhevsky and Hinton 2009) and TinyImageNet (Le and Yang 2015), are used for classification, and the iCoseg (Batra et al. 2010) dataset is used for segmentation. In all experiments, we compare AMCL with others under the same experimental setups for fair comparisons. In addition to this section, the details about the experimental setups and the hyperparameter studies are provided in the supplementary material.

4.1 Image Classification

We compare all methods using three conventional large CNN models. ResNet-18 is used for the ablation and the effect studies while VGGNet-16, ResNet-34, and GoogLeNet are applied for the comprehensive experiments. All experiments in this section uses ensembles of 5 networks, $M = 5$. The experiment with a larger ensemble size is presented in the supplementary material.

Both oracle and top-1 error rates are measured for evaluating the classification performances. The oracle error is the ratio when all models failed to classify for a given test image, and the top-1 error is calculated by averaging the output probabilities of all models. We note that a lower oracle error rate implies high diversity but not always the highest top-1 accuracy.

Ablation study We study the contributions of our three major techniques under the comparison with IE and the variants of MCL. All methods in comparison are trained on CIFAR-10 with an ensemble of 5 ResNet-18. As shown in Table 1, the performance improves gradually as each technique is incrementally applied. Through this study, we note that memory-based assignment contributes more to lowering the oracle error while feature fusion module reduces the top-1 error. When all techniques are applied, AMCL achieves the best performance compared to all others. We find that AMCL provides 36.3% and 10.1% relative error reductions in the oracle and top-1 error rates, respectively, from CMCL, the second best method in terms of harmonic mean of the oracle and top-1 error.

Analysis of auxiliary class Due to the auxiliary class, AMCL effectively prevents overconfident predictions to the non-specialized classes by setting the ground truth label to $A(\hat{y})$ (See Equation (5) and (6)). Since the probability of auxiliary class explicitly indicates whether or not an input example is specialized to the corresponding model, it can be exploited to measure the uncertainty on unseen examples. To verify the quality of the uncertainty level through AMCL, we investigate the probability of the auxiliary class for the images from SVHN (Netzer et al. 2011) using an ensemble model consists of 5 ResNet-18 trained by only the CIFAR-10 dataset with the overlap parameter $K = 1$. As presented in Figure 3(a), AMCL produces a high probability towards the auxiliary class on the unseen data, which is almost as high as the probability of the non-specialized classes used in training. Additionally, we examine the cross-entropies between the ground truth labels and the predictions of the images from specialized and non-specialized classes, separately, as presented in Figures 3(b) and 3(c). The predictions are from the models trained with three variants of MCL. In contrast to the other methods, AMCL shows a large margin between the cross-entropies from the specialized and the non-specialized classes because AMCL distinctively knows the specialization of each model through the auxiliary class. This cross-entropy results demonstrate that AMCL achieves the most extreme specialization compared to the previous MCL approaches.

Effect of feature fusion module We also examine the comparative performance of feature fusion module by replacing it with the corresponding techniques from CMCL and vMCL. The dataset and the network used for this study are identical to those of the ablation study. The feature fusion module is inserted just before the first pooling layer, which is the same to the application of feature sharing in CMCL.

As shown in Table 2, the use of feature fusion module results in more improvement over the other techniques. Although the number of total parameters in the network increases by 1.8%, the improvement in classification, 16.7% and 5.8% relative error reductions in the oracle and the top-1 errors, shows that feature fusion module achieves better generalization. Additionally for a qualitative analysis, the Grad-CAM visualizations (Selvaraju et al. 2017) are examined to check the regions of the input focused to make the predictions. As presented in Figure 4, the feature maps from the model trained with our feature fusion module highlight

| Method | MBA | FFM | Oracle | Top-1 |
|--------|-----|-----|--------|-------|
| IE     | -   | -   | 5.10   | 10.18 |
| sMCL   | -   | -   | 3.27   | 56.54 |
| CMCL   | -   | -   | 3.36   | 10.83 |
| vMCL   | -   | -   | 3.54   | 11.49 |
| AMCL   | √   | -   | 2.54   | 11.63 |
|         | -   | √   | 3.13   | 10.86 |
|         | √   | √   | 2.14(-10.1%) | 9.74(-10.1%) |

Table 1: The classification error rates (%) on CIFAR-10 with the IE and the previous MCL methods. “MBA” and “FFM” are short for memory-based assignment and feature fusion module, respectively. “√” implies selected. The best score is marked in bold, and the values in parentheses as subscript represent the relative error reductions from CMCL, which is the second best result in terms of harmonic mean.
more definitive regions of the objects. Additional visualizations from both specialized and non-specialized models are presented in the supplementary material.

