

Black-Box Test-Time Ensemble

Abstract—Privacy considerations have become increasingly critical in the deployment of modern machine learning models. To protect sensitive data and reduce storage or transmission costs, many service providers offer trained models via APIs, effectively creating privacy-preserving black-box models. However, evaluating the performance of such models remains to be a significant challenge, especially for downstream tasks lacking labeled data. This paper proposes an unsupervised combination method for black-box test-time ensemble. By utilizing only the models’ predictions on unlabeled test data, the proposed approach estimates the reliability of individual base classifiers and constructs a weighted ensemble that favors more accurate ones. Our approach is compatible with both traditional machine learning classifiers and modern large language models, and accommodates a wide range of scenarios, including binary and multi-class classification, hard and soft outputs, and both offline and online settings. Extensive experiments on 13 real-world text, image, and time series datasets verified the effectiveness and flexibility of the approach, consistently outperforming majority voting and other combination approaches. Notably, the proposed approach is hyperparameter-free and computationally efficient, rendering it well-suited for applications that require online real-time inference.

Index Terms—Ensemble learning, large language model, privacy protection, transfer learning, unsupervised learning

I. INTRODUCTION

MACHINE learning has achieved remarkable progress by leveraging increasing amounts of data and larger model architectures. Privacy has emerged as a growing concern, as the machine learning pipeline is susceptible to various security and information leakage attacks [1]. These concerns are further amplified by rising demands for privacy protection, driven by regulations such as the European General Data Protection Regulation, the American Data Privacy and Protection Act, and China’s Personal Information Protection Law, as well as heightened awareness among users.

Black-box models provide a straightforward yet effective means to achieve privacy-preserving machine learning. In this paradigm, users submit inputs and receive outputs from a model without access to its internal parameters or training data. Beyond privacy, black-box deployment also addresses efficiency concerns due to the increasing costs of storing and transmitting large-scale models and data [2]. Fig. 1 highlights the growing importance of black-box models and the need for robust evaluation or ensemble techniques. While such models still remain vulnerable to certain attacks, the lack of access to internal parameters typically limits the efficacy of these threats.

Despite these advantages, accurately assessing the downstream performance of black-box models on specific unseen test data remains a significant challenge [3]. In the absence of annotations for direct comparison of performance, unsupervised evaluation becomes essential for reliably estimating the performance of multiple accessible black-box models.

Unsupervised ensemble learning offers a viable strategy for performance evaluation or aggregation, solely based on multiple pre-trained models’ predictions on the unlabeled test data. In particular, combination methods [4] aim to optimize aggregation weights over base classifiers, or maximize the likelihood over observed predictions. Compared to single models, combination methods provide improved statistical stability, computational robustness, and representational diversity [5]. However, most existing combination approaches rely on transductive analysis over the entire unlabeled test set and are often tailored for offline evaluation in crowdsourcing applications [6]. These methods typically require iterative optimization, and some have also largely been limited to binary classification tasks, thereby limiting their applicability in real-world settings.

This paper addresses the challenging scenario where multiple black-box models make inferences to test samples, without disclosing model parameters or training samples. We propose a test-time ensemble method based on the Spectral Meta-Learner (SML) [7], an unsupervised spectral ensemble learning approach. The SML approach estimates model performance using only predictions on unlabeled test data, exploiting inter-model prediction correlations. Classifiers estimated to be more reliable are assigned higher weights, resulting in a weighted ensemble whose weights are approximately proportional to the inferred performance scores in multi-class settings.

Our main contributions are summarized as follows:

- 1) Extension of the SML [7] from binary to multi-class classification, enabling unsupervised performance estimation without access to labels.
- 2) Proposal of a black-box test-time unsupervised ensemble learning approach applicable to a wide range of scenarios, including hard/soft predictions, online/offline evaluation, and binary/multi-class tasks.
- 3) Extensive experiments were conducted on 13 real-world datasets of text, image, and time series classification tasks, demonstrating the versatility and effectiveness of our proposed approach.

The proposed approach enables unsupervised classifier ranking, pruning, and ensemble, making it a promising component for modern privacy-preserving machine learning systems.

II. RELATED WORK

A. Unsupervised Ensemble Learning

While most ensemble learning methods are supervised [8], [9], unsupervised ensemble learning has received comparatively less attention. The unsupervised combination methods typically involve two components: unlabeled test data and predictions from multiple pre-trained models. Within this context, crowdsourcing [10] aims to infer worker reliability, whereas model combination [4] focuses on estimating ground-truth labels, which are interrelated objectives.

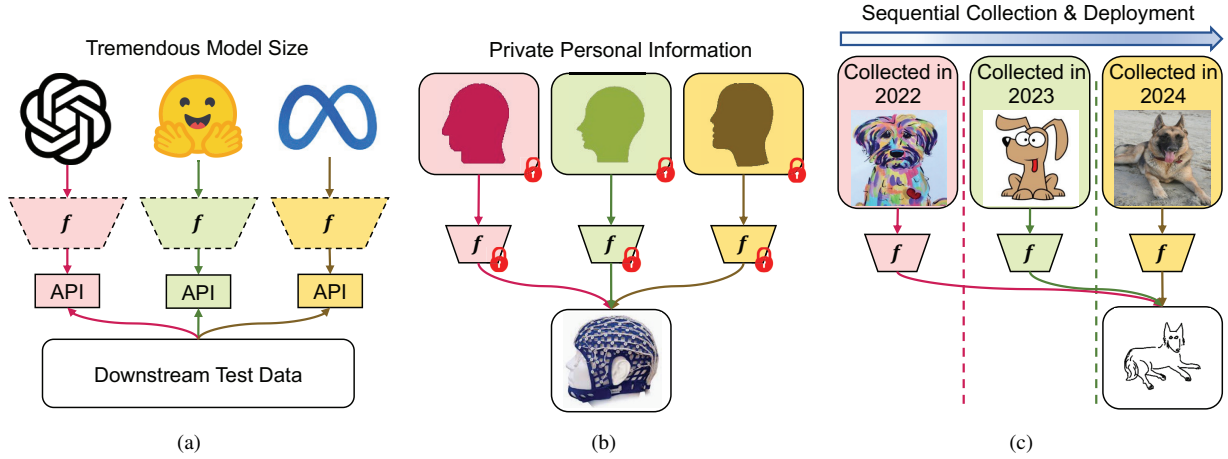


Fig. 1. Scenarios where black-box models are deployed, and performance evaluation becomes essential. (a) Models with tremendous amount of parameters are only accessible via APIs offered by service providers; (b) private information in data restricts transfer of data or even model parameters; and (c) data are collected intermittently from multiple domains, and each model is encapsulated to preserve domain-specific knowledge and reduce retraining overhead.

The simplest combination strategy is majority voting, which treats all models equally and has a known theoretical performance bound [11]. To jointly estimate model reliability and latent labels using only unlabeled data, many unsupervised ensemble methods employ maximum likelihood estimation, typically implemented via the Expectation-Maximization (EM) algorithm [10].

One representative approach is the Dawid-Skene model [12], which assumes conditional independence among classifiers and models two components:

- 1) The confusion matrix of each classifier and the class prior distribution.
- 2) The latent true labels for all test samples.

The algorithm is often initialized with majority voting. EM then iteratively maximizes the likelihood function by alternating between estimating one component while keeping the other fixed, continuing until convergence.

