Learning Objective Adaptation by Correlation-Based Model Reuse

Lanjihong Ma, Yao-Xiang Ding, Peng Zhao, *Member, IEEE*, and Zhi-Hua Zhou, *Fellow, IEEE*

***Abstract*— In open-environment machine learning (open ML), the learning objectives can vary according to specific real-world requirements. Models tailored for initial objectives may not be appropriate for the varied objectives. Retraining models from scratch for every single objective can be computationally intensive. Therefore, it is desirable to reuse models trained on the original objectives to help learn under the varied objectives. To this end, it is essential to characterize the objective correlations to better reuse the models. Previous works only consider the relative importance between pairs of previous and varied objectives, also known as *previous-varied objectives correlations*, ignoring correlations among the original objectives themselves. In this article, we demonstrate the importance of *cross-original objective correlations*. We propose a novel approach that employs the *optimal transport* technique to model correlations across all previous and varied objectives and then facilitates model reuse by utilizing learned transportation discrepancies to incorporate model reusabilities. Our empirical results show that our approach significantly outperforms existing benchmarks and well captures the underlying objective structure, validating the importance of accurate objective correlation modeling for learning with varied objectives.**

***Index Terms*— Correlation exploration, learning with varying objectives, model reuse, open-environment machine learning (open ML).**

I. Introduction

LEARNING objectives of machine learning models refer to the specific goals or tasks that machine learning algorithms aim to achieve and define what the algorithms are trying to learn or accomplish during the process. Traditionally, the learning objective is often assumed to be designed by the user and kept invariant during the learning procedure. In recent years, there has been a growing trend to address open-environment machine learning (open ML) [1] in real-world applications, where significant changes, e.g., input and output spaces, hypothesis spaces, and learning objectives, exist among previous and current learning tasks. Among them, the *changing objective* is less studied but quite challenging to handle. In an open and dynamic environment, the learning objectives are defined by the specific task or domain applications and can therefore vary beyond what already trained models can accommodate.

Despite the significance of this topic, the exploration of learning with varied objectives remains relatively rare in machine learning research. One approach to handling varied objectives is training a new model whenever the objective changes. However, this can result in significant computational waste, and the original training data may be limited—especially in open ML scenarios where objectives frequently change. Is there a more efficient alternative to retraining from scratch? A more advanced method, inspired by the idea of model reuse [2], considers adapting already trained models to varied objectives. We formally define this class of methods as objective adaptation (OA) methods. The central challenge of such adaptation lies in the requirement to utilize obtained information to decide "how much" each model should be reused—a concept referred to as the *reusability* of the base models.

One approach to constructing reusability in OA is to explore the *previous-varied objective correlations*, defined in this article as the relationships between the original objectives and the varied new objectives. The key idea is that if the varied objectives are not significantly different from the original ones, models trained on the original objectives can still be useful for the new objectives. Without explicitly modeling these correlations, Li et al. [3] proposed an OA method that adjusts the original hypothesis based on the target performance measure, leveraging the observation that many performance measures are correlated. Later, Ding and Zhou [4] modeled the varied objectives as a weighted linear combination of the original objectives, with the weights representing objective importance and explicitly capturing the previous-varied objective correlations.

However, existing studies have overlooked the crucial role of the correlations among the original objectives themselves, which we define in this article as the *cross-original objective correlations*. In this article, we emphasize that the cross-original objective correlations are crucial to the previous-varied correlations, which, in turn, are crucial for OA in learning with varied objectives.

