IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 June 2023, accepted 18 July 2023, date of publication 24 July 2023, date of current version 31 July 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3298057*

RESEARCH ARTICLE

# Deep Neural Network Ensembles Using Class-vs-Class Weighting

**RENÉ FABRICIUS[1], ONDREJ ŠUCH[2], AND PETER TARÁBEK[1]**
[1]Faculty of Management Sciences and Informatics, University of Žilina, 01026 Žilina, Slovakia
[2]Mathematical Institute of Slovak Academy of Sciences, 97411 Banská Bystrica, Slovakia

Corresponding author: René Fabricius (fabricius.rene@gmail.com)

The work of René Fabricius and Peter Tarábek was supported in part by the Operational Program "Integrated Infrastructure" of the Project "Integrated Strategy in the Development of Personalized Medicine of Selected Malignant Tumor Diseases and its Impact on Life Quality" ITMS code: 313011V446, co-financed by the Resources of European Regional Development Fund. The work of Ondrej Šuch was supported in part by the VEGA through "Classification using Ensembles of Neural Networks" under Grant 2/0172/22, in part by the Operational Program Integrated Infrastructure (OPII) for "InoCHF–Research and Development in the field of Innovative Technologies in the Management of Patients with Chronic Heart Failure (CHF)" under Project 313011BWH2, and in part by the European Regional Development Fund.

**ABSTRACT** Ensembling is a popular and powerful technique to utilize predictions from several different machine learning models. The fundamental precondition of a well-working ensemble model is a diverse set of combined constituents. Rapid development in the deep learning field provides an ever-increasing palette of diverse model architectures. This rich variety of models provides an ideal situation to improve classification accuracy by ensembling. In this regard, we propose a novel weighted ensembling classification approach with unique weights for each combined classifier and each pair of classes. The novel weighting scheme allows us to account for the different abilities of individual classifiers to distinguish between pairs of classes. First, we analyze a theoretical scenario, in which our approach yields optimal classification. Second, we test its practical applicability on computer vision benchmark datasets. We evaluate the effectiveness of our proposed method and averaging ensemble baseline on an image classification task using the CIFAR-100 and ImageNet1k benchmarks. We use deep convolutional neural networks, vision transformers, and an MLP-Mixer as ensemble constituents. Statistical tests show that our proposed method provides higher accuracy gains than a popular baseline ensemble on both datasets. On the CIFAR-100 dataset, the proposed method attains accuracy improvements ranging from 2% to 5% compared to the best ensemble constituent. On the Imagenet dataset, these improvements range from 1% to 3% in most cases. Additionally, we show that when constituent classifiers are well-calibrated and have similar performance, the simple averaging ensemble yields good results.

**INDEX TERMS** Pairwise coupling, multi-class classification, deep neural networks, deep ensembles, linear discriminant analysis, homoscedastic data.

## I. INTRODUCTION

The ultimate challenge in classification is achieving the highest possible accuracy. Experts often employ ensembles to gain an extra edge. Two examples of famous winners who use ensembles include the Alexnet entry in the 2012 Imagenet challenge [1] and the winning entry in the 2007 Netflix Prize [2].

Ensemble models combine the predictions of several constituent models to produce a final prediction. The following are three key reasons why ensembles yield improved accuracy [3]:

- **Representational** - overcoming the limits of the representational abilities of the individual models
- **Statistical** - circumventing the need to choose among multiple models fitting limited training data.
- **Computational** - addressing the stochastic and heuristic nature of training algorithms.

Our goal in this paper is to advance the current ensemble methods by developing a more flexible method that has a solid theoretical motivation, the ability to combine

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang.

heterogeneous deep neural networks, and a demonstrable accuracy edge on benchmark tasks.

Deep learning models generally have high variance and low bias due to their large number of parameters. The ability of ensembling to reduce the variance of its constituent models, thus providing a more robust prediction, may be the reason for the ongoing success and popularity of ensembles [4]. This popularity is illustrated by recent ensemble applications for various tasks, such as image classification [5], [6], [7], [8], audio classification [9], time series classification [10], facial expression recognition [11], predictive uncertainty estimation [12], and multilabel classification on various types of data [13].

In many of the aforementioned works (i.e. [5], [7], [8], [9], [10], [14]), pre-trained neural networks are used by applying fine-tuning. This technique enables the use of large deep-learning models with low computational expenses for training and also for problems with a small amount of training data available. Several repositories are publicly available with state-of-the-art computer vision models that are pre-trained on the large dataset ImageNet21k [15]. Repository [16] contains multiple pre-trained convolutional neural networks, and repository [17] can be used to find vision transformers and architectures that incorporate them.

Ensembling approaches can be divided into two groups based on the training procedure of their constituents:

- Randomization-based
- Boosting-based

Randomization-based ensembles can combine constituents trained in parallel. A well-working randomization ensemble requires its constituents to provide diverse predictions. Randomization ensembles can generally be applied as a post-processing step on an arbitrary set of trained classifier instances. In contrast, boosting-based ensembles train their constituents in series. The training process of each constituent is dependent on previously trained constituents. Boosting-based ensembles cannot combine an arbitrary set of trained classifier instances which disqualifies them as a simple post-processing approach. In this work, our focus is on randomization-based ensembles that do not impose any requirements on the training process of the ensemble members.

The majority of the aforementioned applications use randomization-based ensembling approaches simple or weighted averaging. These strategies assign weights to constituent classifiers and perform a linear combination of predictions either in probability space or in logit space [6]. In the case of simple averaging the weights are uniform, whereas in the case of weighted averaging, the weights can be proportionate to the accuracies of the constituent classifiers [18] or be learned on a validation set [6]. Weighted or unweighted averaging is simple and successful, and is, therefore, widely used. However, it lacks the ability to utilize the unique properties of the combined classifiers. A single weight per constituting classifier doesn't enable consideration of the varied distinguishing abilities of different models across various classes or pairs of classes. Considering the availability of a wide palette of diverse classifier models, this constraint could present a limiting factor when utilizing simple weighted ensembles. To alleviate these limitations, we propose a novel ensembling approach that is able to assign a unique weight to each pair of classes for each of the constituent classifiers. Our strategy can be interpreted as a stacking ensemble with a meta-learner that learns pairwise weights using the outputs of the constituent classifiers. The combined pairwise predictions are processed by a pairwise coupling method that produces the final multi-class prediction. Our strategy provides a general scheme with modularity in the choice of the meta-learner and of the pairwise coupling method. We denote the proposed strategy as the pairwise-weighted ensemble (PWE). In Table 1, we provide an overview of existing weighting-based ensembles, including our proposed approach.

There has also been recent progress in boosting-based deep ensembles utilizing transfer learning [14], Residual Networks [19], or snapshot ensembles [20]. However, these approaches require a specific training regime for ensemble constituents that is only applicable sequentially. Consequently, boosting-based ensembling approaches are out of the scope of this work.

Our work offers the following contributions:

1) We propose a novel classification ensembling approach that assigns weights to each pair of classes for each of the combined classifiers. Our ensembling approach includes two modular steps, which allows for multiple model configurations.
2) We present two specific configurations of our approach and empirically evaluate them alongside a baseline in the form of a popular averaging ensemble. Using the datasets CIFAR-100 and ImageNet1k, we demonstrate that in the majority of the examined cases, our method provides more accurate predictions than the baseline.
3) We present a theoretical scenario in which our approach yields optimal classification.