**Comparison with MCL methods** We now compare the performance using standard, large-scale convolutional networks on three public datasets. As summarized in Table 3, AMCL consistently shows improvements compared to the other MCL methods. Specifically, in the case with the CIFAR-10 dataset, our method with GoogLeNet shows the most impressive result producing 15.79% and 6.25% relative error reductions from the second best result in terms of the oracle and the top-1 errors, respectively. Even though both CIFAR-100 and Tiny-ImageNet are more challenging datasets with significantly larger numbers of classes compared to CIFAR-10, AMCL shows the superiority with all types of CNNs. Interestingly, we observe that only our method improves the top-1 error rate compared to the results of IE with Tiny-ImageNet, which none of the other methods have achieved.

**Comparison with non-MCL ensemble methods** To show the reliability of AMCL, we compare the performance of the recent ensemble methods, including EMADNN (Opitz, Possegger, and Bischof 2016), ONE (Lan, Zhu, and Gong 2018) and KDCL (Guo et al. 2020), on CIFAR-10/100. EMADNN enforces the diversity by minimizing the similarity between all pairs of the base models, while ONE and KDCL apply the concept of knowledge distillation (Hinton, Vinyals, and Dean 2015) among the base models. The main difference of our method compared to these previous works is to lead each base model to become a specialist for a certain subset of the dataset, not on the entire set. In this comparison, we test PreAct ResNet-10 (He et al. 2016b) and MobileNetV2 (Sandler et al. 2018) as the base models. As presented in Table 4, AMCL outperforms by large margins in terms of oracle error rates and shows impressive top-1 error rates superior to those of all baselines in most cases.

| Method | Oracle | Top-1 | Total Parameters | Relative Ratio |
|--------|--------|-------|------------------|----------------|
| WS     | 2.87   | 10.91 | 35,438K          | 1.000          |
| FS     | 2.57   | 10.34 | 56,058K          | 1.011          |
| FFM    | 2.14(-16.7%) | 9.74(-5.8%) | 56,444K | 1.018          |

Table 2: The classification error rates (%) and the number of parameters with different generalization modules. “WS” is the weight sharing from vMCL, and “FS” indicates the feature sharing scheme from CMCL. “FFM” is feature fusion module of our method. Auxiliary class and memory-based assignment are applied in all comparisons. The best score is marked in bold, and the values in parentheses as subscript represent the relative error reductions from FS, the second best result in terms of harmonic mean.

**4.2 Image segmentation**

In this section, we evaluate to verify that AMCL produces higher performance in the image segmentation task using the iCoseg dataset. To evaluate the image segmentation task, we define the prediction error rate as the percentage of incorrectly labeled pixels. The oracle error rate is defined by the lowest prediction error among the ensemble models. Meanwhile, the top-1 error is measured by the error rate of the final prediction. For a fair comparison, we report both the oracle and the top-1 errors with varying ensemble sizes. Table 5 shows that AMCL significantly outperforms all en-
Table 3: The comprehensive classification results with various CNNs. We report the mean and standard deviation (as subscript) by repeating 3 times without the standard deviation due to the space limitation. The best score is marked in bold, and the second best result is in blue.

| Model Name | Ensemble Method | CIFAR-10 | CIFAR-100 | Tiny-ImageNet  |
|------------|-----------------|----------|-----------|---------------|
| VGGNet-16  | IE              | 1.04 ± 0.12 | 3.11 ± 0.44 | 10.2 ± 0.89   |
|            | sMCL            | 1.98 ± 0.10 | 4.88 ± 0.22 | 23.7 ± 0.12   |
|            | CMCL            | 2.70 ± 0.12 | 7.52 ± 0.26 | 27.9 ± 0.28   |
|            | vMCL            | 1.75 ± 0.09 | 12.3 ± 0.33 | 32.0 ± 0.10   |
|            | AMCL            | 1.30 ± 0.12 | 7.42 ± 0.11 | 22.4 ± 0.22   |