However, the EM algorithm has several limitations. Due to the non-convexity of the likelihood function, it may converge to suboptimal local maxima [7], [13]. The time complexity is $\mathcal{O}(nMK)$, where n is the number of test samples, M is the number of classifiers, and K is the number of classes. In streaming applications, where test samples arrive sequentially and require immediate inference, retraining the EM model per instance is computationally infeasible. As a result, EM-based methods are more suitable for crowdsourcing scenarios than for real-time ensemble deployment. Developing efficient alternatives for practical deployment remains to be a challenge.

B. The Spectral Meta-Learner

The Spectral Meta-Learner (SML) [7] takes a different approach for combination, through directly estimating classifier reliability instead of modeling confusion matrices. SML approximates maximum likelihood estimation in binary classification using spectral (i.e., eigenvalues and eigenvectors) decomposition. Unlike EM-based methods, SML offers a near closed-form solution, eliminating the need for iterative

optimization. It constructs a weighted ensemble based on estimated balanced classification accuracy (BCA) scores, assigning higher weights to more reliable classifiers.

Subsequent extensions have adapted SML to dependent classifiers [14]. Several alternatives have been proposed, but they often introduce additional limitations such as iterative training [13], [15], extra parameters [16], neural network overhead [16], or reliance on ranking approximations [17]. Although some work has briefly explored multi-class extensions [18], [19], evaluations remain limited to synthetic or small-scale datasets.

C. Transfer Learning

Transfer learning [20] addresses the scenario where the target (test data) domain probability distribution is different from the source (training data) domain. A key component involves estimating the transferability of source models [21], which is similar to the objective of combination methods that assess model reliability.

Transferability estimation often relies on sample-wise correlations, using either uncertainty or similarity metrics. Uncertainty-based approaches analyze prediction confidence, under the assumption that in-distribution models yield high-confidence predictions, while out-of-distribution models [22] produce less confident or randomized outputs. Similarity metrics, such as cosine similarity, are computed using latent representations (e.g., outputs from layers preceding the classifier). However, such techniques typically require access to model parameters and cannot be applied to black-box models.

Online test-time adaptation methods [23] aim to adapt pre-trained models to a distribution-shifted target domain. These approaches generally target a single model and often involve backpropagation, incorporate generative modeling, or at least update model normalization statistics, which are incompatible with black-box models under privacy restrictions.

III. BLACK-BOX TEST-TIME ENSEMBLE

A core challenge in combination methods is to assess the reliability of each classifier within the ensemble. The Spectral Meta-Learner (SML) [7] addresses this issue by estimating a performance score for each base classifier through the utilization of correlations across them. By assigning weights proportional to these scores for aggregating the base classifiers, a weighted combination can be built.

We extend SML from binary to multi-class classification by introducing the SML One-Vs-Rest (SML-OVR) approach, illustrated in Fig.2.

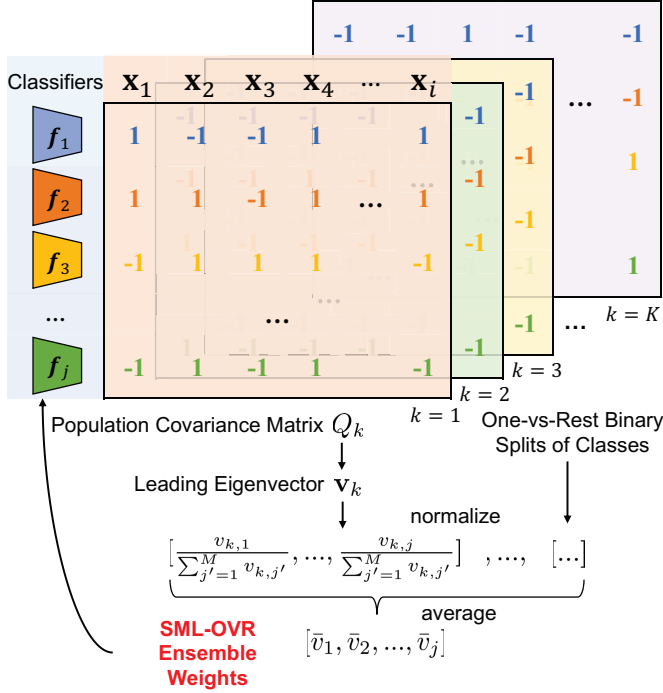


Fig. 2. Illustration of the black-box test-time ensemble approach SML-OVR. Ensemble weights are computed based on the covariance of classifiers' predictions on the accessible test set, assigning higher weights to more reliable models. For multi-class classification, weights are normalized and averaged over all corresponding binary one-vs-rest subtasks to build a convex combination.

A. Preliminaries under Binary Case

Most of the preliminaries follow the original SML publication [7].

Consider a binary classification task with $K = 2$ classes. Let $\{\mathbf{x}_i\}_{i=1}^n$ be n test samples evaluated by $M \geq 3$ classifiers $\{f_j\}_{j=1}^M$, where each classifier produces a binary prediction $f_j(\mathbf{x}_i) \in \{-1, 1\}$. The true labels $\{y_i\}_{i=1}^n$ are assumed to be unavailable. Only the predictions $\{f_j(\mathbf{x}_i)\}_{i=1, j=1}^{n, M}$ of the classifiers are utilized in unsupervised ensemble learning.

Define Q as the $M \times M$ population covariance matrix of the base classifiers. The entries of Q are:

$$q_{ij} = \mathbb{E}[(f_i(\mathbf{x}) - \mu_i)(f_j(\mathbf{x}) - \mu_j)], \quad (1)$$

where \mathbb{E} denotes expectation with respect to the density $p(x, y)$, and $\mu_i = \mathbb{E}[f_i(\mathbf{x})]$.

Let ψ and η denote the sensitivity and specificity of a classifier:

$$\psi = P[f(\mathbf{x}) = Y | Y = 1], \quad (2)$$

$$\eta = P[f(\mathbf{x}) = Y | Y = -1]. \quad (3)$$

The balanced classification accuracy (BCA), denoted by π , is defined as the mean of sensitivity and specificity:

$$\pi = \frac{\psi + \eta}{2}. \quad (4)$$

Under *Assumption 1*, *Lemma 1* and *Lemma 2* can be proved. Notably, weights proportional to the true BCA scores of the base classifiers can be inferred, without access to groundtruth labels.

Assumption 1. 1) *Test data i.i.d.* The test samples are independently and identically distributed:

$$\{\mathbf{x}_i\}_{i=1}^n \stackrel{i.i.d.}{\sim} \mathcal{D}. \quad (5)$$

for some unknown data distribution \mathcal{D} .

2) **Conditional independence.** Classifier predictions are conditionally independent given the true label y_i :

$$P(f_1(\mathbf{x}_i)f_2(\mathbf{x}_i) \dots f_M(\mathbf{x}_i) | y_i) = \prod_{j=1}^M P(f_j(\mathbf{x}_i) | y_i), \forall i. \quad (6)$$

3) **Majority competence.** A majority of classifiers outperform random guessing, i.e., chance-level prediction, such that aggregation provides positively informative predictions rather than misleading ones.

$$\sum_{j=1}^M \mathbb{I}\left(\pi_j > \frac{1}{K}\right) > \frac{M}{2}, \quad (7)$$

Lemma 1. [7] The entries $q_{jj'}$ of Q are equal to

$$q_{jj'} = \begin{cases} 1 - \mu_j^2, & j = j' \\ (2\pi_j - 1)(2\pi_{j'} - 1)(1 - b^2), & \text{otherwise} \end{cases} \quad (8)$$

where $b = P[Y = 1] - P[Y = -1]$ denotes the difference in class prior, i.e., class imbalance.