## II. METHODOLOGY

The proposed ensemble method transforms the multiclass predictions of ensemble members into a set of binary predictions for each pair of classes. The fusion of predictions originating from different ensemble constituents is created as a linear combination in the space of binary classification problems. This enables our method to utilize differing capabilities of combined classifiers to distinguish between specific pairs of classes. The situation with differing capabilities of combined classifiers is visualized in Fig. 1 of 20 pairs of classes picked from the ImageNet1k dataset [21]. We can see that the accuracies of the four classifiers differ significantly for each of these pairs. We can also observe that the order from the most accurate to the least accurate classifier changes across different pairs. Displayed pairs of

**TABLE 1. Overview of weighted ensembling approaches.**

| combination type | weighting scheme | weight values | ensemble input | references |
|---|---|---|---|---|
| simple averaging | constituent averaging | uniform | probabilities | [5]–[10], [12], [13] |
| | | | logits | [6] |
| weighted combination | constituent weighting | based on accuracy | probabilities | [18] |
| | | trained | logits | [6] |
| | class pair weighting | trained | logits | proposed approach |

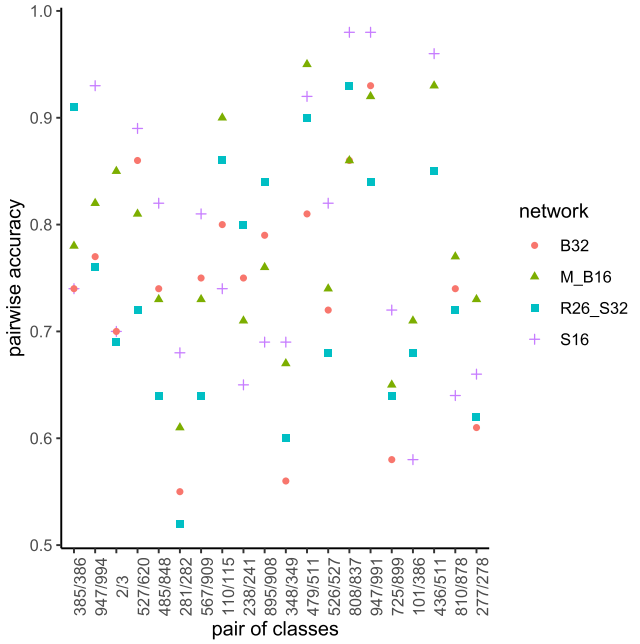


**FIGURE 1. On the horizontal axis, there are the 20 picked pairs of classes. On the vertical axis, there are pairwise accuracies of the four examined neural networks. Classes are indexed from 0 according to alphabetically ordered ImageNet1k class names. Details about the examined networks can be found in the section IV.**

classes were picked as those with the highest variance among combined classifier pairwise accuracies. For many pairs of classes all of the studied classifiers attain almost perfect pairwise accuracy.

After the combination in the binary problems domain is performed, we have a single prediction for each pair of classes. To produce the final multiclass prediction, we utilize a pairwise coupling method. Pairwise coupling methods have their origin in extending the binary classifier of Support vector machines (SVM) (and others) to a multiclass classifier [22]. Several other pairwise coupling methods have been proposed since. We report experimental results for two of them, incorporated into our proposed fusion strategy. These are the methods explained in [23] as the first and second approaches. In our work, we refer to them as **m1** and **m2**, respectively. The second method **m2** is implemented in the popular library LIBSVM [24].

### A. PROPOSED METHOD

Our method consists of three steps - extraction of logits, finding a linear combination of logits for each pair of classes, and finally, a pairwise coupling method to arrive at the multiclass prediction.

The first step in the proposed method is to obtain binary classifications from the multiclass classifications of the ensemble constituents. Let $\mathbf{p}^c$ be the probability distribution estimate outputted by the constituent classifier $c$. The output of a binary classifier distinguishing between classes $i$ and $j$ formed from the multiclass classifier $c$ can be expressed as

$$\left(\frac{p_i^c}{p_i^c + p_j^c}, \frac{p_j^c}{p_i^c + p_j^c}\right). \tag{1}$$

Equation (1) assumes that the axiom of independence from irrelevant alternatives (IIA) holds for the outputs of combined classifiers. In the technical implementation, we work with a mathematically analogous approach, where we use logit vectors instead of probabilities. Logits are inputs into the final softmax layer in the combined neural networks. Let $\mathbf{l}^c$ be the logit vector extracted from a combined classifier $c$. In place of probabilities given by (1) we use logit differences computed as

$$\left(l_i^c - l_j^c, l_j^c - l_i^c\right). \tag{2}$$

Note that the difference for the more probable class is positive, whereas, for the other class, the difference is negative. This strongly suggests the use of linear classification methods without the intercept rather than using models with the intercept.

The second step is the combining of binary predictions obtained by (2) from all of the constituent classifiers $c \in (1, \ldots, C)$. This combination is performed separately for each pair of classes. We do this by performing a linear combination of logit differences and by applying function $\text{expit}(x) = \frac{\exp(x)}{1+\exp(x)}$ to the combination result. Formally it can be expressed as

$$r_{ij} = \text{expit}\left(\sum_{c=1}^{C} w_{ij}^c (l_i^c - l_j^c) + b_{ij}\right). \tag{3}$$

Weights $w_{ij}^c$ can be unique for each classifier $c \in (1, \ldots, C)$ and each pair of classes $i, j \in (1, \ldots, K), i \neq j$, with $K$ being the number of classes, whereas biases $b_{ij}$ are tied only to pairs of classes. Binary probabilities $r_{ij}$ for each pair of classes $i, j \in (1, \ldots, K), i \neq j$ form a matrix $\mathbf{R}$ of combined binary classifications.

The third step is the processing of matrix $\mathbf{R}$ by a pairwise coupling method. The output of the pairwise coupling method is the resulting probability distribution estimate $\mathbf{p}$.

What remains to be clarified is the way to determine values for weights $w_{ij}^c$ and biases $b_{ij}$. This task can be interpreted for each pair of classes $i, j$ as a binary classification problem with predictors given by $l_i^c - l_j^c$ for $c \in (1, \ldots, C)$ and target given by the correct class $i$ or $j$. If we solve this problem by a linear classification method such as linear discriminant analysis (LDA) or logistic regression [25], their learned parameters exactly correspond to the coefficients $w_{ij}^c$ and bias $b_{ij}$ that we need. In the Experiments section, we report results obtained using logistic regression without intercept term and denote them as **logreg_ni**. We use L2 regularization for both methods.

Another way of looking at this problem is through the optimization of the final multiclass prediction $\mathbf{p}$. Pairwise coupling methods we work with can all be implemented using only differentiable operations. This enables the back-propagation of gradients and the use of stochastic gradient descent (SGD). This way we can train all the weights $w_{ij}^c$ and biases $b_{ij}$ simultaneously. As the loss function for the SGD, we use negative log-likelihood (NLL) with L2 regularization. In the Experiments section we report results obtained through gradient training with the m2 pairwise coupling method and denote them as **grad_m2**.

At prediction time, every method of training ensemble weights can be combined with every pairwise coupling method. We denote these configurations as *weight training method + pairwise coupling method*. Ensemble configuration formed by training the weights using logistic regression without intercept and m1 pairwise coupling method will be referred to as **logreg_ni + m1**.

### B. FAST INFERENCE MODIFICATION

The proposed ensembling approach has quadratic complexity in the number of classes. This could pose a limitation mainly for the inference on large problems with many classes. We propose a modification of the inference process that would alleviate this limitation.

Before the ensembling starts, we have available the logits of constituent classifiers. From these, we can infer a subset of classes deemed to be the most probable by the constituent classifiers. We can do this by looking at the *topl* classes with the highest logits for each constituent classifier and by creating a union of these classes across all the constituents. We can then perform the ensembling on a problem consisting only of this subset of classes. Classes not belonging to this subset are assigned a zero probability in the final multiclass ensemble prediction.

This approach is controlled by the hyperparameter *topl*, which can be set as a positive integer up to the number of classes in the problem. If *topl* is equal to the number of classes in the problem, inference works without modification.