Table 4: The classification error rates (%) on CIFAR-10/100 with various ensemble methods. All methods are used to train an ensemble of 3 networks. EMADNN uses 16 multi-branch networks, and KDCL is trained with KDCL-Naive for the ensemble logits. AMCL is trained with the overlap parameter $K = 2$. We report the mean and standard deviation (as subscript) by repeating each test 3 times. The best score is marked in bold, and the values in parentheses as subscript represent the relative reductions from the second best result in blue.

| Base Models | Ensemble Method | CIFAR-10 | CIFAR-100  |
|-------------|-----------------|----------|-----------|
| ResNet-10   | IE              | 7.60 ± 0.12 | 39.42 ± 0.37 |
|             | sMCL            | 5.0 ± 0.10  | 23.6 ± 0.28 |
|             | CMCL            | 7.92 ± 0.22 | 31.8 ± 0.10 |
|             | vMCL            | 1.76 ± 0.10 | 7.05 ± 0.22 |

Table 5: The foreground-background segmentation error rates (%) on the iCoseg dataset with varying values of the ensemble size $M$. We report the mean by repeating 3 times without the standard deviation due to the space limitation. The best score is marked in bold, and the values in parentheses as subscript represent the relative reductions from the second best result in blue.

| Method | sMCL | CMCL | vMCL | AMCL  |
|--------|------|------|------|-------|
|        | M    | Oracle | Top-1 | Oracle | Top-1 | Oracle | Top-1 | Oracle | Top-1 |
| IE     | 1    | 13.05 | 13.05 | 13.05 | 13.05 | 13.05 | 13.05 | 13.05 | 13.05 |
|        | 2    | 11.71 | 15.12 | 13.33 | 13.31 | 10.73 | 15.49 | 10.48 | 15.14 |
|        | 3    | 10.65 | 12.67 | 10.86 | 11.20 | 10.27 | 12.26 | 8.68 | 12.36 |
|        | 4    | 9.52  | 13.25 | 10.42 | 11.19 | 9.30  | 12.88 | 7.97 | 14.36 |
|        | 5    | 9.37  | 11.82 | 9.75  | 10.58 | 9.44  | 12.06 | 7.66 | 14.36 |

5 Conclusion

This paper presents an advanced ensemble method to train for an extreme model specialization which results in diverse solutions of high quality. For this purpose, we introduce simple but effective techniques: 1) auxiliary class, 2) memory-based assignment, and 3) feature fusion module. The extensive experiments with various networks and large datasets show that AMCL results in better performances in most ensemble methods. With 5 model ensemble, AMCL provides 18.2% and 18.6% relative reductions from the second best results in the oracle and the top-1 errors, respectively. These experiments demonstrate the reliability of AMCL in image segmentation beyond classification. The network architecture and the experiment results with more elaborated visualizations are illustrated in the supplementary material.
cases. We believe that our newly proposed approach can be applied in many related application fields with slight modifications and provide better results.

References

Batra, D.; Kowdle, A.; Parikh, D.; Luo, J.; and Chen, T. 2010. icoseg: Interactive co-segmentation with intelligent scribble guidance. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 3169–3176.

Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; and Luo, P. 2020. Online knowledge distillation via collaborative learning. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 11020–11029.

Guzman-Rivera, A.; Batra, D.; and Kohli, P. 2012. Multiple choice learning: Learning to produce multiple structured outputs. *Advances in Neural Information Processing Systems (NIPS)*, 1799–1807.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. *European Conference on Computer Vision (ECCV)*, 630–645.

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

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*.

Lan, X.; Zhu, X.; and Gong, S. 2018. Knowledge distillation by on-the-fly native ensemble. *Advances in Neural Information Processing Systems (NIPS)*.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*.

Lee, K.; Hwang, C.; Park, K.; and Shin, J. 2017. Confident multiple choice learning. *International Conference on Machine Learning (ICML)*, 2014–2023.