The off-diagonal entries are identical to those of a rank-one matrix $R = \lambda \mathbf{v} \mathbf{v}^\top$ with unit-norm leading eigenvector $\mathbf{v} \in \mathbb{R}^M$ and eigenvalue λ that

$$\lambda = (1 - b^2) \cdot \sum_{j=1}^M (2\pi_j - 1)^2. \quad (9)$$

Lemma 2. [7] The entries of \mathbf{v} are proportional to the BCAs of the M classifiers, up to a sign ambiguity:

$$v_j \propto (2\pi_j - 1). \quad (10)$$

Beyond ranking, predictions of the base classifiers can be further weighted-combined using the entries of \mathbf{v} :

$$\hat{y}_i^{\text{SML}} = \text{sign} \left(\sum_{j=1}^M f_j(\mathbf{x}_i) \cdot \hat{v}_j \right), \quad (11)$$

where \hat{v}_j is the estimate of the entry v_j of \mathbf{v} . Various estimation approaches, including (weighted) linear systems, semi-definite programming, and direct spectral decomposition, are discussed in [7].

Intuitively, SML assigns higher weights to classifiers with superior estimated BCA, making it more accurate than unweighted methods such as majority voting.

B. Extension of SML to Multi-Class Classification

We extend SML to multi-class classification with $K > 2$ via a one-vs-rest strategy. For each class $k \in \{1, \dots, K\}$, define a binary subtask where class k is positive ($\mathcal{A}_k = \{k\}$) and all other classes are negative ($\mathcal{Y} \setminus \mathcal{A}_k = \{1, 2, \dots, k-1, k+1, \dots, K\}$). Given the one-hot prediction vector $f_j(\mathbf{x}_i)$ from the j -th classifier on the i -th test sample, we derive binary predictions $\hat{f}_{j,k}(\mathbf{x}_i) \in \{-1, 1\}$ for each one-vs-rest subtask.

For each class-wise binary subtask, Lemma 2 still applies, allowing the principal eigenvector $\{\mathbf{v}_k\}_{k=1}^K$ to be computed. However, in multi-class settings, the concept of specificity becomes ill-defined. Specifically, the term $\eta = P[f(\mathbf{x}) = Y | Y \in (\mathcal{Y} \setminus \mathcal{A}_k)]$ aggregates both correct and incorrect predictions across all negative classes, introducing ambiguity in estimating performance via eigendecomposition.

Jaffe *et al.* [19] formalized this limitation by proving that confusion matrices in the multi-class setting cannot be uniquely recovered from predictions alone. Since BCA is defined as the average sensitivity across all classes, per-class BCA estimates become unreliable without labeled data.

To address this limitation, we propose an empirical extension that avoids recovering the full multi-class confusion matrices. Instead, we adhere to the core principle of combination methods: estimating the reliability of individual base classifiers.

Specifically, reliability scores for each classifier can be obtained by averaging the normalized weight vector:

$$\bar{v}_j = \frac{1}{K} \sum_{k=1}^K \frac{v_{k,j}}{\sum_{j'=1}^M v_{k,j'}}, \quad (12)$$

where $v_{k,j}$ denotes the j -th entry of \mathbf{v}_k , the principal eigenvector for the k -th binary subtask. The per-class normalization ensures each class contributes on an equal scale.

These reliability scores can then be used to construct a convex combination of classifier outputs for final ensemble prediction:

$$\hat{y}_i^{\text{SML-OVR}} = \arg \max_k \left(\sum_{j=1}^M \hat{f}_{j,k}(\mathbf{x}_i) \cdot \bar{v}_j \right). \quad (13)$$

Unlike the binary case in Eq. (11), where binary decisions are based on the sign of a scalar, multi-class decisions are made by selecting the class with the maximum aggregated score.

Note that in each one-vs-rest subtask, the resulting weight is approximately proportional to the sum $\psi + \eta$. While the term η corresponding to multi-class specificity is ill-defined and introduces ambiguity in performance estimation, this effect diminishes as classifiers approach perfect accuracy, where the off-diagonal entries in the confusion matrix tend toward zero.

In such cases, the estimated weight remains a valid indicator of classifier performance.

Overall, this unsupervised approach offers a hyperparameter-free and computationally efficient mechanism for test-time ensemble in the multi-class setting, despite the lack of strong theoretical guarantees.

C. Online Test-Time Update of Combination Weights

In many real-world settings, the test set is not fully accessible in advance. Instead, online test samples arrive sequentially as a data stream. The proposed SML-OVR approach naturally supports online test-time evaluation by incrementally updating the ensemble weights as new test data arrive.

More specifically, the online update of SML-OVR consists of two main steps:

- 1) Covariance Matrix Update: Computing the full covariance matrix Q for n test samples takes time complexity $\mathcal{O}(nM^2K)$. However, when samples arrive incrementally, the matrix can be updated (instead of a full recalculation) with time complexity $\mathcal{O}(M^2K)$ per sample using running sums.
- 2) Eigenvector Estimation: For each of the K one-vs-rest binary subtasks, the principal eigenvector of the $Q \in \mathbb{R}^{M \times M}$ matrix is computed, with time complexity $\mathcal{O}(M^3K)$. Faster alternatives, such as solving linear systems or using power iteration, may accelerate convergence [7].

Since M is typically small, the computational overhead per test sample remains low, making SML-OVR well-suited for real-time deployment. Unlike EM-based methods, which incur a time complexity of $\mathcal{O}(nMK)$ and generally do not support online updates, SML-OVR can achieve significantly higher efficiency. Its update cost is independent of n , allowing the approach to scale seamlessly with test stream of continuously arriving test samples. In practice, updating ensemble weights for each new sample takes only milliseconds.

Note that under small n (i.e., $n \leq M$ during the early test phase), the covariance matrix may be ill-conditioned. In such cases, fallback strategies such as majority voting are preferred over more sophisticated combination methods.

SML-OVR is also memory-efficient: storing classifier predictions requires $\mathcal{O}(nMK)$ space, while maintaining running sums for covariance updates needs only $\mathcal{O}(M^2K)$, independent of the size of the test set.

D. Summary of SML-OVR

In some scenarios, classifiers produce soft prediction probabilities rather than hard labels, e.g., neural networks output class probability vectors instead of discrete predictions. In this case, $f_j(\mathbf{x}_i)$ denotes a K -dimensional probability vector. These probabilities can be converted into binary labels for each one-vs-rest task to estimate ensemble weights. Once the weights are computed, they can be applied to the original soft predictions for improved aggregation.

The pseudo-code of SML-OVR is given in Algorithm 1. It can be implemented in a few line of codes.

Algorithm 1 Multi-Class Spectral Meta-Learner via One-Vs-Rest Class Splits (SML-OVR).