### C. BASELINE METHOD

As a baseline method, we have chosen a method based on averaging strategy. The averaging strategy is simple, powerful, and widely used. However, this strategy can have problems in case of poorly calibrated ensemble members [6]. For this reason, we perform calibration using temperature scaling [26] for each ensemble member before the averaging. Temperature scaling works by scaling the logits of a classifier before the softmax function in the final layer is applied. Scaling is performed by a temperature $T$ which is learned to minimize NLL on the validation set. Prediction of a classifier $c$ with applied temperature scaling is given by

$$p_i^c = \frac{\exp(l_i^c/T^c)}{\sum_{j=1}^K \exp(l_j^c/T^c)}, \text{ for } i \in (1, \ldots, K), \tag{4}$$

where $\mathbf{l}^c$ is the vector of logits and $K$ is the number of classes. This approach effectively performs a variant of the weighted averaging fusion strategy.

## III. THEORETICAL ANALYSIS OF HOMOSCEDASTIC SCENARIO

We illustrate the learning capacity of the proposed method on a model case. We will show that the proposed method is able to recover the optimal classifier given an ensemble whose elements carry complementary information.

Suppose we have three classes $C_1, C_2, C_3$ with three different real-valued features $\mathbf{x} = (x_1, x_2, x_3)$. Suppose that each class is distributed as multivariate normal distribution in $\mathbf{R}^3$ with identical covariance matrix $\Sigma$ so that

$$p(\mathbf{x}|C_i) \sim N(\mathbf{m}_i, \Sigma). \tag{5}$$

Now suppose our ensemble consists of the three classifiers where each saw only one of the features during training. Assume that each is the best possible classifier given the one-dimensional feature $x_k$. The optimal multi-class classification given dimension $x_k$ is stipulated by the Bayes theorem

$$p(C_i|\mathbf{x_k}) = \frac{p(\mathbf{x_k}|C_i)p(C_i)}{\sum_j p(\mathbf{x_k}|C_j)p(C_j)}. \tag{6}$$

It follows that

$$\log \frac{p(C_i|x_k)}{p(C_j|x_k)} = \log \frac{p(x_k|C_i)}{p(x_k|C_j)} + \log \frac{p(C_i)}{p(C_j)}. \tag{7}$$

Any linear transformation of a normal distribution is again a normal distribution. Concretely, the projection of the distribution of $C_i$ to $k$'th coordinate $x_k$ is distributed as

$$\sim N(\mathbf{e}_k'\mathbf{m}_i, \mathbf{e}_k'\Sigma\mathbf{e}_k) =: N(m_{ki}, \Sigma_k) \tag{8}$$

where $\mathbf{e}_k$ is the $k$-th unit vector.

We have that logarithm of probability distribution function $p_N(\mu, \sigma^2)$ of normal distribution $N(\mu, \sigma^2)$ is

$$\log p_N(\mu, \sigma^2) = -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 - \log(\sigma\sqrt{2\pi}) \tag{9}$$

Therefore we have

$$\log \frac{p_N(m_{ki}, \Sigma_k)}{p_N(m_{kj}, \Sigma_k)} \tag{10}$$

$$= -\frac{1}{2}\left\{ \frac{(x_k - m_{ki})^2}{\Sigma_k} - \frac{(x_k - m_{kj})^2}{\Sigma_k} \right\} \tag{11}$$

$$= \frac{1}{2\Sigma_k}\left( 2x_k(m_{ki} - m_{kj}) + (m_{kj}^2 - m_{ki}^2) \right) \tag{12}$$

Note that this is an affine transformation of the coordinate $x_k$, namely a translation of $\frac{m_{ki} - m_{kj}}{\Sigma_k} x_k$. It follows from (7) that the same holds for $\log \frac{p(C_i|x_k)}{p(C_j|x_k)}$ which is the expression we use in (2).

The class boundary in three-dimensional space between classes $C_i$ and $C_j$ is just a linear hyperplane. This is essentially the result of Fisher in his ground-breaking paper [27] on linear discriminant analysis (LDA).

Assume for the moment that $m_{ki} \neq m_{kj}$ for $i \neq j$. Then by taking a linear combination of (7), we can fit any linear hyperplane. If we optimize, as we do in our second step, we will fit the linear boundary between classes $C_i$ and $C_j$ in $\mathbf{R}^3$.

What happens if for some $i \neq j$ we have $m_{ki} = m_{kj}$? Recall that the normal to the LDA hyperplane is given by $\Sigma^{-1}(\mathbf{m}_i - \mathbf{m}_j)$. It follows that the coefficient for $x_k$ of the optimal hyperplane is zero and thus this does not prevent us from fitting the optimal hyperplane.

So far we deduced that we find optimal pairwise classifiers. It remains to show that by coupling we obtain the optimal multi-class classifier. In this paper we use two coupling methods devised by Wu et al. [23]. They aim to (approximately) solve the (likely inconsistent) system of Bradley-Terry equations

$$r_{ij} = \frac{p_i}{p_i + p_j}, \tag{13}$$

where $r_{ij}$ is given and represents the probability of $i$-th class being the right one, if one assumes the right classification is either $i$-th or $j$-th class. Concretely, they respectively minimize the two quadratic forms

$$\min_{\mathbf{p}} \sum_i \Big( \sum_{j:j\neq i} r_{ji} p_i - \sum_{j:j\neq i} r_{ij} p_j \Big)^2, \quad \text{and} \tag{14}$$

$$\min_{\mathbf{p}} \sum_i \sum_{j:j\neq i} \big( r_{ji} p_i - r_{ij} p_j \big)^2, \tag{15}$$

under the restrictions $\sum_i p_i = 1$ and $p_i \geq 0$.

Since we find the optimal pairwise classifier for each pair of classes, the resulting set of Bradley-Terry equations is consistent. In this case, both coupling methods attain zero lower bound in optimization of (14) and (15) and thus yield in the third step of our method the optimal multi-class prediction, as we desired to demonstrate.

Let us discuss briefly the special case when features $x_k$ are uncorrelated. The matrix $\Sigma$ is diagonal and (12) shows that the normal of the optimal (LDA) hyperplane is just the arithmetic average of log-odds of the ensemble members. Transferring from log-odds to probabilities means that the optimal multi-class decision is just the rescaled geometric average of the multi-class predictions.

Our analyses crucially used homoscedasticity and normality assumptions. In real applications, these assumptions will rarely hold, which leads us to consider logistic regression instead. Logistic regression fails to converge under maximum likelihood estimation if the classes are linearly separable. This problem can be countered by employing regularization. Note that although in general the effect of regularization depends on the scaling of variables, in our case the pairwise logits are canonically scaled.

## IV. EXPERIMENTS

We performed the evaluation of the proposed method PWE on standard benchmarks for computer vision CIFAR-100 [28] and ImageNet1k [21]. We also used the CIFAR-10 dataset [28] for hyperparameter tuning.

For each of the evaluated ensemble models, we compute its accuracy improvement over its most accurate constituent. We use these accuracy improvements to be able to compare ensemble models constituting of different base models with different accuracies.

In each of the following experiments, we evaluate the averaging baseline and two configurations of the proposed approach **grad_m2 + m2** and **logreg_ni + m1**.

We also perform comparisons of the accuracy improvement between the baseline ensemble and the proposed approach by means of statistical testing. We use permutation tests [29] to carry out these comparisons.

To obtain a diverse set of ensemble constituents we have used neural networks with different architectures and also networks pre-trained on different datasets.