Lee, S.; Prakash, S. P. S.; Cogswell, M.; Ranjan, V.; Crandall, D.; and Batra, D. 2016. Stochastic multiple choice learning for training diverse deep ensembles. *Advances in Neural Information Processing Systems (NIPS)*, 2119–2127.

Mun, J.; Lee, K.; Shin, J.; and Han, B. 2018. Learning to Specialize with Knowledge Distillation for Visual Question Answering. *Advances in Neural Information Processing Systems (NIPS)*, 8092–8102.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. 2011. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop*.

Opitz, M.; Possegger, H.; and Bischof, H. 2016. Efficient model averaging for deep neural networks. *Asian Conference on Computer Vision (ACCV)*, 205–220.

Park, J.; Woo, S.; Lee, J.; and Kweon, I. 2018. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L. C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 4510–4520.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Conference on Computer Vision (ICCV)*, 618–626.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representation (ICLR)*.

Siqueira, H.; Magg, S.; and Wermt, S. 2020. Efficient Facial Feature Learning with Wide Ensemble-based Convolutional Neural Networks. *AAAI Conference on Artificial Intelligence (AAAI)*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 1–9.

Tian, K.; Xu, Y.; Zhou, S.; and Guan, J. 2019. Versatile multiple choice learning and its application to vision computing. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 6349–6357.

Woo, S.; Park, J.; Lee, J.; and Kweon, I. 2018. Cbam: Convolutional block attention module. *European Conference on Computer Vision (ECCV)*, 3–19.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NIPS)*, 3320–3328.

Zhao, W.; Zheng, B.; Lin, Q.; and Lu, H. 2019. Enhancing Diversity of Defocus Blur Detectors via Cross-Ensemble Network. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 8897–8905.

Zheng, H.; Zhang, Y.; Yang, L.; Liang, P.; Zhao, Z.; Wang, C.; and Chen, D. 2019. A New Ensemble Learning Framework for 3D Biomedical Image Segmentation. *AAAI Conference on Artificial Intelligence (AAAI)*, 5909–5916.

Zhu, T.; Sun, Y.; Ma, X.; and Lin, X. 2019. Hand Pose Ensemble Learning Based on Grouping Features of Hand Point Sets. *International Conference on Computer Vision Workshop (ICCVW)*, 2856–2865.

---

Park, D.; Lee, K.; Park, J.; and Kweon, I. 2018. BAM: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*.
Supplementary Material:

Auxiliary Class Based Multiple Choice Learning

A. Pseudo Code for AMCL

We describe the full training procedure of AMCL with the pseudo code presented in Algorithm 1. As explained in Section 3.2, the assignments of model specialization and the corresponding computation of loss change after a set number of epochs, $T_\tau$. When the number of epoch is smaller than $T_\tau$, an input example is assigned to the models with the lower losses. However, after $T_\tau$, the assignment is fixed by the counts from the prior iterations, and the gradient of the loss is directly computed.

**Algorithm 1:** Auxiliary class based Multiple Choice Learning (AMCL)

| Input: Training dataset $D = \{(x_j, y_j)\}_{j=1}^N$, parameters $\{\beta, \gamma\}$ and training threshold $T_\tau$ |
| Output: Trained $M$ ensemble models |

Let $A(\tilde{y})$ be a one-hot encoded vector where only the auxiliary class is set to 1 and $N_c$ denotes the total number of classes

Initialize $\{\theta_m\}_{m=1}^M$ and set $w_{mk} = 0$, $\forall k, m$

repeat

Sample a batch $B \subset D$

if $t_s \leq T_\tau$ then

/* Compute the loss for each model

$L_{m}^{j} \leftarrow \ell(y_j, P_{\theta_m}(y | x_j)) + \beta \sum_{\hat{m} \neq m} DKL\left(A(\tilde{y}) || P_{\theta_{\hat{m}}}(y | x_j)\right) \forall (x_j, y_j) \in B$