To illustrate, consider the example of Fig. 1. Suppose that the learner faces a classification task with three classes, 1–3, with originally trained one-vs-one classifiers: $f_{1,2}$, $f_{2,3}$, and $f_{1,3}$. Now, suppose that the objective changes to $\ell_{3,*}$, where $*$ represents the other classes and focuses on distinguishing

Received 25 November 2023; revised 1 June 2024 and 27 September 2024; accepted 17 November 2024. This work was supported in part by the National Science Foundation of China under Grant 62250069, Grant U23A20382, and Grant 62206245; and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization. The work of Lanjihong Ma was supported by the Outstanding Ph.D. Candidate Program of Nanjing University. *(Corresponding author: Zhi-Hua Zhou.)*

Lanjihong Ma, Peng Zhao, and Zhi-Hua Zhou are with the National Key Laboratory for Novel Software Technology and the School of Artificial Intelligence, Nanjing University, Nanjing 210023, China (e-mail: maljh@lamda.nju.edu.cn; zhaop@lamda.nju.edu.cn; zhouzh@lamda.nju.edu.cn).

Yao-Xiang Ding is with the State Key Laboratory of Computer Aided Design and Computer Graphics, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: yxding@zju.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2024.3507362

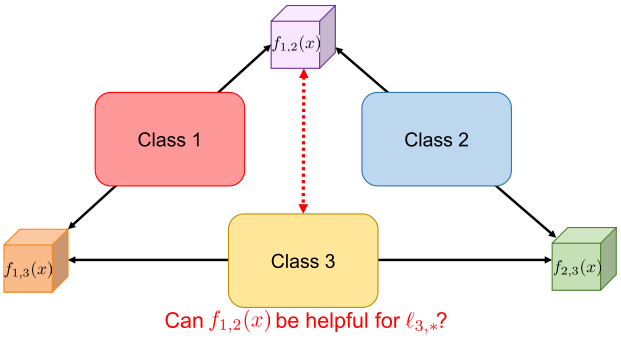


Fig. 1. Illustrative example showing the importance of considering correlations within original objectives when addressing objective discrepancies. The example begins with three trained one-vs-one binary classifiers, denoted as $f_{1,2}(x)$, $f_{2,3}(x)$, and $f_{1,3}(x)$, where the subscripts indicate their specific classification objectives. Suppose that the new objective shifts to $\ell_{3,*}$, focusing on whether class 3 can be correctly distinguished from the other classes. Without considering cross-original objective correlations, $f_{1,2}$ may appear irrelevant to the new objective. However, when combined with $f_{2,3}$ or $f_{1,3}$, $f_{1,2}$ provides additional marginal information that can benefit the new objective.

class 3 from the rest. The classifier $f_{1,2}$ appears irrelevant to this new objective using only previous-varied objective correlations. However, incorporating cross-original objective correlations shows that combining $f_{1,2}$ with $f_{2,3}$ or $f_{1,3}$ provides useful marginal information (the multiparty multiclass margin [5]). This example demonstrates the importance of cross-original correlations in revealing the true reusability of base models.

In this article, we revisit the problem of learning with varied objectives, where the learning objective differs from that of the original models. Assuming that the original models remain effective due to correlations between objectives, the task is restricted to OA methods, which adapt models to new objectives without retraining from scratch. The key challenge lies in determining each model's reusability, which determines its contribution to adaptation. To address this, we formalize the cross-objective correlations as transportation discrepancies within a unified framework using optimal transport techniques. Leveraging these correlations, we introduce a reusability criterion and propose the OA by correlation-based reuse (OACR) algorithm, which jointly estimates correlations and optimizes model reuse. Experiments on real-world applications demonstrate significant improvements, underscoring the importance of cross-objective correlations in learning with varied objectives.

The rest of this article is organized as follows. We introduce related works, the problem setups, and some preliminaries in Sections II and III. Then, we introduce the proposed approach in Section IV. We finally present the experimental results and conclude this article in Sections V and VI.

II. Related Work and Discussion

The problem of OA arises in learning with varied objectives in open ML, where the learning objective can frequently shift to meet real-world demands [1]. One line of approaches involves training new models for each objective independently, without leveraging prior information [6], [7]. Wu and Zhou [8] further categorize common objectives into a universal framework for multilabel learning. In contrast, another line of approaches focuses on model reuse, where previously trained models are adapted to new objectives, also known as OA [5], [9]. Li et al. [3] propose a weighted ensemble of original models without explicitly modeling objective correlations, while Ding and Zhou [4] address cases where objectives are not provided in advance. They model the varied objectives as a weighted combination of the original ones and learn these weights from implicit feedback to capture objective importance and reusability. Despite these advances, existing approaches overlook the correlations among objectives, which we argue are crucial for accurately modeling reusability and effectively learning with varied objectives.