### A. CIFAR-100

During our experiments on the CIFAR-100 dataset, we have used a total of seven architectures trained from scratch. These architectures are: googlenet [30], resnet34 [31], stochasticdepth50 [32], resnext101 [33], densenet121 [34], xception [35] and seresnet34 [36]. We obtained these networks from a GitHub repository [37]. Apart from these, we have also used a feature extractor CLIP [38] pre-trained on 400 million image-text pairs obtained from publicly available sources on the internet. We obtained this model from a GitHub repository [39]. We have fine-tuned two architectures of this model using the linear probe method in which only the weights of the final layer are trained.

For the evaluation on the CIFAR-100 dataset, we have split the training set into two parts. The first part, containing 45000 samples, was used to train the neural networks. The second part, containing 5000 samples, was used to train the combining coefficients of our ensembling method. The second part was also used to determine the calibration temperatures of the baseline ensembling method. We performed the

**TABLE 2.** Accuracy of neural networks trained (or fine-tuned) on the CIFAR-100 dataset.

| neural network | accuracy |
|---|---|
| googlenet | 0.7631 |
| stochasticdepth50 | 0.7631 |
| resnext101 | 0.7734 |
| seresnet34 | 0.7746 |
| clip_ViT-B-32 | 0.7980 |
| clip_ViT-B-16 | **0.8221** |

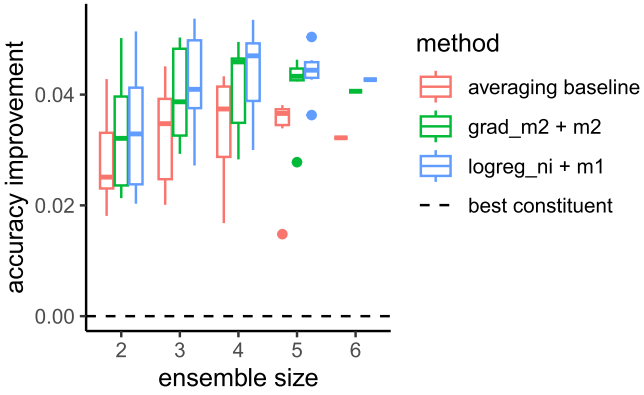


**FIGURE 2.** Improvement in accuracy on the CIFAR-100 dataset for ensembles built from all combinations of networks from Table 2. On the vertical axis is ensemble improvement in accuracy over the best of the constituent models. On the horizontal axis is the number of combined models. Accuracy attained by the best ensemble constituent is displayed as a horizontal line at 0.

**TABLE 3.** Results of statistical tests comparing PWE and baseline ensemble on the CIFAR-100 dataset. The results are in favor of PWE in all the cases.

| configuration | ensemble size | p-value |
|---|---|---|
| grad_m2 + m2 | 2 | 0.0000 |
| | 3 | 0.0001 |
| | 4 | 0.0001 |
| logreg_ni + m1 | 2 | 0.0005 |
| | 3 | 0.0001 |
| | 4 | 0.0002 |

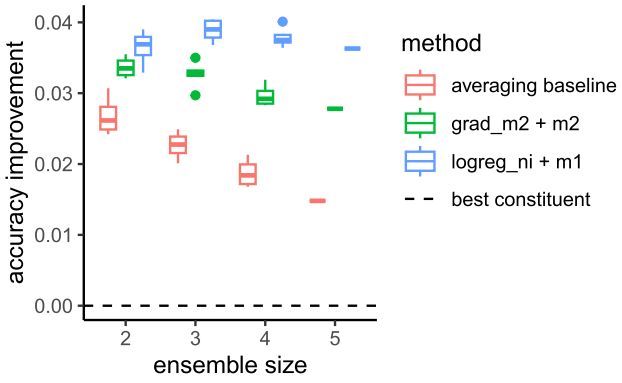


**FIGURE 3.** Improvement in the accuracy of a subset of ensembles. Each ensemble in the subset contains clip_ViT-B-16 and does not contain clip_ViT-B-32. The plot structure is the same as for Fig. 2.

split in a way that maintains an equal proportion of each class in both parts. We have trained four networks from scratch and fine-tuned two architectures of CLIP using the aforementioned training set split. Accuracies of these networks are displayed in Table 2. We can see that CLIP models have higher accuracy than models trained from scratch, especially dominant is clip_ViT-B-16.

Ensembling approaches that we are using can combine an arbitrary number of constituents. We test all possible combinations of constituents for each ensemble size from 2 up to 6.

Ensemble improvements for all network combinations are displayed in Fig. 2. All examined ensembling methods obtained accuracy improvements over the most accurate constituent ranging from 2% to 5% in the most cases. We have performed statistical tests comparing our approach and the baseline for ensemble sizes 2 up to 4. Results of the performed tests are displayed in Table 3. At significance level 5% all these tests show an advantage of the proposed PWE approach. Ensemble sizes 5 and 6 do not contain enough constituent combinations to perform statistical tests. However, the advantage of the proposed approach for these cases is quite clear from Fig. 2.

In Fig. 2 we can also observe, that accuracy improvements of all ensembling methods are increasing with the increasing ensemble size up to size 4. At ensemble size 5, the accuracy improvement for all three examined ensembling approaches starts to decrease. As can be observed from Table 2, an ensemble of size 5 always contains at least one of the CLIP models and a majority of less accurate models trained from scratch. Therefore there is a minority of more accurate models and a majority of less accurate ones. We hypothesize, that this fact could be causing the observed changes in accuracy improvement at ensemble size 5.

To better examine this situation, we display a plot evaluating only those ensembles which **do** contain the model clip_ViT-B-16 and **do not** contain the model clip_ViT-B-32. This way, there is always one model with higher accuracy (by almost 5%) than the remaining models with similar accuracies. This evaluation is displayed in Fig. 3. In this plot, the most accurate constituent is always clip_ViT-B-16. For these network combinations, we can observe more clearly the differences between the compared ensembling methods and less variance. The weaker ensemble members are evidently able to contribute some useful information to the more powerful CLIP model.

The baseline ensemble benefits most from adding just one weaker model to the more accurate CLIP. By adding more weak models, the accuracy improvement of the baseline decreases. This could perhaps be caused by the fact, that the CLIP model is pre-trained on a different dataset. The pre-training could lead the CLIP model to make different errors than the models trained from scratch. Adding a single weaker model could help to fix some of these errors. However, by adding several weaker models, the CLIP may not be able to tip the scales for the common errors made by these

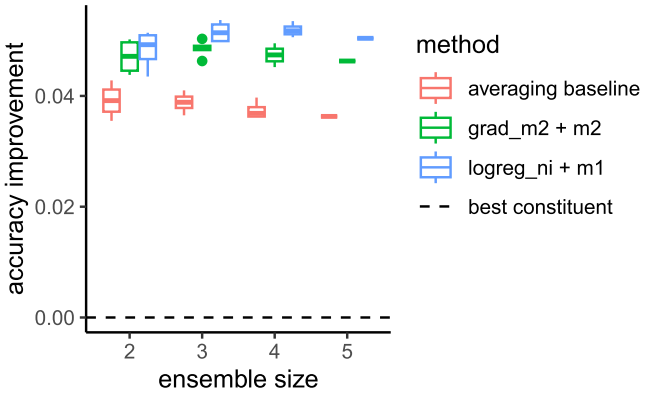


**FIGURE 4.** Improvement in the accuracy of a subset of ensembles. Analogous plot to Fig. 3. In this case, we exclude clip_ViT-B-16 and include clip_ViT-B-32 in every ensemble.