for $j = 1$ to $|B|$ do

\n
Set $v_{mk}^j = 0$, $m = 1, \ldots, M$

$m \leftarrow$ Select $K$ models with the lowest $L_{m}^{j}$

for $k = 1$ to $K$ do

$v_{\tilde{m}(k)}^j = 1$ /* Set the indicator variable

$w_{c(j)}^{\tilde{m}(k)} = = 1$ /* Count the indicator variable for $\tilde{m}(k)$-th model

end

$\partial L_{\text{loss-based}}(x_j; \beta)/\partial \theta_m$ /* Compute the gradient with respect to $\theta_m$, $\forall m$

end

else /* Binarize top-$K$ models for each class once at $t_s = T_\tau + 1$

for $k = 1$ to $N_c$ do

Binarize $[w_{1k}^j, \ldots, w_{MK}^j]$ to top-$K$

end

for $j = 1$ to $|B|$ do

$\partial L_{\text{memory-based}}(x_j; \gamma)/\partial \theta_m$ /* Compute the gradient with respect to $\theta_m$, $\forall m$

end

end

Update the model parameters

until convergence;

B. Experimental Details

In this section, we provide detailed setups used in our experiments.

B.1 Datasets

- **SVHN** is a set of real-world number images (0 through 9) with 73,257 training and 26,032 testing examples. Each image has $32 \times 32$ RGB pixels.
• **CIFAR-10** is a set of object images from 10 different classes consisting of 50,000 training and 10,000 testing examples. Each image has $32 \times 32$ RGB pixels.

• **CIFAR-100** is a dataset similar to CIFAR-10 with 100 classes instead of 10.

• **Tiny-ImageNet** is a subset of the ILSVRC 2012 classification dataset. It has 200 classes, and for each class, 500 training, 50 validation, and 50 test images are provided. All images are in $64 \times 64$ RGB pixels.

• **iCoseg** consists of 643 images in 38 groups with the pixel-level ground truths on the foreground-background segmentation. It can be considered as pixel-level two-class classification problem, *i.e.*, 0 (background) or 1 (foreground).

### B.2 Implementations

All models in our experiments are trained from scratch. For both classification and segmentation tasks, we pre-process the images with global contrast normalization and do not augment any data in all experiments. We train all models with the initial learning rate of 0.1 and drop it by 0.2 for every 25 epochs. The Nesterov momentum is set to 0.9 for SGD, and weight decay is set to 0.0005. The minibatch size is 128, and $T^*$, the epoch threshold used in memory-based assignment, is set to 5. Due to the inherent complexity of the data, we set the total epochs differently for each task. In classification, the total number of epochs is set to 100, 125 and 150 for CIFAR-10, CIFAR-100 and TinyImageNet, respectively. Meanwhile, the total number of epochs is set to 300 in the segmentation task.

In segmentation task, following the experimental setting of CMCL, we select the images that are larger than $300 \times 500$ pixels and randomly split into the training and testing sets with the ratio of 8:2 from each class. The images are resized into $75 \times 125$ using bicubic interpolation to train the network based on a Fully Convolutional Network (FCN) [2] with the decoder architecture presented in [3]. We train the FCN from scratch for all methods in comparison.
C. Effects of Memory-based Assignment and Feature Fusion Module

C.1 Assignment of the Model Specialization with Different MCL methods

Following from Section 3.2, we examine how the actual assignment of input examples to the models realize as the training epochs proceed. The ensembles with simple CNN models (See Section 3.1) are trained by CMCL, vMCL, and AMCL with the epoch threshold $T_e$ at 10. As shown in Figure 1, when the epoch is 10, about 2~3% of the inputs from each class are not assigned properly to the specialized models using CMCL. From the perspective of specialization, these inputs become potential noise, which indeed affect the performance. In the case of AMCL, each model becomes a specialist for certain classes with high purity compared to the other methods.

| Class  | AMCL | vMCL | CMCL |
|--------|------|------|------|
| Airplane | 0.0% | 0.0% | 0.0% |
| Automobile | 0.0% | 0.0% | 0.0% |
| Bird | 0.0% | 0.0% | 0.0% |
| Cat | 0.0% | 0.0% | 0.0% |
| Deer | 0.0% | 0.0% | 0.0% |
| Dog | 0.0% | 0.0% | 0.0% |
| Frog | 0.0% | 0.0% | 0.0% |
| Horse | 0.0% | 0.0% | 0.0% |
| Ship | 0.0% | 0.0% | 0.0% |
| Truck | 0.0% | 0.0% | 0.0% |