Input: M trained classifiers $\{f_j\}_{j=1}^M$;
 Streaming test data $\{\mathbf{x}_i\}_{i=1}^n$;
Output: Classification $\{\hat{y}_i\}_{i=1}^n$ for $\{\mathbf{x}_i\}_{i=1}^n$.
// Online Inference
for $i = 1 : n$ **do**
 if $i \leq M$ **then**
 // Use Simple Averaging
 Calculate $\hat{y}_i = \arg \max_k \sum_{j=1}^M f_{j,k}(\mathbf{x}_i)$;
else
 // Calculate Ensemble Weights
 for $k = 1 : K$ **do**
 // Convert Probabilities to Hard Binary Prediction
 Compute class prediction $\hat{f}_{j,k}(\mathbf{x}_i) \in \{0, 1\}$ from the probability vector $f_{j,k}(\mathbf{x}_i)$;
 Compute Q_k or update it incrementally, whose ij -th element $q_{ij} = \mathbb{E}[(\hat{f}_{i,k}(\mathbf{x}) - \mathbb{E}[\hat{f}_{i,k}(\mathbf{x})])(\hat{f}_{j,k}(\mathbf{x}) - \mathbb{E}[\hat{f}_{j,k}(\mathbf{x})])]$;
 Perform eigendecomposition of Q_k for its leading eigenvector \mathbf{v}_k ;
 end for
 Compute entries $\{\bar{v}_j\}_{j=1}^M$ by Eq. (12);
 Calculate \hat{y}_i by Eq. (13);
end if
end for

E. Comparison with Prior Extensions of SML

Several prior studies have sought to extend the SML to multi-class classification. This subsection highlights the key differences among these approaches:

- 1) The original SML approach [7] directly estimates classifier performance via the BCA π and offers a rigorous theoretical foundation for binary classification. Unlike the Dawid-Skene approach, SML does not estimate confusion matrices or compute ψ and η .
- 2) Jaffe *et al.* [19] addressed the binary case by estimating the class imbalance b , enabling the derivation of each classifier's ψ and η , and facilitating confusion matrix estimation. They also explored a one-vs-rest extension to multi-class problems, but their analysis was focused on confusion matrix estimation. Furthermore, they introduced an ambiguity theorem showing that multi-class performance cannot be uniquely inferred from classifier predictions alone.
- 3) Li *et al.* [18] proposed a direct extension of SML to the multi-class setting by incorporating soft prediction probabilities into the covariance computation. However, this approach is not applicable to black-box models where prediction probabilities are inaccessible.
- 4) The proposed SML-OVR method preserves the core principle of the original SML, computing ensemble weights proportional to classifier performance, without relying on estimates of ψ and η .

IV. DATASETS AND BASE CLASSIFIERS

Thirteen public datasets of text, image, or time series classification were used in the experiments. These datasets were selected to demonstrate the effectiveness and versatility of SML-OVR and combination approaches across varying data characteristics, numbers of classes, and dataset sizes.

A. Text Datasets

Four text classification datasets were used:

- Three sentiment analysis datasets: Multimodal Opinion-level Sentiment Intensity (MOSI) [24], Twitter US Airline Sentiment (TUSAS) [25], and TweetEval [26]. Continuous sentiment scores were discretized into evenly split categorical labels.
- One topic classification dataset: Web of Science (WOS) [27].

Table I summarizes their statistics.

Ten different large language models (LLMs), which are decoder-only Transformer models for text generation from Huggingface [28], were employed for classification tasks. Their parameters range from [1, 7] billions. The classification results were generated using the following prompt:

“Classify the sentiment/topic of the following text into one of [K] classes: [ClassNames].

Text: [Text].

Class: ”

Only class predictions were used, whereas token probabilities were not considered. If a model's response did not match any of the predefined classes (fewer than 1% of cases), a random prediction was assigned. As no training was required, all samples of the datasets were used for test purpose.

B. Time Series Datasets

Four public time series datasets were used:

- Two EEG-based motor imagery datasets from MOABB [29]: BNCI2014001 [30] and HighGamma [31], where subjects imagined body part movements.
- The SEED dataset [32], where subjects viewed video clips and EEG signals were recorded to decode discrete emotional states.
- The Handwriting dataset [33], capturing smartwatch motion data as subjects wrote the 26 letters of the alphabet.

Table II shows the dataset statistics. For EEG datasets, leave-one-subject-out cross-validation was employed: each subject was used as test set once, and each of the remaining subjects contributed a base classifier. For Handwriting, 11 deep neural network classifiers with different architectures were trained on the same training set.

C. Image Datasets

Five image classification datasets from the DomainBed benchmark [22] were used: VLCS [37], PACS [38], OfficeHome [39], TerraIncognita [40], and DomainNet [41]. Each dataset contains multiple domains with the same label space. Table III summarizes their statistics.

TABLE I
STATISTICS AND EXAMPLES OF THE FOUR TEXT DATASETS.

Dataset	# Classes	# Samples	Max Class-Imbalance	Example Text
MOSI	3	2,199	1.04:1	this movie isn't just bad its diabolical
TUSAS	3	14,640	3.88:1	@VirginAmerica it was amazing, and arrived an hour early. You're too good to me.
TweetEval	3	59,899	2.42:1	School is over and it's Friday I love life
WOS	7	46,985	4.44:1	A novel approach to tuning an inverted F antenna is investigated by using PZT materials ...

TABLE II
STATISTICS OF THE FOUR TIME SERIES DATASETS.

Dataset	# Subjects	# Classes	# Samples	Feature & Classifier
BNCI2014001	9	4	5,184	Common Spatial Patterns [34]
HighGamma	14	4	11,244	Common Spatial Patterns [34] + Linear Discriminant Analysis
SEED	15	3	152,730	Differential Entropy [32] + Linear Discriminant Analysis
Handwriting	-	26	425	Autoformer [35], etc., from time series library [36]

Leave-one-domain-out cross-validation was applied. ResNet-50 [42] models pre-trained on ImageNet were used and fine-tuned on each domain with empirical risk minimization. For fair ensemble evaluation despite the limited number of domains, each training domain contributed 10 independently trained models using different random seeds.

TABLE III
STATISTICS OF THE FIVE IMAGE DATASETS.

Dataset	# Domains	# Classes	# Samples	Max Class-Imbalance
VLCS	4	5	10,729	10:1
PACS	4	7	9,991	52:1
OfficeHome	4	65	15,588	7:1
TerraIncognita	4	10	24,330	1495:1
DomainNet	6	345	586,575	318:1

V. EXPERIMENTS

All algorithms were implemented in Python, and the code is available on GitHub¹.

A. Algorithms

We compare the following approaches for unsupervised ensemble:

- 1) Voting, which employs majority voting, or more precisely, plurality voting, where the class receiving the highest number of votes is selected, even if it does not achieve an absolute majority.
- 2) Worker Agreement with Aggregate (WAwA), which computes the majority vote label and estimates each classifier's reliability as the fraction of its predictions that agree with the majority vote. A weighted majority vote is then computed using these reliability scores.
- 3) Dawid-Skene [12], which is a probabilistic model that jointly estimates the true labels and classifier reliability by learning a confusion matrix for each classifier. It is optimized via the EM algorithm to iteratively infer latent true labels and update classifier-specific error rates.