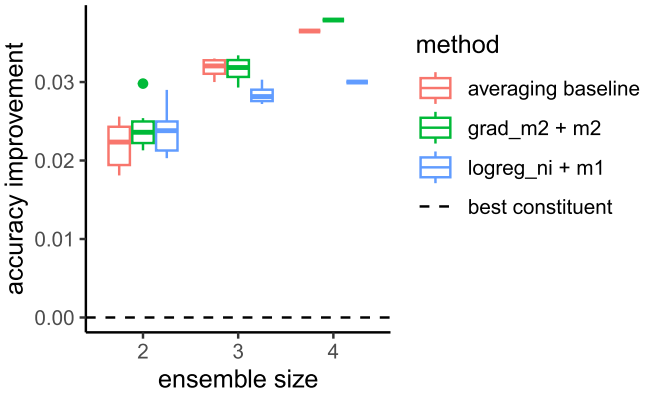


**FIGURE 5.** Improvement in the accuracy of a subset of ensembles. The subset contains only the ensembles built without the two CLIP models. The plot structure is the same as for Fig. 2.

more similar models trained from scratch. We should note, however, that the accuracy improvements are still positive even when the less accurate models outnumber the more accurate four to one.

Our proposed ensembling methods are able to benefit from two added weak models. By adding more weak models the performance is stagnating or decreasing. Ensembling method **logreg_ni + m1** is the most robust in this regard. The difference in the accuracy improvement between our methods and the baseline is increasing with the increasing ensemble size.

Similar, but less pronounced trends can be observed when we switch the place of the two CLIP models i.e. we include the less accurate clip_ViT-B-32 and exclude the more accurate clip_ViT-B-16. This situation is displayed in Fig. 4. If we compare figures 3 and 4 we can see, that the rate of decline in accuracy improvement with the increasing ensemble size is higher for Fig. 3 and lower for Fig. 4. The difference is visible mainly for the baseline and for the method **grad_m2 + m2**. The faster decline corresponds to the situation with a larger difference in the accuracy of combined models.

We also examine the complementary situation of combining only models with very similar accuracies. Ensemble improvements for combinations that do not contain either of the CLIP models are displayed in Fig. 5. In this plot, we can observe that the accuracy improvements for all three ensembling methods are increasing with raising ensemble size. For our first method **logreg_ni + m1**, the increase is slower than for the baseline. For our second method **grad_m2 + m2**, the increase is similar to that for the baseline. However, for ensemble sizes 2 and 4, the accuracy of our approach is slightly higher than that of the baseline.

Presented experiments suggest that our method can be well utilized in cases with significant differences between the accuracies of combined models, especially when the more accurate models are in the minority. It is also the case that these high-quality models are pre-trained on a different dataset and therefore can be expected to add diversity to the set of combined predictions. We show that these well-performing pre-trained models can be improved upon by ensembling with a small number of smaller, worse-performing models trained from scratch. When combining models of similar accuracies, and arguably less diversified predictions, the baseline method performs very well and our method does not provide a substantial improvement over the baseline ensemble.

**TABLE 4.** Accuracy of neural networks trained (or fine-tuned) on half of the CIFAR-100 dataset.

| neural network | accuracy |
|---|---|
| resnet34 | $0.6908 \pm 0.0035$ |
| xception | $0.7033 \pm 0.0033$ |
| densenet121 | $0.7064 \pm 0.0019$ |
| clip_ViT-B-32 | $\mathbf{0.7852} \pm 0.0020$ |

To verify these observations, we have performed another experiment on the CIFAR-100 dataset. We have created 10 random splits of the training set with each part containing half of the data while maintaining equal class frequencies. The first half becomes the training set for ensemble constituents. From the second half, we randomly pick 50 samples per class, obtaining a validation set of 5000 samples. On this validation set, we train the coefficients of our ensemble and also calibrate the networks for the baseline ensembling method. We fine-tune clip_ViT-B-32 and train three neural networks from scratch on each training split. These three networks differ in architecture from the networks used in the previous experiment. The accuracy of the networks is displayed in Table 4. We can again observe, that the CLIP model has a pronounced dominance over the other neural networks in terms of accuracy.

For building ensemble models, we always use only the four networks trained on the same split. Ensemble improvements for ensembles that incorporate the CLIP model are displayed in Fig. 6. We can observe similar trends that we have observed in figures 3 and 4. The decline in the performance of ensembling methods baseline and **grad_m2 + m2** with increasing ensemble size is even more pronounced than in figures 3 and 4. This observation is consistent with the hypothesis that the rate of decline in accuracy of the baseline method and

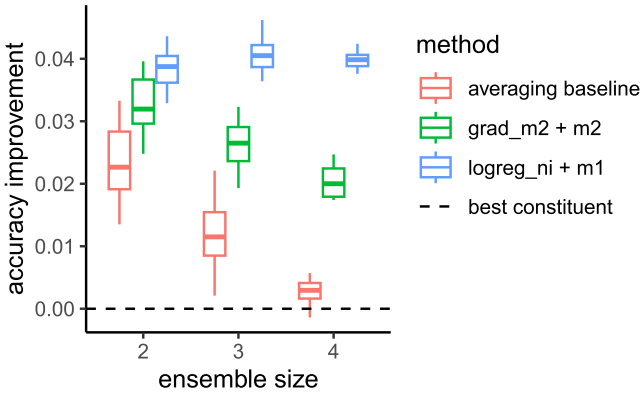


**FIGURE 6.** Improvement in the accuracy of a subset of ensembles built from the networks trained on half of CIFAR-100. The subset contains only ensembles including clip_ViT-B-32.

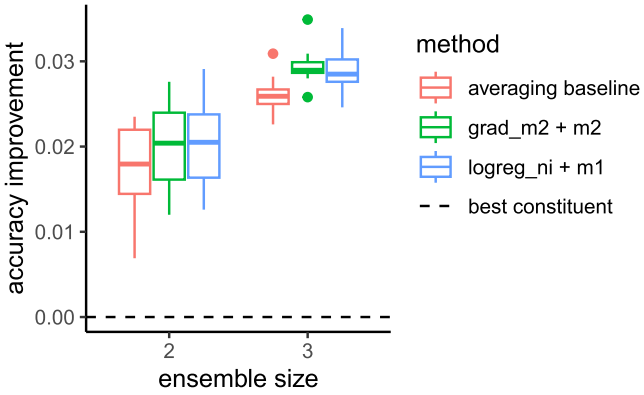


**FIGURE 7.** Improvement in the accuracy of a subset of ensembles built from networks trained on half of CIFAR-100. The subset excludes ensembles incorporating clip_ViT-B-32.

also of the method **grad_m2 + m2** when combining a single well-performing model with several weaker-performing models is proportional to the difference in combined models performance.

Equivalently to the previous experiment, we also evaluate combinations of the three similarly performing networks from Table 4. Results are displayed in Fig. 7. Similarly to the evaluation in Fig. 5, here we can also observe an increase in the accuracy improvement with increasing ensemble size for all three displayed ensembling methods. The difference between our two methods is smaller than in Fig. 5 and both our methods perform slightly better than the baseline. This could hint at the possibility that our methods work better with not-so-well-trained models, but it would require further experiments to study this possibility.

### B. ImageNet

On the ImageNet1k dataset, we have used six neural networks pre-trained on the full ImageNet21k dataset. All these networks have different architectures. The majority of them comprise vision transformers. These architectures are: vision transformers B32, Ti16, S16, B16, and a

**TABLE 5.** Accuracy of neural networks fine-tuned on the ImageNet1k dataset.

| neural network | accuracy top1 | accuracy top5 |
|---|---|---|
| Ti16 | 0.6745 | 0.8919 |
| B32 | 0.7561 | 0.9332 |
| S16 | 0.7660 | 0.9426 |
| M_B16 | 0.7664 | 0.9399 |
| R26_S32 | 0.7848 | 0.9478 |
| B16 | **0.8014** | **0.9525** |

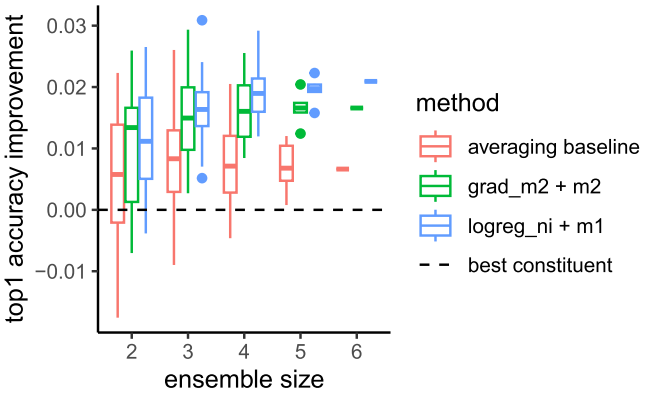


**FIGURE 8.** Improvement in the top1 accuracy of ensembles built from networks in Table 5.

combination of a convolutional neural network and a vision transformer R26_S32 [40]. The last architecture used is Mixer-MLP based on multi-layer perceptrons [41], we refer to it as M_B16. Pre-trained checkpoints were obtained from a GitHub repository [17]. We performed fine-tuning by training only the final layer.