Figure 1: The change in within-class ratios for the cumulative sum of $u^m_j$ with $K = 1$ at epoch 1, 10, and 100. As the training lasts, AMCL with the memory-based assignment shows more intensive ratios towards the specialized classes for each model even with respect to $u^m_j$, which is the indicator variable from the loss-based assignment (Please refer to Equation (5) in Section 3.2).

C.2 Grad-CAM Visualizations with Different Generalization Modules

The quantitative comparisons of our technique, feature fusion module, abbreviated as FFM, to the other generalization modules, which are used in the previous MCL variants, are presented in Section 4.1. Although the changes in the classification error rates represent how much the FFM improves the performance compared to the other techniques, we additionally present the Grad-CAM visualizations [1] of the feature maps from the models trained with different generalization modules in order to check how well the model trained with FFM focuses on the discriminative regions. From both examples presented in Figure 2, it is apparent that the model trained with FFM has better focuses on the objects themselves. Especially for the case with the boy image, presented on the bottom part of the figure, the models trained with weight sharing and feature sharing do not fully concentrate on the head of the boy. These two ensembles wrongly predict the boy example to a baby.
D. In-depth Studies on the Training and the Hyperparameters

D.1 Penalty Parameters $\beta$ and $\gamma$

We investigate the performance with varying $\beta$ and $\gamma$ values to check the sensitivity. Table 1 shows the changes in the error rates with the combinations of $\beta \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$ and $\gamma \in \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$ on the CIFAR-10 dataset. From the results, we can point out that the performance of AMCL with respect to the oracle error decreases when the value of $\beta$ gets larger. In terms of the optimization process, a larger $\beta$ value imposes more focus on leading the predictions of the non-specialized models to $A(y)$ in the early iterations rather than minimizing the loss of the specialized models (Please refer to Equation (5) in Section 3.2). As this case gets extreme, some models get specialized to all the images while the others do not get specialized to any. Therefore, increasing $\beta$ naturally leads to the polarization of model specialization which is against the idea of diversifying the base models in the ensemble and results in high oracle errors. However, even when $\beta$ increases, the performance with respect to the top-1 error does
not show an explicit trend since the top-1 error is measured by averaging the predictions from all models, including the weak predictions from the non-specialized models. In contrast to the results with $\beta$, we could not find an apparent trend in the performance with respect to the value of $\gamma$. Through this study, we set the hyperparameters $\beta = 0.01$ and $\gamma = 0.5$ in all the other experiments.

| $\beta$ | 0.01 | 0.05 | 0.1 | 0.5 | 1.0 | 2.0 |
|---------|------|------|-----|-----|-----|-----|
| 2.84 / 14.31 | 2.33 / 12.83 | 2.46 / 12.26 | **2.38 / 11.62** | 2.43 / 12.05 | 2.51 / 13.63 |
| 3.23 / 14.10 | 2.65 / 12.23 | 3.07 / 11.97 | 2.25 / 12.12 | 3.24 / 13.23 |
| 3.53 / 14.39 | 3.89 / 13.12 | 3.32 / 12.60 | 3.23 / 12.75 | 4.10 / 13.80 |
| 5.06 / 12.18 | 5.94 / 12.31 | 5.22 / 12.79 | 5.32 / 12.81 | 5.95 / 11.90 | 5.62 / 12.80 |
| 5.89 / 12.52 | 5.27 / 12.26 | 5.22 / 12.10 | 5.81 / 12.36 | 6.45 / 12.07 | 6.43 / 12.90 |

Table 1: The classification error rates (%) of an ensemble of 5 ResNet-10 on CIFAR-10 with the combinations of hyperparameters $\beta$ and $\gamma$. The overlap parameter $K$ is set to 1. The results are represented as oracle / top-1. The best results, in terms of harmonic mean, are marked in **bold**.