- 4) Zhang *et al.* [13], which uses second- and third-order empirical moments to compute a better initialization for the Dawid-Skene model.
- 5) Generative model of Labels, Abilities, and Difficulties (GLAD) [43], which uses a probabilistic model to simultaneously model the three elements, optimized with EM.
- 6) Multi-Annotator Competence Estimation (MACE) [44], which is a probabilistic method that accounts for spamming behavior by modeling label distributions for unreliable annotators, optimized using EM.
- 7) Matrix-Mean-Subsequence-Reduced (M-MSR) [45], which estimates classifier reliability by filtering out extreme values in iterative alternating updates, handling adversarial classifiers that deviate arbitrarily from the Dawid-Skene model.
- 8) ZenCrowd [46], which uses the probabilistic graphical model to create micro-tasks for uncertain cases and applies EM-based inference over factor graphs to estimate true links and classifier reliability, optimized with EM.
- 9) Participant-Mine voting (PM) [47], [48], which iteratively optimizes for truths and per-classifier reliability by minimizing weighted disagreement between observations and inferred truths.
- 10) Label Aggregation (LA) [49] uses a dynamic Bayesian network to estimate classifier qualities and true labels by traversing all labels twice.
- 11) Label-Aware Autoencoders (LAA) [50] uses a neural network model, integrating an encoder and a decoder to infer true labels through minimizing a reconstruction loss, optimized by backpropagation.
- 12) Enhanced Bayesian Classifier Combination (EBCC) [51] iteratively optimizes a variational inference framework that models each true label as a mixture of latent subtypes, where classifier reliability varies by subtype.
- 13) SML-OVR, which is the proposed approach that uses one-vs-rest converted binary labels in SML covariance matrix calculation.

¹<https://github.com/sylyoung/TestEnsemble>

B. Main Results

This subsection presents the main results under the offline setting, where the entire set of unlabeled test data is available for transductive analysis. This allows SML-OVR and other combination methods to compute ensemble weights using the full test set predictions. Performance metrics are reported in Tables IV and V. In Table V, each dataset contains multiple subdatasets, and leave-one-out cross-validation was applied as discussed. For fairer comparison across datasets with varying scales of performance score, we also reported the average rank of each approach across all evaluation cases.

Key observations are as follows:

- 1) Ensemble methods consistently outperformed the average performance of individual classifiers. While the best single classifier occasionally surpassed all ensemble methods, identifying it in advance is infeasible. In contrast, combining base classifiers generally yields more robust and reliable performance. Further discussion is provided in Section V-D.
- 2) No single ensemble method dominated across all datasets. The Dawid-Skene EM-based approach showed stable and competitive performance, while extensions of it did not consistently offer improvements.
- 3) The proposed SML-OVR method achieved the best overall performance among all test-time ensemble approaches. These results highlight the promise of ensemble learning in black-box settings where ground-truth labels are unavailable.

C. Online Test-Time Ensemble

This subsection evaluates the performance of SML-OVR in online inference settings, where test samples arrive sequentially rather than being available for transductive offline analysis. In such cases, SML-OVR incrementally updates ensemble weights with each new test sample. Notably, many approaches, such as LAA, require iterative optimization and are therefore unsuitable for online use.

Experimental results are summarized in Table VI. SML-OVR consistently outperformed voting on almost all datasets. Moreover, its online performance remained comparable to the offline setting, with negligible degradation. The only exception was observed on the DomainNet dataset, where early performance was hindered due to the small number of initial test samples and the high number of classes.

D. Unsupervised Ensemble Ranking and Pruning

A long-standing question in ensemble learning is whether combining predictions is superior to simply selecting the best-performing classifier [52]. As observed in Tables IV and V, the top individual classifier occasionally outperforms all ensemble methods. Additionally, selecting a single classifier reduces inference and query costs by eliminating the need for combination strategies.

However, there is no reliable method or theoretical guarantee to determine whether model selection or combination will yield better performance in a given scenario. To investigate

this further, we conducted experiments to assess the potential of SML-OVR for ranking and pruning classifiers.

Recall that the core principle behind SML-OVR and many combination techniques is to assign higher weights to base classifiers with superior estimated performance. This enables not only aggregation but also unsupervised ensemble ranking and pruning, which are also explored in prior works [53].

In the following experiments, base classifiers were ranked using their SML-OVR weights, after which the lowest-ranked classifier was pruned from the ensemble. The remaining base classifiers were re-aggregated using SML-OVR until only three base classifiers are left in the aggregation. Experimental results on LLM-based text classification datasets are shown in Fig. 3, which yields the following findings:

- 1) SML-OVR accurately estimated relative classifier performance without ground-truth labels. The inferred rankings closely matched with ground-truth BCA-based performance rankings.
- 2) Underperforming models were successfully filtered out, enabling the identification of high-performing LLMs without requiring labeled data.
- 3) Nonetheless, the best classifier was not always assigned the highest weight, indicating that classifier selection alone may not be optimal. None of the current approaches could achieve it reliably.
- 4) Pruning low-performing models further improved ensemble performance while reducing computational cost. Notably, combining the top three models ranked by SML-OVR surpassed the best individual classifier (by true BCA) on all four datasets, highlighting the advantage of ensemble pruning.

E. Statistical Tests

To assess whether the performance of the SML-OVR ensemble significantly differed from that of other methods, two-sided Wilcoxon signed-rank tests were conducted. The resulting p -values were adjusted using the Benjamini-Hochberg False Discovery Rate correction [54]. The results are shown in Table VII, which reports comparisons on LLM-based text classification datasets. Overall, SML-OVR predictions exhibited statistically significant differences from most other algorithms.

F. Computational Cost

Table VIII reports the computational cost of SML-OVR evaluated on an Intel(R) Xeon(R) Platinum 8176 CPU @ 2.10GHz. For datasets of multiple subdatasets, the first subdataset was designated as the test set for measuring computation time. Except for DomainNet, which is substantially larger, SML-OVR processes the entire test set in under one second, whereas iterative expectation-maximization approaches may require several minutes. Notably, in the online setting, the update for each incoming test sample completes only in tens of milliseconds.

To further reduce computational overhead in scenarios with a large number of classes, ensemble weight updates may be deferred until a batch of test samples has been collected, rather than being executed on every individual sample.

TABLE IV

BCA (%) AND ACCURACY (% IN PARENTHESES) ON THE TEXT AND TIME SERIES CLASSIFICATION DATASETS. “SINGLE” DENOTES THE (LOWEST-AVERAGE-HIGHEST) PERFORMANCE OF BASE CLASSIFIERS. BEST AND SECOND-BEST RESULTS ARE MARKED IN BOLD AND UNDERLINED, RESPECTIVELY. AVERAGE RANKS ARE ALSO REPORTED FOR FAIRER COMPARISON ACROSS DATASETS WITH VARYING SCALES.