On the ImageNet1k dataset, it is customary to evaluate also top5 accuracy. Top5 accuracy measures the proportion of the testing samples for which the correct class is among the five classes with the highest predicted probability. Accuracies top1 and top5 of these networks are reported in Table 5. Networks in the table are sorted according to top1 accuracy from the least accurate to the most accurate. The accuracies of these networks are more evenly spread than those of the networks in previous experiments. Only two networks have very similar accuracies.

Ensemble improvements in the top1 accuracy for all combinations of networks from Table 5 are displayed in Fig. 8. This figure represents a situation of combining ensemble members of varied accuracy. We can observe that the median performance of the baseline ensemble is mostly constant across different ensemble sizes. The accuracy improvement of our methods is slowly increasing with increasing ensemble size. We have observed similar behavior in Fig. 2 of all ensembles on the full CIFAR-100 dataset. However, here we don't observe the slight decrease in improvement for the largest ensemble sizes which is visible in Fig. 2. For ensemble sizes 2 up to 4, we have performed statistical tests comparing the proposed method and the baseline ensemble. P-values of these tests are displayed in Table 6. Evaluated at significance

**TABLE 6.** Results of statistical tests comparing PWE method and the baseline ensemble based on the improvement in top1 accuracy over the best constituent on the ImageNet1k dataset. The results are in favor of PWE in all the cases.

| configuration | ensemble size | p-value |
|---|---|---|
| grad_m2 + m2 | 2 | 0.0002 |
| | 3 | 0.0000 |
| | 4 | 0.0000 |
| logreg_ni + m1 | 2 | 0.0002 |
| | 3 | 0.0000 |
| | 4 | 0.0002 |

**TABLE 7.** Results of statistical tests comparing the proposed PWE method and the baseline ensemble based on the improvement in top5 accuracy over the best constituent on the ImageNet1k dataset. The results are in favor of PWE in all the cases.

| configuration | ensemble size | p-value |
|---|---|---|
| grad_m2 + m2 | 2 | 0.0002 |
| | 3 | 0.0001 |
| | 4 | 0.0000 |
| logreg_ni + m1 | 2 | 0.0003 |
| | 3 | 0.0000 |
| | 4 | 0.0002 |

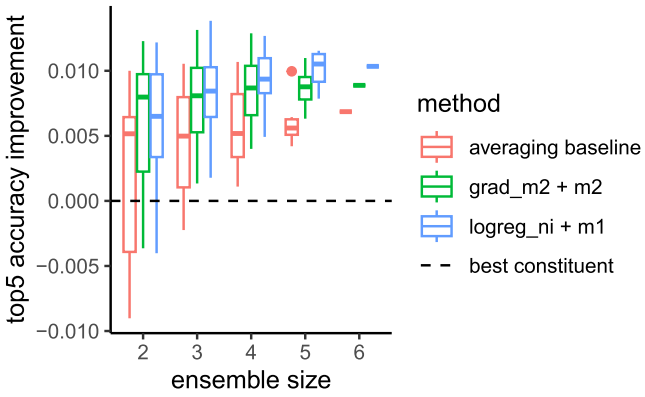


**FIGURE 9.** Improvement in the top5 accuracy of ensembles built from networks trained on ImageNet1k.

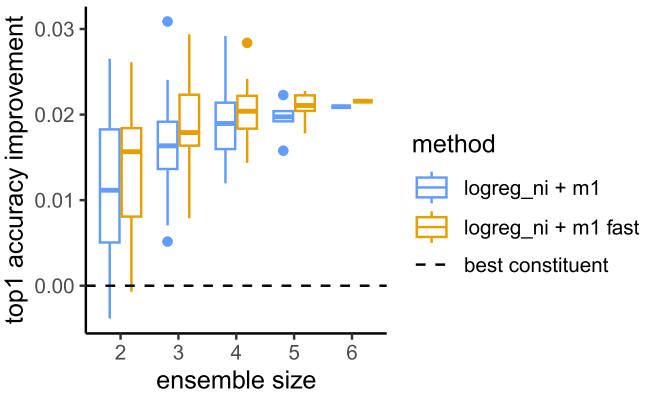


**FIGURE 10.** Comparison of the top1 accuracy improvement of the simplified inference method fast and of the standard inference. Ensembles are built from all combinations of networks from Table 5.

level 5%, all these tests show an advantage of PWE over the baseline ensemble. Ensemble sizes 5 and 6 do not have a sufficient number of samples to perform a statistical test. For the available data, the advantage of our method for these sizes is clear from Fig. 8.

Ensemble improvements in the top5 accuracy for all combinations of ImageNet1k networks are displayed in Fig. 9. The behavior of our methods for top5 accuracy and top1 accuracy is very similar as can be observed by comparing Fig. 9 with Fig. 8. Top5 accuracy improvement of the baseline method is slowly rising with increasing ensemble size. P-values of statistical tests for top5 accuracy are displayed in Table 7. At the significance level 5%, all these tests show an advantage of the proposed method. For available data, the advantage of the proposed method for ensemble sizes 5 and 6 is apparent from Fig. 9. Improvements obtained by ensembles for top5 accuracy are smaller than those for top1 accuracy. But that is to be expected as the top5 accuracies of the combined networks are much higher than the top1 accuracies, which leaves less space for improvement.

On ImageNet1k we do not evaluate specific subsets of ensembles as we did on CIFAR-100. Here the performances of combined neural networks are more varied and there are not more than two similarly performing networks.

In subsection II-B, we have proposed a modification to our method's inference process with lower computation expenses. The modification is controlled by a hyperparameter *topl*. We have evaluated different values of this hyperparameter on a hold-out set and found the value 5 to work well for the method **logreg_ni + m1**. For an ensemble of size 4, the unmodified prediction process on the full validation set of ImageNet1k of 50000 samples takes, on average, 163 seconds. In contrast, for a simplified inference process with topl 5, the inference takes, on average, 1.24 seconds. These prediction times cover only the ensembling process excluding the inference time of ensemble members. Using the same GPU accelerator, the inference for the validation set of ImageNet1k by the largest used network, B16, takes, on average, 609 seconds. Considering the use of several networks, we can see that the ensembling process takes only a small part of the entire ensemble inference time.

We have examined whether the substantial improvement in the computation time provided by lowered *topl* value has any detrimental effects on the prediction quality. Comparison of the top1 accuracy improvement obtained with the standard inference and with the reduced value *topl* of 5 (denoted as fast) is displayed in Fig. 10. We can see that the inference modification *fast* does not harm the top1 accuracy improvement of the standard inference. Median top1 accuracy improvement of the inference method *fast* is higher than that of the standard method for all evaluated ensemble sizes.

We also studied the effect of the inference modification on the top5 accuracy improvement. Comparison is displayed in Fig. 11. Here again, we can observe higher median top5 accuracy improvement of the *fast* inference method.