### D.2 Overlap Parameter $K$

Similar to other MCL methods, AMCL can pick the top-$K$ specialized models instead of only one at the training stage. Overlapping implies that there exist $K$ specialized models for an individual example, i.e., $\sum_{m=1}^{M} v_{ij}^{m} = K$. As $K$ increases, the weak generalization issue can be naturally resolved since the data capacity of each model increases. Hence, all ensemble methods including AMCL tend to show better performance in the top-1 error as $K$ increases.

Table 2 compares the performance on all datasets with an ensemble of 5 GoogLeNet. AMCL consistently outperforms all baselines with respect to the oracle and top-1 error rates. As shown throughout the paper, finding the right overlap parameter $K$ affects the overall performance. However, when $K$ gets close to $M$, all MCL methods becomes the same as the independent ensemble model, and thus, $K$, smaller than $M$, have to be searched thoroughly.

| Ensemble Method | $K$ | CIFAR-10 | CIFAR-100 | Tiny-ImageNet |
|-----------------|-----|---------|-----------|--------------|
|                 |     | Oracle  | Top-1     | Oracle       | Top-1        |
| IE              | -   | 3.22 ±0.15 | 7.53 ±0.18 | 14.99 ±0.29 | 25.65 ±0.14 |
| sMCL            | 1   | 2.16 ±0.11 | 50.90 ±0.33 | 19.28 ±0.12 | 45.61 ±0.23 |
|                 | 2   | 2.08 ±0.10 | 19.77 ±0.74 | 14.58 ±0.27 | 32.56 ±0.21 |
|                 | 3   | 2.31 ±0.11 | 9.72 ±0.19  | 13.40 ±0.17 | 28.36 ±0.29 |
|                 | 4   | 2.69 ±0.17 | 8.26 ±0.08  | 13.83 ±0.31 | 26.67 ±0.38 |
| CMCL            |     | 4.57 ±0.28 | 10.08 ±0.15 | 23.64 ±0.12 | 34.55 ±0.14 |
|                 | 2   | 2.77 ±0.28 | 8.03 ±0.29  | 15.55 ±0.54 | 29.39 ±0.52 |
|                 | 3   | 2.28 ±0.08 | 7.52 ±0.26  | 13.67 ±0.10 | 26.94 ±0.19 |
|                 | 4   | 2.33 ±0.09 | 7.23 ±0.15  | 13.51 ±0.37 | 25.97 ±0.34 |
| vMCL            |     | 4.33 ±0.23 | 11.61 ±0.39 | 22.65 ±0.39 | 36.57 ±0.38 |
|                 | 2   | 3.24 ±0.26 | 9.24 ±0.21  | 15.91 ±0.26 | 32.82 ±0.41 |
|                 | 3   | 2.09 ±0.18 | 8.67 ±0.62  | 14.68 ±0.36 | 28.36 ±0.49 |
|                 | 4   | 3.01 ±0.46 | 8.23 ±0.24  | 14.11 ±0.28 | 26.67 ±0.47 |
| AMCL            |     | 3.42 ±0.12 | 8.73 ±0.17  | 17.81 ±0.11 | 33.71 ±0.23 |
|                 | 2   | 3.21 ±0.07 | 7.65 ±0.35  | 13.33 ±0.26 | 28.99 ±0.22 |
|                 | 3   | **1.76 ±0.10** | **7.05 ±0.22** | **12.33 ±0.23** | **26.25 ±0.18** |
|                 | 4   | 2.28 ±0.08 | 7.09 ±0.17  | 12.86 ±0.09 | **24.42 ±0.26** |

Table 2: The classification error rates (%) on all datasets with varying values of the overlap parameter $K$. We report the mean and standard deviation (as subscript) by repeated 3 times, and the best results are marked in **bold**.
D.3 Extension of the Ensemble Size $M$

We additionally show the superiority of AMCL with a larger ensemble size $M$. Specifically, we train the ensemble model with 10 ResNet-10 using the CIFAR-10/100 datasets with the overlap parameter $K = \{1, 4\}$. As presented in Table 3, AMCL significantly enhances the oracle and top-1 error rates compared to the previous MCL-based methods. We found that under the setting of $K = 4$, AMCL provides 12.49% and 6.10% relative reductions in the oracle and top-1 error rates from the second best result on the CIFAR-100 dataset.