OA is closely related to *multiobjective learning* [10], [11], [12], [13], but they address distinct challenges. OA focuses on adapting existing models to new objectives by leveraging relationships among objectives for targeted adjustments. In contrast, multiobjective learning seeks to optimize multiple objectives simultaneously, often using evolutionary algorithms [14], [15] to identify the Pareto front—the set of solutions where no single objective can be improved without degrading another. While multiobjective learning balances all objectives equally, OA prioritizes smooth transitions between objectives, exploiting their relationships to guide adaptation more effectively.

Another related area is domain adaptation (DA) [16], [17], [18], [19], which focuses on transferring existing models or knowledge to new tasks by addressing differences in data distribution between source and target domains. DA emphasizes knowledge transfer across domains with varying distributions. In contrast, OA assumes consistent data distributions but shifts learning objectives. This consistency enables OA to reuse representations learned by the original models, addressing challenges distinct from DA. The differing contexts of environmental variation in these problems render their methods generally incompatible.

More broadly, handling varied objectives is an essential yet underexplored challenge in open ML. Unlike traditional ML with static settings, open ML faces dynamic challenges, such as emerging new classes [20], [21], evolving features [22], [23], changing data distributions [24], [25], and varying objectives [26]. These challenges require the learning models to be flexible to adapt to open and dynamic environments. As objectives evolve, models must adjust seamlessly to remain effective. Addressing varied objectives is, therefore, fundamental for building robust and practical solutions in open ML.

III. Problem Formulation and Preliminaries

In this section, we first formulate the problem setup of OA and then provide preliminaries useful for the subsequent technical discussions.

A. Problem Formulation

The OA method considered in this article is formalized as follows. Suppose that the learner has already trained N models, denoted by $\mathcal{F}(x) := \{f_1(x), f_2(x), \ldots, f_N(x)\}$, where each $f_j(x)$, $j \in [N] : \mathcal{X} \mapsto \mathcal{Y}$ is a mapping from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$, trained independently according

to some unknown learning objectives $\{\ell_1, \ldots, \ell_N\}$.[1] Although the data distribution $\mathcal{D}$ remains unchanged, the learning objectives and their corresponding performance measures may vary based on real-world requirements, meaning that the trained models in $\mathcal{F}$ might not satisfy new objectives.

Suppose that the learning objective shifts to a known ℓ^*, which evaluates the quality of learning. The task is to obtain a new model that performs well on ℓ^*. A straightforward approach is to retrain a completely new model on the data distribution $\mathcal{D}$, using ℓ^* or its convex surrogate as the risk function and employing empirical risk minimization (ERM). However, retraining becomes inefficient in open ML, where objectives frequently change. By the time retraining is complete, objectives may have shifted again, leading to repeated training cycles where no model is operational during the transition.

An alternative approach is model reuse [2], which adapts existing models according to the new objectives. In the following, we leverage optimal transport theory to model the correlations among the objectives of the base models.

B. Preliminaries: Background for Optimal Transport

To better present our proposed approach, we first introduce some preliminaries on optimal transport theory, which is essential for modeling objective correlations.

Optimal transport quantifies the discrepancy between two probability distributions by measuring the effort required to transform one distribution into the other. This effort is represented by a cost matrix, typically denoted as M, which encodes the pairwise costs of moving mass between points in the two distributions. Unlike traditional discrepancy measures that compare distributions in a fixed space, optimal transport accounts for the underlying geometries of the probability spaces, making it particularly effective for spaces with intricate or heterogeneous structures. Before entering applications, we outline the fundamental definitions related to optimal transport.