For the *fast* inference method to work well, the correct class has to be in the set of the picked classes. In case this is true, the

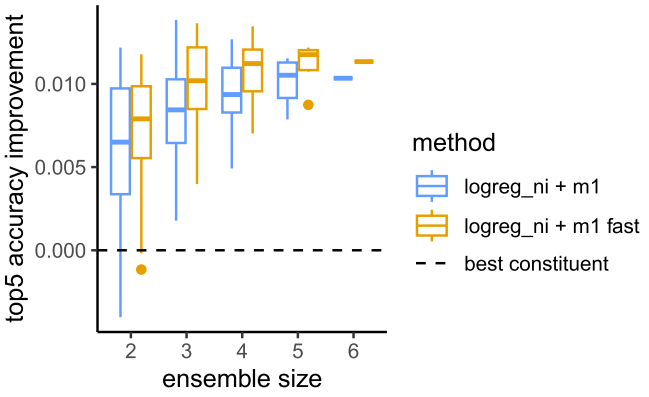


**FIGURE 11. Comparison of the top5 accuracy improvement of the simplified inference method fast and of the standard inference. Ensembles are built from all combinations of networks from Table 5.**

fast inference reduces the amount of noise from the irrelevant classes and allows the ensemble to perform better. We think that this is the cause of the improvements we are seeing in the presented results.

We emphasize, that the inference modification does not affect the ensemble training. A single set of weights, trained in a standard way, can be used both for the standard inference and for the simplified inference.

We can conclude, that for our ensembling method **logreg_ni + m1**, the fast inference modification does not harm the quality of the ensemble output, while it substantially reduces the required computation.

The training process of our method also has a quadratic complexity in the number of classes. However, we show that with parallel implementation and the use of a GPU accelerator, our method has good utility even on the ImageNet1k dataset with 1000 classes. With the use of NVIDIA GeForce RTX 2080 Ti, the training time of gradient-based method **grad_m2** is about 30 minutes for an ensemble with five constituents. For the method **logreg_ni**, based on logistic regression, the training time for the same ensemble is only 14 seconds. The prediction time for 50000 samples of the ImageNet1k validation set is around 160 seconds. However, we have shown that, with the simplified inference process, this time can be reduced to around 1.3 seconds. For configuration **logreg_ni + m1**, this modification didn't cause any drop in the accuracy of prediction. Both the training and the prediction time of our method present only a small fraction of the combined training (or fine-tuning) and prediction times of ensemble constituents.

## V. DISCUSSION

We have proposed a novel pairwise weighted ensembling approach. This approach is a general template and provides modularity in the way in which the weights are trained and in the choice of the pairwise coupling method. We have suggested two configurations of our approach and evaluated them on the CIFAR-100 and ImageNet1k datasets. Alongside our method, we have also tested a popular averaging ensemble as a baseline. Apart from experimental testing, we also performed a theoretical analysis of the proposed method in a model scenario.

Performed statistical tests showed a statistically significant advantage of the proposed method over the averaging baseline in all examined cases. The proposed method achieved the most pronounced improvements over the baseline in the cases with high differences in the prediction quality of combined models. This was shown on the CIFAR-100 dataset in cases of combining several similarly performing weaker models with a single better-performing model. In this setting, the configuration **logreg_ni + m1** of our method displayed especially favorable behavior. On the ImageNet1k dataset, we have examined a case of combining several classifiers with varied performance. Our method provided a clear advantage over the baseline also in this case.

All classifiers chosen as ensemble members in our experiments are modern deep neural networks. We used mainly convolutional neural networks and vision transformers. In the case of vision transformers, we have utilized models pre-trained on large amounts of data and fine-tuned them for use in our experiments. These large models provide very high prediction quality. However, further improving the prediction quality by simply enlarging these models to even larger sizes poses problems with the training process. Non-trivial changes to their architecture may be needed, and the larger they are, the more data-hungry the training process is [42]. The proposed ensembling approach was able to provide consistent improvements in the prediction accuracy of these large models without any changes to their architecture or training process. This ability makes the proposed approach highly practical in cases where a few extra percent of accuracy improvement provides a high value.

We revealed some limitations of our method on the CIFAR-100 dataset. When combining several similarly performing models, our method performed similarly, or only slightly better than the baseline. For some ensemble sizes, one of our configurations performed slightly worse than the baseline. However, it still provided an improvement over the most accurate ensemble constituent. Our second tested configuration **grad_m2 + m2** proved to perform better in these cases. In the case of similarly performing networks trained only on half of the CIFAR-100 dataset, the improvement of our method over the baseline was more noticeable.

Another possible limitation of our method is its quadratic complexity in the number of combined classes. However, with our parallel implementation, the training time, even for the ImageNet1k dataset with 1000 classes, stayed at the practical levels. We managed to alleviate the effects of this complexity on the inference time by creating a simplified inference process. We have shown in the performed experiments that this simplified inference process can significantly reduce the inference time without impairing the accuracy of prediction. For our more successful configuration **logreg_ni + m1** the training and inference on 50000 samples of ImageNet1k took 14 and 1.3 seconds, respectively.

To obtain the best results with the proposed method, we recommend keeping a separate hold-out set during the ensemble members' training. This hold-out set should then be used to train the ensemble weights. Methods that we have used for training the ensemble weights use regularization. Regularization strength is controlled by a hyperparameter that needs to be tuned. This tuning can be performed by a $k$-fold cross-validation on the hold-out set or by keeping a separate set of validation data. The extra data needed for ensemble training and hyperparameter tuning could pose a limitation for application on problems with only very limited training data available. In such cases, we would recommend the use of pre-trained ensemble members due to their lower requirements on the amount of available training data.

We have also evaluated the averaging ensemble used as a baseline. We can conclude that the averaging ensemble proved to be a simple yet powerful tool as it provided good results in all examined situations. It worked surprisingly well even in the case of combining several weaker-performing models with a single better-performing model. Even in this case, the baseline managed to provide consistent improvements over the predictions of the better-performing model. However, the improvements rapidly decreased with an increasing number of weaker constituent models.

Our proposed approach is mainly a general template and we report only on two specific configurations. We have examined several other configurations incorporating linear discriminant analysis for training the combining weights. We have also tested two more pairwise coupling methods [43], [44]. These configurations, however, had less stable performance. Other different configurations of pairwise weighted ensembles using generalized linear models as weights training methods provide a venue for further research.

## VI. CONCLUSION

Our work has wide applicability beyond the visual recognition tasks presented here.

In our work, we varied the architecture of deep neural networks to obtain a diverse set of classifiers. However, our method is equally applicable if bootstrap aggregation (bagging) is used to construct ensemble members. Alternatively, time series classification methods are known for ensembling diverse classifiers [45], [46], [47], both using deep networks as well as more traditional machine-learning models, indicating that our method can be advantageously applied in this area.

Using the proposed ensembling approach, it is possible to incorporate into an ensemble the models that are trained on the sub-problems of the problem at hand.

One example is to utilize specialized classifiers trained on subsets of classes that are especially hard to distinguish. For instance, one could start with a visual transformer classifier pre-trained on a very large dataset. Based on the confusion matrix on an application-specific dataset, one could identify subsets of hard-to-separate classes. For these subsets, one could train separate deep convolutional networks and combine the results using our method.

Another example is an application for federated learning [48], [49], [50], [51]. One may cover the set of classes by overlapping subsets and then train a classifier for each of the subsets concurrently on different computational nodes. Afterwards, a multi-class classifier for the complete set of classes may be constructed using our method.