AMCL seems to take more advantages of larger $M$ compared to the other methods. This is because the increase in the ensemble size $M$ causes more chances of misassigned inputs for the previous methods while AMCL does not experience these misassignments due to the memory-based assignment procedure presented in Section 3.2.

| Ensemble Method | $K$ | CIFAR-10 | CIFAR-100 |
|-----------------|-----|----------|-----------|
|                 |     | Oracle   | Top-1     | Oracle   | Top-1     |
| IE              | -   | 4.20±0.08 | 12.11±0.11| 16.49±0.19| 34.26±0.14|
| sMCL            | 1   | 0.00±0.00 | 70.23±0.15| 24.61±0.12| 68.54±0.43|
|                 | 4   | 1.54±0.07 | 21.89±0.13| 15.78±0.11| 41.69±0.28|
| CMCL            | 1   | 2.67±0.11 | 12.91±0.12| 23.31±0.08 | 43.71±0.14|
|                 | 4   | 1.22±0.06 | 9.24±0.15 | 14.09±0.17| 34.67±0.24|
| vMCL            | 1   | 2.13±0.10 | 14.36±0.13| 22.42±0.13 | 45.36±0.28|
|                 | 4   | 1.39±0.11 | 11.62±0.09| 15.14±0.15| 36.13±0.37|
| AMCL            | 1   | 0.44±0.09 | 10.44±0.11| 19.75±0.11 | 41.20±0.13|
|                 | 4   | 0.54±0.07 | 8.26±0.12 (-10.61%) | 12.33±0.12 (-12.49%)| 32.17±0.16 (-6.10%)|

Table 3: The classification error rates (%) of ResNet-10 on CIFAR-10/100 datasets with the overlap parameter $K = \{1, 4\}$ at ensemble size $M = 10$. We report the mean and standard deviation (as subscript) by repeated 3 times. The best scores are marked in bold, and the values in parentheses represent the relative reductions from the second best result in blue.

E. Network Architecture and Additional Results for the Foreground-background Segmentation

In this section, the details about the network used for segmentation and the examples of the segmentation results are presented as a supplementary to Section 4.3.

E.1 Network Architecture

The fully convolutional network, used in the foreground-background segmentation task, is illustrated in Figure 3. The structure of the network consists of encoding and decoding parts. The encoding part has two convolution-batch normalization-ReLU block operations with 128 and 256 filters consisting of $4 \times 4$ receptive fields and strides of 2. Similarly, the decoding part consists of transformed structures of the encoding part with a bi-linear upsampling layer. Our feature fusion module (FFM) is inserted after the first encoding block in order to moderate the overconfident issue by collecting features from each model in the lower layers.

![Figure 3: The detailed architecture of the network used for the segmentation task.](image-url)
E.2 Visualization of the Segmentation Results

The predictions of the foreground-background segmentation with different ensemble methods are presented with selected test images from the iCoseg dataset in Figure 4. In the AMCL results, the pixels covered in green indicate that they are classified as auxiliary class, meaning that the model is not specialized in deciding whether those particular pixels are foreground or background. The difference in specialization is apparent between two models of the ensemble with AMCL that results in larger gaps in error rates (2.6\% vs 50.0\% in the sample of Red Sox players, 6.1\% vs 40.1\% in the sample of gymnastics, and 79.3\% vs 4.4\% in the sample of airshow-plane). The last example with an airshow image is a case in which AMCL is the second best result in terms of the error rate, which is the percentage of incorrectly labeled pixels. However, the prediction output looks semantically better than that of vMCL, which is the method with the best result in this example of the airshow image. Therefore, we can discover that AMCL overall achieves both quantitative and qualitative results that outperform other variant methods of MCL.

Figure 4: Three samples of segmentation results with all the variants of MCL, including IE. The three samples are from the classes of Red Sox players, gymnastics, and airshow-plane from top to bottom.
References

[1] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[2] J. Long, E. Shelhamer, and T. Darrel. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[3] A. Radford and L. Metz and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representation (ICLR)*, 2016.