Algorithm	MOSI	TUSAS	TweetEval	WOS	Handwriting	Avg.	Rank
Single	53.17–63.46–77.13 (52.75–63.17–77.08)	54.62–66.69–79.46 (60.45–71.69–83.20)	47.32–58.78–70.48 (38.28–53.47–70.68)	17.11–32.91–47.27 (15.57–29.52–46.16)	9.39–17.10–22.12 9.41–16.79–22.12	- -	- -
Voting	71.52 (71.17)	71.63 (77.98)	66.01 (58.10)	38.08 (32.91)	19.80 (19.53)	53.41 (51.94)	8
WAwA	70.32 (69.94)	70.54 (77.59)	65.20 (56.68)	41.16 (35.13)	20.35 (20.00)	53.51 (51.87)	7
Dawid-Skene	76.77 (76.63)	77.97 (79.52)	71.54 (67.24)	44.75 (38.68)	20.28 (20.00)	58.26 (56.41)	2
Zhang <i>et al.</i>	77.26 (77.17)	78.48 (77.93)	41.71 (32.64)	7.36 (9.18)	20.69 (20.47)	45.10 (43.48)	5
GLAD	70.27 (69.90)	70.52 (77.56)	65.13 (56.62)	41.67 (35.50)	20.13 (19.76)	53.54 (51.87)	10
MACE	65.27 (64.80)	65.28 (73.93)	64.36 (55.84)	<u>46.02</u> (38.51)	20.11 (19.76)	52.21 (50.57)	13
M-MSR	67.69 (67.26)	70.52 (77.57)	63.42 (54.21)	42.47 (36.06)	20.38 (20.00)	52.90 (51.02)	11
ZenCrowd	68.13 (67.71)	70.51 (77.56)	62.74 (53.18)	43.29 (36.59)	20.35 (20.00)	53.00 (51.01)	12
PM	64.21 (63.71)	64.97 (75.04)	60.72 (50.42)	46.20 (38.61)	22.02 (21.65)	51.62 (49.89)	8
LA	70.23 (69.85)	70.52 (77.58)	64.45 (55.61)	43.00 (36.41)	20.75 (20.47)	53.79 (51.98)	6
LAA	<u>76.83 (76.72)</u>	<u>78.06 (79.29)</u>	71.67 (68.30)	43.90 (36.90)	18.96 (18.35)	57.88 (55.91)	3
EBCC	72.46 (72.12)	73.04 (77.77)	69.43 (62.83)	37.53 (34.02)	20.69 (20.47)	54.63 (53.44)	4
SML-OVR	75.52 (75.31)	76.50 (79.84)	70.65 (65.88)	43.74 (37.10)	22.02 (22.11)	57.69 (<u>56.05</u>)	1

TABLE V

BCA (%) AND ACCURACY (% IN PARENTHESES) ON THE EEG AND IMAGE CLASSIFICATION DATASETS. BEST AND SECOND-BEST RESULTS ARE MARKED IN BOLD AND UNDERLINED, RESPECTIVELY. AVERAGE RANKS ARE ALSO REPORTED FOR FAIRER COMPARISON ACROSS DATASETS WITH VARYING SCALES.

Algorithm	BNCI2014001	HighGamma	SEED	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Avg.	Rank
Voting	50.52	57.86	62.73 (63.08)	72.51 (79.78)	85.43 (84.37)	67.93 (69.02)	33.77 (41.20)	38.71 (39.25)	58.68 (60.64)	11
WAwA	51.25	58.54	63.91 (64.26)	72.73 (80.72)	86.67 (85.39)	68.51 (69.54)	33.47 (41.05)	39.12 (39.74)	59.28 (61.31)	4
Dawid-Skene	51.14	58.09	61.63 (61.98)	75.96 (76.87)	89.38 (88.07)	69.66 (70.21)	40.72 (45.20)	41.51 (41.82)	61.01 (61.67)	<u>2</u>
Zhang <i>et al.</i>	49.98	54.43	55.84 (56.04)	66.16 (67.19)	89.35 (88.02)	68.41 (69.24)	35.68 (43.18)	36.91 (39.05)	57.10 (58.39)	13
GLAD	51.87	58.19	64.83 (65.18)	72.74 (80.60)	86.03 (84.88)	68.23 (69.29)	34.83 (43.29)	40.11 (40.39)	59.60 (61.71)	5
MACE	52.45	55.05	65.01 (65.38)	69.35 (78.98)	85.56 (85.02)	68.35 (69.40)	31.20 (39.32)	38.60 (39.16)	58.20 (60.60)	10
M-MSR	51.41	58.39	64.30 (64.64)	72.19 (80.92)	86.56 (85.34)	68.28 (69.33)	33.42 (41.07)	39.26 (39.84)	59.23 (61.37)	7
ZenCrowd	51.37	58.20	65.31 (65.66)	72.35 (81.09)	87.12 (85.79)	68.31 (69.36)	35.18 (44.03)	38.73 (39.28)	59.57 (61.85)	6
PM	50.91	55.38	61.43 (61.71)	70.49 (80.17)	85.97 (83.85)	65.90 (67.08)	33.83 (45.95)	38.78 (39.45)	57.84 (60.56)	12
LA	51.70	58.38	65.27 (65.61)	72.28 (81.04)	87.03 (85.73)	68.63 (69.62)	35.35 (44.69)	38.06 (38.76)	59.59 (<u>61.94</u>)	3
LAA	51.46	59.16	64.03 (64.39)	73.41 (76.02)	88.92 (87.59)	66.48 (67.28)	30.51 (39.93)	30.10 (33.15)	58.01 (59.87)	8
EBCC	51.95	57.41	65.16 (65.50)	74.09 (77.85)	87.33 (86.18)	66.47 (68.27)	33.91 (43.82)	37.19 (39.08)	59.19 (61.26)	9
SML-OVR	<u>52.20</u>	<u>59.07</u>	65.43 (65.73)	<u>75.40 (80.82)</u>	88.29 (87.07)	<u>68.74 (69.90)</u>	<u>37.44 (45.72)</u>	<u>40.20 (40.73)</u>	<u>60.22 (62.28)</u>	1

TABLE VI

AVERAGE BCA (%) AND ACCURACY (% IN PARENTHESES) ON 13 DATASETS.

Algorithm	MOSI	TUSAS	TweetE.	WOS	Handw.	BNCI.	HighG.	SEED	VLCS	PACS	OfficeH.	TerraI.	DomainNet
Voting	71.52 (71.17)	71.63 (77.98)	66.01 (58.10)	38.08 (32.91)	19.80 (19.53)	50.52	57.86	62.73 (63.08)	72.51 (79.78)	85.43 (84.37)	67.93 (69.02)	33.77 (41.20)	38.71 (39.25)
SML-OVR (Offline)	75.52 (75.31)	76.50 (79.84)	70.65 (65.88)	43.74 (37.10)	22.02 (22.11)	52.20	59.07	65.43 (65.73)	75.40 (80.82)	88.29 (87.07)	68.74 (69.90)	37.44 (45.72)	40.20 (40.73)
SML-OVR (Online)	75.52 (75.31)	76.54 (79.88)	70.65 (65.87)	43.75 (37.12)	21.88 (22.02)	51.12	58.79	64.74 (64.07)	75.11 (80.88)	88.04 (86.80)	68.18 (69.14)	37.04 (45.09)	39.69 (39.91)

VI. CONCLUSIONS

This paper presented SML-OVR, a hyperparameter-free, computationally efficient, and privacy-preserving approach for test-time ensemble combination with multiple black-box classifiers. SML-OVR extends the unsupervised SML approach from binary to multi-class classification. Extensive experiments across diverse applications demonstrated that SML-OVR consistently outperforms traditional ensemble approaches. Moreover, SML-OVR also supports ensemble ranking and pruning. A notable limitation of the current imple-

mentation is the increased computational cost associated with a large number of classes. Note that the SML-OVR relies on the underlying assumptions to be effective.