Definition 1 (Transport Polytope): Given two N-dimensional probability distributions, denoted by Δ_a and Δ_b, the transport polytope between Δ_a and Δ_b, denoted by $U(\Delta_a, \Delta_b)$, is defined as the polyhedral set of $N \times N$ matrices as

$$U(\Delta_a, \Delta_b) := \left\{ P \in \mathbb{R}_+^{N \times N} \,\middle|\, P\mathbf{1}_N = \Delta_a,\, P^\top \mathbf{1}_N = \Delta_b \right\}$$

where $\mathbf{1}_N$ is the N-dimensional vector of ones.

Definition 2 (Optimal Transport): Given an $N \times N$ cost matrix M, the optimal transport from the distribution Δ_a to Δ_b on the cost matrix M is defined as the minimum Frobenius dot product of the cost matrix M and all possible transport $P \in U(\Delta_a, \Delta_b)$, formally defined as

$$\mathrm{OT}(\Delta_a, \Delta_b, M) := \min_{P \in U(\Delta_a, \Delta_b)} \langle P, M \rangle.$$

The transport polytope represents all possible joint distributions that transfer mass from a source distribution to a target distribution. Optimal transport identifies the best transport plan that minimizes the transportation cost. Represented by a triplet (Δ_a, Δ_b, M), optimal transport measures the discrepancy between the simplex pair (Δ_a, Δ_b) defined on the cost matrix M. However, there are computational challenges. Solving the optimal transport problem often involves handling large linear programs, particularly for high-dimensional data or large sample sizes. Furthermore, computing the pairwise costs between all points in the two distributions can be computationally intensive. Recent advancements propose various approximation and computational strategies [27], [28], [29], [30] that effectively address these issues.

Unlike traditional metrics that compare distributions pointwise in a fixed space, optimal transport accounts for the underlying geometry and structure of the probability spaces. This makes it particularly effective in capturing the heterogeneous characteristics of cross-original objective correlations.

In the following, we demonstrate how to model and adapt to varied learning objectives by capturing the correlations between original and new objectives using the optimal transport theory. By constructing a cost matrix that encodes pairwise transport costs between objectives, the method quantifies the discrepancy between objective distributions. This allows for the modeling of cross-original objective correlations, which are crucial for accurately estimating model reusability in new contexts. The optimal transport framework provides a solid mathematical foundation for this estimation, facilitating the computation of a transport plan that minimizes the cost of transforming the original objective distribution to the new one. This ensures that the adaptation process accounts for the underlying geometries and structures of the objective spaces, offering a robust mechanism for effective model reuse and OA in dynamic learning environments.

IV. Proposed Approach

In this section, we first introduce the OA task, explaining how we can construct models with the original obtained models. Then, we introduce how to estimate the reusability based on the cross-original objective correlations and how this contributes to our OA procedure.

A. Correlation-Based Model Reuse

Recall that the learning task is to adapt the base models to the varied objective ℓ^*. One effective approach is to construct an ensemble of the base models. Specifically, given the set of base models $\mathcal{F}(x) := \{f_1(x), f_2(x), \ldots, f_N(x)\}$, we first construct a weighted linear ensemble of the base models and then add an extra perturbation to this ensemble. Formally, the target model $h(x)$ is constructed as

$$h(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i f_i(x) + \boldsymbol{\beta}^\top \phi(x) \right) \tag{1}$$

where $f_i(x)$ are the base models and $f_\delta(x) := \boldsymbol{\beta}^\top \phi(x)$ is the auxiliary δ-function. Notably, the extension from binary classification to multiclass classification is conceptually straightforward using one-hot encoding for labels. For simplicity and without loss of generality, this article focuses on the binary classification case, as all methods described can be naturally extended to the multiclass scenario.

[1] Strictly speaking, the objective function should be $\ell_{(x,y)\sim\mathcal{D}}(f(x), y)$. For simplicity, we use this notation where no confusion arises.