Multimodal classification is another possible area of application. The theoretical analysis of the proposed method provided in section III examines a situation where each of the combined classifiers is provided only a subset of features. In this case, the analysis highlights the good properties of our method. The common practical setting for combining classifiers working on disjoint feature sets occurs when processing audiovisual data. The McGurk effect suggests that for some sublexical units, the human brain outweighs auditory perception compared to visual perception [52], similarly as we do in our method. Combining separate classifiers trained on different modalities is common also for other types of multimodal data [53]. We suggest applying our method in such tasks as a prospective future research direction.

Finally, we provided a theoretical justification for geometrically averaging predictions when features are no correlated as in the theoretical scenario in Section III. This suggests that geometric averaging could outperform arithmetic averaging of predictions in some applications.

## REFERENCES


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[2] R. M. Bell, Y. Koren, and C. Volinsky, "The BellKor solution to the Netflix prize," Netflix, Los Gatos, CA, USA, KorBell Team's Rep., 2007.

[3] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems* (Lecture Notes in Computer Science), vol. 1857, J. Kittler and F. Roli, Eds. Cagliari, Italy: Univ. of Cagliari, Jun. 2000, pp. 1–15.

[4] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, pp. 105–151, Oct. 2022.

[5] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, "The power of ensembles for active learning in image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9368–9377.

[6] C. Ju, A. Bibaut, and M. van der Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," *J. Appl. Statist.*, vol. 45, no. 15, pp. 2800–2818, Nov. 2018.

[7] R. Bravin, L. Nanni, A. Loreggia, S. Brahnam, and M. Paci, "Varied image data augmentation methods for building ensemble," *IEEE Access*, vol. 11, pp. 8810–8823, 2023.

[8] L. Liu, W. Wei, K.-H. Chow, M. Loper, E. Gursoy, S. Truex, and Y. Wu, "Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness," in *Proc. IEEE 16th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 274–282.

[9] L. Nanni, G. Maguolo, S. Brahnam, and M. Paci, "An ensemble of convolutional neural networks for audio classification," *Appl. Sci.*, vol. 11, no. 13, p. 5796, Jun. 2021.

[10] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep neural network ensembles for time series classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[11] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, and E. Xun, "Ensemble of deep neural networks with probability-based fusion for facial expression recognition," *Cogn. Comput.*, vol. 9, no. 5, pp. 597–610, Oct. 2017.

[12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2017, pp. 6405–6416.

[13] L. Nanni, L. Trambaiollo, S. Brahnam, X. Guo, and C. Woolsey, "Ensemble of networks for multilabel classification," *Signals*, vol. 3, no. 4, pp. 911–931, Dec. 2022.

[14] A. Mosca and G. D. Magoulas, "Deep incremental boosting," in *Proc. Global Conf. Artif. Intell.*, vol. 41, 2016, pp. 293–302.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[16] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. (2021). *Big Transfer (Bit): General Visual Representation Learning*. [Online]. Available: https://github.com/google-research/big_transfer

[17] A. Steiner. (2023). *Vision Transformer and MLP-Mixer Architectures*. [Online]. Available: https://github.com/google-research/vision_transformer

[18] X. Frazão and L. A. Alexandre, "Weighted convolutional neural network ensemble," in *Proc. Iberoamer. Congr. Pattern Recognit.*, 2014, pp. 674–681.

[19] A. Mosca and G. Magoulas, "Customised ensemble methodologies for deep learning: Boosted residual networks and related approaches," *Neural Comput. Appl.*, vol. 31, pp. 1713–1731, Jun. 2019.

[20] W. Zhang, J. Jiang, Y. Shao, and B. Cui, "Snapshot boosting: A fast ensemble framework for deep neural networks," *Sci. China Inf. Sci.*, vol. 63, no. 1, pp. 1–12, Jan. 2020.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[22] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Statist.*, vol. 26, no. 2, pp. 451–471, Apr. 1998.

[23] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, Dec. 2004.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011.

[25] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Springer, 2009.

[26] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 1321–1330.

[27] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.

[28] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, Apr. 2009.

[29] S. S. Mangiafico, "Summary and analysis of extension program evaluation in R," Rutgers Cooperat. Extension, New Brunswick, NJ, USA, Tech. Rep., 2016.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2016, pp. 770–778.

[32] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 646–661.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 2261–2269.

[35] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 1800–1807.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[37] Weiaicunzai. (2022). *PyTorch-CIFAR100*. [Online]. Available: https://github.com/weiaicunzai/pytorch-cifar100

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *Proc. Mach. Learn. Res.*, vol. 139, pp. 8748–8763, Jul. 2021.

[39] J. W. Kim, S. Castro, T. J. Huei, K. Costa, H. Wang, I.-H. Yi, R. Beaumont, S. Berns, and J. Sutor. (2020). *Clip*. [Online]. Available: https://github.com/openai/CLIP

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth $16 \times 16$ words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Austria, May 2021, pp. 1–22.

[41] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-mixer: An all-MLP architecture for vision," in *Advances in Neural Information Processing Systems*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds. Red Hook, NY, USA: Curran Associates, 2021, pp. 24261–24272.

[42] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems*, vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015.

[43] O. Such, S. Benus, and A. Tinajová, "A new method to combine probability estimates from pairwise binary classifiers," in *Proc. Conf. Theory Pract. Inf. Technol.*, 2015, pp. 194–199.

[44] O. Šuch and S. Barreda, "Bayes covariant multi-class classification," *Pattern Recognit. Lett.*, vol. 84, pp. 99–106, Dec. 2016.

[45] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 5, pp. 1–35, 2018.

[46] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Müller, and F. Petitjean, "InceptionTime: Finding AlexNet for time series classification," *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1936–1962, Nov. 2020.

[47] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "HIVE-COTE 2.0: A new meta ensemble for time series classification," *Mach. Learn.*, vol. 110, nos. 11–12, pp. 3211–3243, Dec. 2021.

[48] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[49] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[50] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.

[51] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.

[52] A. D. Mitchel, M. H. Christiansen, and D. J. Weiss, "Multimodal integration in statistical learning: Evidence from the McGurk illusion," *Frontiers Psychol.*, vol. 5, p. 407, May 2014.

[53] E. Alpaydin, *Classifying Multimodal Data*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 49–69.

**RENÉ FABRICIUS** received the B.S. and M.S. degrees in informatics from the University of Žilina, Žilina, Slovakia, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in applied informatics.

From 2021 to 2023, he was a Research Assistant with the Research Centre, University of Žilina. His research interests include computer vision, classification ensembles, and optimization using evolutionary metaheuristics.

**ONDREJ ŠUCH** received the M.Sc. degree in mathematics from Queen's University, Kingston, ON, Canada, in 1993, and the Ph.D. degree in mathematics from Princeton University, Princeton, NJ, USA, in 1997.

He was with Microsoft Corporation, as a Software Engineer, from 1997 to 2002. Since 2002, he has been moved to academia, doing research with the Mathematical Institute of Slovak Academy of Sciences. He is teaching computer science with Univerzita Mateja Bela v Banskej Bystrici and with the University of Žilina. He has supervised two Ph.D. students while teaching in Žilina, with topics in cybersecurity and computer vision. Most recently he has been studying ensembling methods with special focus on pairwise coupling methods. His research interests include graph theory, number theory, neuromorphic computing, and artificial intelligence.

**PETER TARÁBEK** received the Ph.D. degree from the University of Žilina, in 2009.

From 2021 to 2023, he was with the Research Centre, University of Žilina, where he is currently an Assistant Professor with the Department of Mathematical Methods and Operations Research. He has participated in several research and applied projects, in which he developed solutions in areas of computer vision, intelligent transportation systems, and Industry 4.0. His research activities fall in the areas of artificial intelligence, deep machine learning, and computer vision. His research interests include computer vision applied to medical image analysis, learning from small amounts of data, using ensemble methods in classification, and explainability in deep machine learning.