Future research directions include:

- 1) Regression tasks: While ensemble methods for regression have been shown to outperform individual regressors under the assumption of conditional independence [4], designing aggregation strategies that reliably exceed uniform-weighted averaging remains an open challenge.
- 2) Robustness of combination: In settings where classifiers

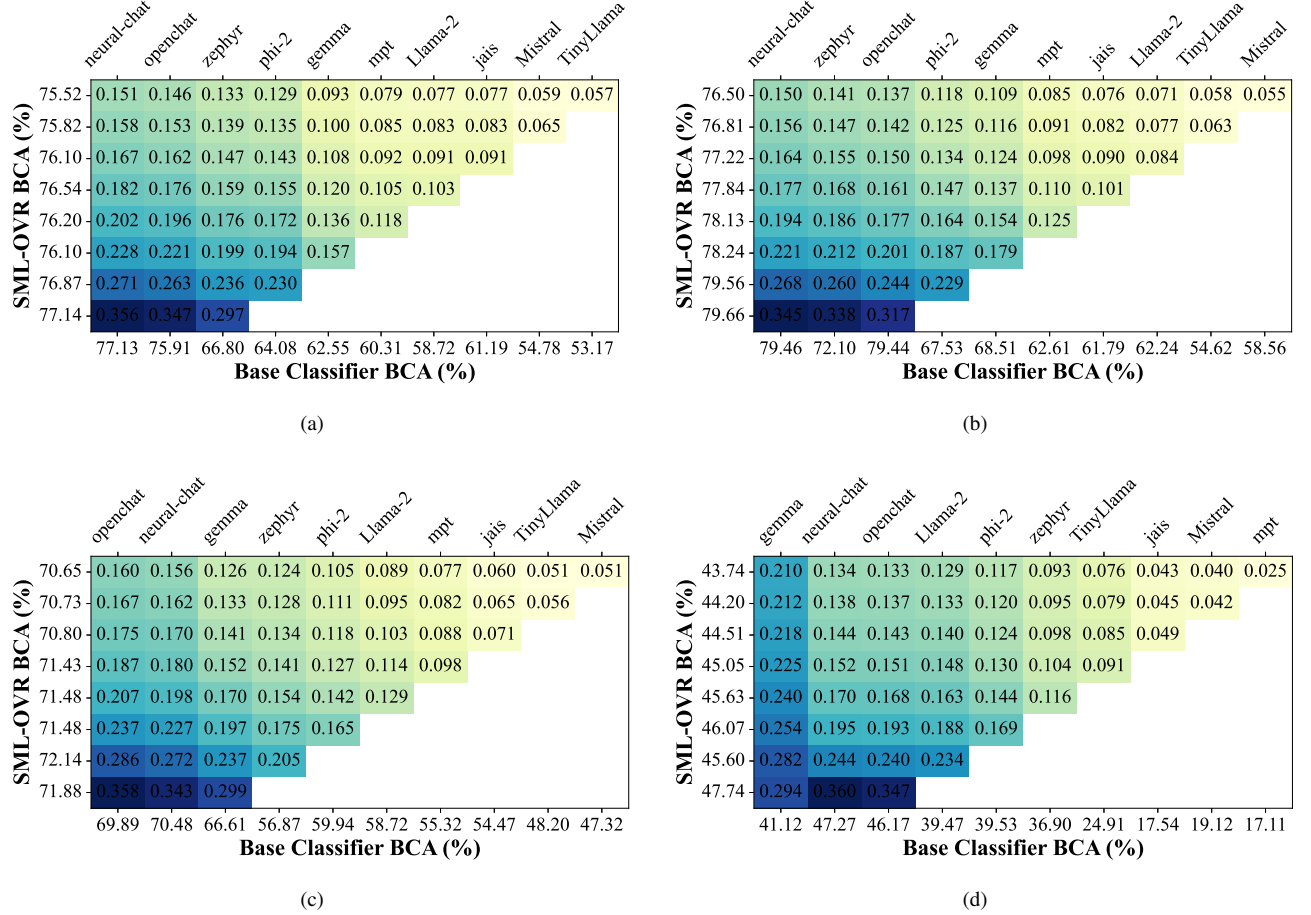


Fig. 3. Unsupervised ensemble ranking and pruning using SML-OVR on four text classification datasets (a) MOSI; (b) TUSAS; (c) TweetEval; and, (d) WOS. Heatmap values are the ensemble weights, i.e., \bar{v} of formula (12), assigned to each classifier. Rows represent sequential pruning steps, where the classifier with the lowest estimated weight is removed at each step. Model names (abbreviated HuggingFace public repository IDs) are shown on the top x-axis, and their true BCAs are listed on the bottom x-axis. The corresponding SML-OVR ensemble BCAs after each pruning step are shown on the y-axis.

TABLE VII
ADJUSTED p -VALUES BETWEEN SML-OVR AND OTHER APPROACHES.
 p -VALUES SMALLER THAN 0.05 ARE MARKED IN BOLD.

SML-OVR vs.	MOSI	TUSAS	TweetEval	WOS
Voting	0.2349	0.0000	0.0000	0.0000
WAwA	0.0122	0.0000	0.0000	0.0000
Dawid-Skene	0.0001	0.0000	0.0000	0.0000
Zhang <i>et al.</i>	0.0003	0.0000	0.0000	0.0000
GLAD	0.0000	0.0000	0.0000	0.0000
MACE	0.0000	0.0000	0.0000	0.0000
M-MSR	0.2874	0.0000	0.0000	0.0000
ZenCrowd	0.7847	0.0000	0.0000	0.0000
PM	0.7847	0.0000	0.0000	0.0000
LA	0.0306	0.0000	0.0000	0.4657
LAA	0.0002	0.0000	0.0000	0.0000
EBCC	0.0010	0.0000	0.0000	0.0003

are accessed via APIs, detecting compromised models and developing robust aggregation strategies is essential.

- 3) Semi-supervised approaches: Existing ensemble methods typically assume a fully unsupervised setting. However, semi-supervised learning that leverages limited labeled data has demonstrated success in deep learning and may yield performance improvements over purely

TABLE VIII
COMPUTATION TIME OF SML-OVR IN THE OFFLINE SETTING USING THE FULL TEST SET, USING EIGENDECOMPOSITION FOR PRINCIPAL EIGENVECTOR COMPUTATION.

Dataset	Eigendecomposition Matrix Dim.	Number of Classes	Computation Time (seconds)
MOSI	(10, 2199)	3	0.0289
TUSAS	(10, 14640)	3	0.1833
TweetEval	(10, 59899)	3	0.7180
WOS	(10, 46985)	7	1.9534
Handwriting	(11, 425)	26	0.4789
BNCI2014001	(8, 576)	4	0.0194
HighGamma	(13, 880)	4	0.0080
SEED	(14, 10182)	3	0.2417
VLCS	(30, 1415)	5	0.0829
PACS	(30, 2048)	7	0.1683
OfficeHome	(30, 2427)	65	2.1659
TerraIncognita	(30, 4741)	10	0.5680
DomainNet	(50, 48129)	345	278.3481

unsupervised combinations.