Notice that the model above is a parameterized model. To obtain the target model, we need to learn the weight parameters $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_N]^\top$ and the δ-function. Intuitively, if one model's objective is more closely related to the varied objective, it should receive more weight. This observation leads to the need for a method to measure the "discrepancy" between the objective of a base model and the new objective, which we define as the *previous-varied objective correlations*. Existing works [4], [31] assume the new objectives to be weighted combinations of the original objectives, directly formalizing the previous-varied objective correlations through weights, without considering the correlations between the original objectives—what we call *cross-original objective correlations*.

Specifically, the learner aims to learn the weight vector $\boldsymbol{\alpha}$ in the $(N-1)$-dimensional simplex,[2] denoted as $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_N]^\top$, such that the new objective ℓ^* is expressed as a weighted sum of the original objectives

$$\ell^* = \sum_{i=1}^{N} \alpha_i \ell_i \tag{2}$$

where ℓ_i for $i \in [N]$ represents the learning objective of base model $f_i(x)$. However, when the objectives of the base models are correlated, co-linearity can destabilize this regression. Meanwhile, the assumption implied by (2) restrains the new objective to be within the convex hull induced by original objectives, which may not always hold. To address these issues, rather than directly considering the one-to-one discrepancy from any original objective to the new objective, we also model the internal correlations within the original objectives. This allows us to better understand the relationship between the new objective and the original objectives. Specifically, we span the objective space by defining the following normalized objective vector as the basis:

$$\mathcal{L}_{(x,y)\sim\mathcal{D}}(h) := \frac{1}{Z(h)}\left[\ell_1(h(x), y), \ldots, \ell_N(h(x), y)\right]^\top \tag{3}$$

where $Z(h) = \sum_{i=1}^{N} \ell_i(h(x), y)$ is the normalization factor, making $\mathcal{L}(h)$ lying in the $(N-1)$-dimensional simplex.

Assuming that the base models are well trained according to their respective objectives, the consistency between the models and objectives in the objective space can be expected. Each objective of a base model can be represented as a one-hot distribution in this space. For example, the vector $[1, 0, \ldots, 0]$ represents the objective of base model $f_1(x)$ in the objective space. When constructing the target model $h(x)$ using the weighted ensemble in (1), the projection of the new objective onto the objective space is simply the weight vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^\top$. By transforming each base model into its corresponding distribution in the objective space, we can define two key quantities: the *correlated discrepancy* (CD) between a given base model and the target model, and *correlated reusability* (CR), which is defined as the negative weighted sum of the correlated discrepancies.

Formally, the CD between ℓ_i and ℓ^*, denoted by CD_i, and the CR of all base models, denoted by CR, are defined as

$$\mathrm{CD}_i := \mathrm{OT}(\boldsymbol{e}_i, \boldsymbol{\alpha}, M) \tag{4}$$

$$\mathrm{CR} := -\sum_{i=1}^{N} \alpha_i \cdot \mathrm{CD}_i \tag{5}$$

where $\boldsymbol{e}_i$ is the one-hot N-dimensional vector representing the original objective of $f_i(x)$, $\boldsymbol{\alpha}$ is the projection of the new objective onto the objective space, and M is the $N \times N$ cost matrix encoding the correlations between the objectives.

The intuition behind these definitions can be explained as follows. CD_i corresponds to the CD between the objective of $f_i(x)$ and the new objective. Maximizing CR is equivalent to minimizing the weighted CD, which intuitively seeks a consensus point in the objective space that is closest to all original objectives, weighted by the discrepancy and the model weights in (1).

Note that when the entries of the cost matrix M are set to 1 for all off-diagonal entries and 0 for the diagonal, this formalization reduces to the traditional one-to-one discrepancy in (2), which does not consider the cross-original objective correlations. Moreover, even though the bases of the objective space in (3) are not guaranteed to be orthogonal, the co-linearity issue is alleviated by the cost matrix M, which explicitly encodes the correlations. Therefore, the cost matrix M plays a critical role in the overall learning procedure.

In the following, we describe how to obtain this cost matrix M, either through learning from data or model specifications.

B. Learning the Cost Matrix

In this section, we introduce our method for learning the cost matrix. A crucial element in this procedure is the model specification [32], [33], which is the detailed description of a model's functionality, capabilities, and applicable contexts. These specifications can include statistical properties, semantic descriptions, and other relevant metadata, enabling the precise identification and reuse of models for various tasks, even beyond their original purposes. They are frequently provided alongside models on platforms such as Hugging Face.

We consider two main scenarios.

1) When no data are available, we can only infer objective correlations from model specifications.
2) When a limited amount of data is available, we can extract additional information from this data.

We first focus on the scenario where only model specifications are available. Among various types of model descriptions, we use the reduced kernel mean embedding (RKME) specification [34] within the learnware framework [2], [32] to demonstrate how to construct the cost matrix from specifications. The same principle applies to other specifications.

Example: In this example, we illustrate the process of generating the cost matrix using the RKME specification.

1) Obtaining RKME for Base Models: The first step is to find the corresponding vectorized representation in a unified reproducing kernel Hilbert space (RKHS) for each base model. Specifically, we introduce the detailed steps as follows.

1) *Kernel Mean Embedding (KME):* Compute the empirical KME for each base model f_i

$$\mu_k^i = \frac{1}{m}\sum_{j=1}^{m} k(x_j, \cdot) \tag{6}$$

[2]The $(N-1)$-simplex Δ_{N-1} is the set of all N-dimensional vectors $[\alpha_1, \alpha_2, \ldots, \alpha_N]$ such that $\alpha_i \geq 0$ for all i, and $\sum_{i=1}^{N} \alpha_i = 1$.

Algorithm 1 Constructing Cost Matrix With RKME

Require: A set of base models $\{f_i\}_{i=1}^N$, learning rate η, number of reduced points R
Ensure: Cost matrix M
Initialize cost matrix $M \in \mathbb{R}^{N\times N}$
for each model f_i **do**
  Calculate μ_k^i according to (6) ▷KME for model i
  repeat
    Update γ^i according to (7): ▷Update weights
    Update Z^i according to (8) ▷Update points
  until convergence
  Construct $(\psi, t)_i$ according to (9) ▷RKME for model i
end for
for each pair of models (i, j) **do**
  Calculate $V_{i,j}$, $Q_{i,j}$ according to (10) (11)
  Calculate $M_{i,j}$ according to (12) ▷Final cost matrix
end for
return Cost matrix M

where $\{x_j\}_{j=1}^m$ are the training samples and $k(x_j, \cdot)$ represents the kernel function k evaluated at the training sample x_j and is a function mapping from the input space to the RKHS.

2) *Reduced Set Construction:* For base model f_i, the reduced set is the set that approximately preserves the corresponding empirical KME information. Specifically, this can be implemented by finding the smaller set of points $\{z_r^i\}_{r=1}^R$ with weights $\{\gamma_r^i\}_{r=1}^R$ such that

$$\left\| \mu_k^i - \sum_{r=1}^{R} \gamma_r^i k\left(z_r^i, \cdot\right) \right\|_{H_k}^2$$

is minimized. Here, H_k denotes the RKHS associated with kernel k and R is the number of reduced points prespecified by the learner, determining the tradeoff between computational efficiency and approximation accuracy. This optimization problem can be solved by the following two-step alternating optimization: fix $\{z_r^i\}$ and calculate γ^i by

$$\gamma^i = K^{-1}C \tag{7}$$

where K is the kernel matrix with entries $K_{sr} = k(z_s^i, z_r^i)$ and C is a vector with entries $C_s = (1/m)\sum_{j=1}^m k(z_s^i, x_j)$; and then, fix γ^i to update z_r^i using gradient descent

$$z_r^i \leftarrow z_r^i - \eta \frac{\partial}{\partial z_r^i} \left\| \mu_k^i - \sum_{r=1}^{R} \gamma_r^i k\left(z_r^i, \cdot\right) \right\|_{H_k}^2 \tag{8}$$

where η is the predefined learning rate.