ACKNOWLEDGMENT

This research was supported by the Shenzhen Science and Technology Program under Grant JCYJ20220818103602004,

the Open Foundation of Henan Key Laboratory of Brain Science and Brain-Computer Interface Technology under Grant HNBBL230204, and Shijiazhuang Science and Technology Bureau 2511303107A.

REFERENCES

- [1] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–36, 2021.
- [2] J. Li, Z. Yu, Z. Du, L. Zhu, and H. T. Shen, "A comprehensive survey on source-free domain adaptation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5743–5762, 2024.
- [3] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Trans. Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.
- [4] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.
- [5] T. G. Dietterich, "Ensemble methods in machine learning," in *Int'l Workshop on Multiple Classifier Systems*, 2000, pp. 1–15.
- [6] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.
- [7] F. Parisi, F. Strino, B. Nadler, and Y. Kluger, "Ranking and combining multiple predictors without labeled data," *Proc. National Academy of Sciences*, vol. 111, no. 4, pp. 1253–1258, 2014.
- [8] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [9] R. E. Schapire, "A brief introduction to boosting," in *Proc. Int'l Joint Conf. Artificial intelligence*, Stockholm, Sweden, Jul. 1999, pp. 1401–1406.
- [10] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2296–2319, 2016.
- [11] A. Narasimhamurthy, "Theoretical bounds of majority voting performance for a binary classification problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1988–1995, 2005.
- [12] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.
- [13] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, Dec. 2014.
- [14] A. Jaffe, E. Fetaya, B. Nadler, T. Jiang, and Y. Kluger, "Unsupervised ensemble learning with dependent classifiers," in *Proc. Int'l Conf. Artificial Intelligence and Statistics*, Cadiz, Spain, May. 2016, pp. 351–360.
- [15] P. A. Traganitis, A. Pagès-Zamora, and G. B. Giannakis, "Blind multi-class ensemble classification," *IEEE Trans. Signal Processing*, vol. 66, no. 18, pp. 4737–4752, 2018.
- [16] U. Shaham, X. Cheng, O. Dror, A. Jaffe, B. Nadler, J. Chang, and Y. Kluger, "A deep learning approach to unsupervised ensemble learning," in *Proc. Int'l Conf. Machine Learning*, New York City, NY, Jun. 2016, pp. 30–39.
- [17] M. E. Ahsen, R. M. Vogel, and G. A. Stolovitzky, "Unsupervised evaluation and weighted aggregation of ranked classification predictions," *Journal of Machine Learning Research*, vol. 20, pp. 1–40, 2019.
- [18] S. Li, Z. Wang, H. Luo, L. Ding, and D. Wu, "T-TIME: Test-time information maximization ensemble for plug-and-play BCIs," *IEEE Trans. Biomedical Engineering*, vol. 71, no. 2, pp. 423–432, 2024.
- [19] A. Jaffe, B. Nadler, and Y. Kluger, "Estimating the accuracies of multiple classifiers without labeled data," in *Proc. Int'l Conf. Artificial Intelligence and Statistics*, San Diego, CA, May. 2015, pp. 407–415.
- [20] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
- [21] W. Zhang, L. Deng, L. Zhang, and D. Wu, "A survey on negative transfer," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 2, pp. 305–329, 2023.
- [22] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. Int'l Conf. Learning Representations*, Vienna, Austria, May. 2021.
- [23] J. Liang, R. He, and T. Tan, "A comprehensive survey on test-time adaptation under distribution shifts," *Int'l Journal of Computer Vision*, vol. 133, no. 1, pp. 31–64, 2025.
- [24] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 82–88, 2016.
- [25] Y. Wan and Q. Gao, "An ensemble sentiment classification system of twitter data for airline services analysis," in *IEEE Int'l Conf. Data Mining Workshop*, Atlantic City, NJ, Nov. 2015, pp. 1318–1325.
- [26] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proc. Int'l Workshop on Semantic Evaluation*, Vancouver, Canada, Aug. 2017, pp. 502–518.
- [27] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical deep learning for text classification," in *IEEE Int'l Conf. Machine Learning and Applications*, Cancun, Mexico, Dec. 2017, pp. 364–371.
- [28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "HuggingFace's Transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.
- [29] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066011, 2018.
- [30] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. Mueller-Putz *et al.*, "Review of the BCI competition IV," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012.
- [31] R. T. Schirmer, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
- [32] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks," *IEEE Trans. Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.
- [33] M. Shokoochi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing DTW to the multi-dimensional case requires an adaptive approach," *Data Mining and Knowledge Discovery*, vol. 31, pp. 1–31, 2017.
- [34] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-r. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.
- [35] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Advances in Neural Information Processing Systems*, Virtual, Dec. 2021, pp. 22419–22430.
- [36] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *arXiv preprint arXiv:2407.13278*, 2024.
- [37] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *Proc. IEEE Int'l Conf. Computer Vision*, Sydney, Australia, Dec. 2013, pp. 1657–1664.
- [38] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int'l Conf. Computer Vision*, Venice, Italy, Oct. 2017, pp. 5542–5550.
- [39] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, Hawaii, Jul. 2017, pp. 5018–5027.
- [40] S. Beery, G. Van Horn, and P. Perona, "Recognition in Terra Incognita," in *Proc. European Conf. Computer Vision*, Munich, Germany, Sep. 2018, pp. 456–473.
- [41] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int'l Conf. Computer Vision*, Seoul, Korea, Oct. 2019, pp. 1406–1415.
- [42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, Jun. 2016, pp. 770–778.
- [43] J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo, "Whose vote should count more: optimal integration of labels from labelers of unknown expertise," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2009.
- [44] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, GA, Jun. 2013, pp. 1120–1130.

- [45] Q. Ma and A. Olshevsky, "Adversarial crowdsourcing through robust rank-one matrix completion," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2020, pp. 21 841–21 852.
- [46] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proc. Int'l Conf. World Wide Web*, New York, NY, Apr. 2012, pp. 469–478.
- [47] B. Aydin, Y. S. Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas, "Crowdsourcing for multiple-choice question answering," in *Proc. AAAI Conf. Artificial Intelligence*, Québec City, Canada, Jul. 2014, pp. 2946–2953.
- [48] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int'l Conf. Management of Data*, Snowbird, UT, Jun. 2014, pp. 1187–1198.
- [49] Y. Yang, Z.-Q. Zhao, G. Wu, X. Zhuo, Q. Liu, Q. Bai, and W. Li, "A lightweight, effective, and efficient model for label aggregation in crowdsourcing," *ACM Trans. Knowledge Discovery from Data*, vol. 18, no. 4, pp. 1–27, 2024.
- [50] L. Yin, J. Han, W. Zhang, and Y. Yu, "Aggregating crowd wisdoms with label-aware autoencoders," in *Proc. Int'l Joint Conf. Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 1325–1331.
- [51] Y. Li, B. Rubinstein, and T. Cohn, "Exploiting worker correlation for label aggregation in crowdsourcing," in *Proc. Int'l Conf. Machine Learning*, Long Beach, CA, Jun. 2019, pp. 3886–3895.
- [52] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?" *Machine Learning*, vol. 54, pp. 255–273, 2004.
- [53] L. Rokach, "Collective-agreement-based pruning of ensembles," *Computational Statistics & Data Analysis*, vol. 53, no. 4, pp. 1015–1026, 2009.
- [54] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society*, vol. 57, no. 1, pp. 289–300, 1995.