3) *Computing RKME:* The RKME specification for the ith base model f_i, denoted by $(\psi, t)_i$, can then be constructed by

$$(\psi, t)_i = \sum_{r=1}^{R} \gamma_r^i k(z_r^i, \cdot). \tag{9}$$

This process results in a compact and informative representation of the model's training distribution.

Algorithm 2 ASQP

Input: The cost matrix M, the regularization parameter λ_1, stopping threshold ϵ.
Initialization: randomly simplex $\boldsymbol{\alpha}$, $\mathrm{CR} = -1$, $\Delta > \epsilon$
while $\Delta \geq \epsilon$ **do**
  Fix $\boldsymbol{\alpha}$, solve CD_i according to (14) for $i \in [N]$;
  Fix CD_i for $i \in [N]$ and update $\boldsymbol{\alpha}$ by solving (15);
  Calculate CR' for current result according to (5).
  Calculate the CR gain $\Delta \leftarrow |\mathrm{CR} - \mathrm{CR}'|$;
  Restore the current results $\mathrm{CR} \leftarrow \mathrm{CR}'$;
end while
Output: Weight vector $\boldsymbol{\alpha}$, correlated reusability CR.

2) Constructing the Cost Matrix: Once the vectorized representations are obtained, the cost matrix can be constructed based on RKHS discrepancies and error overlaps. Specifically, when model specifications are available, the discrepancy between pairs of base models $(f_i(x), f_j(x))$ is defined as

$$V_{i,j} = \|(\psi, t)_i - (\psi, t)_j\|_{H_k}^2. \tag{10}$$

In cases where additional data is accessible, it is possible to incorporate this data to derive a more informative cost matrix. To achieve this, we compute the error overlap $Q_{i,j}$, which quantifies the similarity in errors between $f_i(x)$ and $f_j(x)$

$$Q_{i,j} = \frac{m_{i,j}}{m_i + m_j - m_{i,j}}. \tag{11}$$

Subsequently, the metrics are uniformly combined

$$M_{i,j} = \gamma V_{i,j} + (1-\gamma) Q_{i,j}, \quad \text{for } i, j \in [N] \tag{12}$$

where γ represents the balance between relying on predefined model specifications and leveraging information learned from data. When the base model is suboptimal or there is a shift in data distribution, it is advantageous to prioritize model specifications by assigning a higher value to γ. In these circumstances, increasing γ reflects a stronger emphasis on predefined model characteristics, thereby reducing the influence of data-driven insights $Q_{i,j}$, which may be less reliable.

Remark 1 (Complexity and Scalability): For each model, calculating the KME with m samples requires $\mathcal{O}(m^2)$ operations. Building a reduced set for KME involves $\mathcal{O}(R^2)$ complexity for R reduced sets. In addition, calculating the RKHS discrepancy V, error overlap Q, and cost matrix M each requires the $\mathcal{O}(N^2)$ operations. Therefore, the total complexity is $\mathcal{O}(N \cdot (m^2 + R^2 + N))$. It is important to note that m is the number of selected training samples from the base models, and R is the number of reduced sets; both are usually very small. For scalable base-model pools where N is large, the quadratic complexity term of $\mathcal{O}(N^2)$ may limit scalability. Techniques, such as kernel herding, can efficiently reduce complexities for large-scale computations.

This example demonstrates how the RKME methodology can be employed to construct a cost matrix for optimal transport modeling, considering both predefined specifications and learned data discrepancies. In the following, we introduce how to learn the weight parameter for each model.

Algorithm 3 OACR

1: **Input:** Data $(x_i, y_i) \sim \mathcal{D}, i \in [m]$, base models $\{f_1(x), \ldots, f_N(x)\}$, regularization parameter $\lambda_1, \lambda_2, \lambda_3$;
2: Calculate the cost matrix M according to Algorithm 1;
3: Obtaining the weight vector $\boldsymbol{\alpha}$ and the correlated reusability CR according to ASQP (Algorithm 2);
4: Learn the auxiliary δ-function according to (16);
5: Construct the target model according to (1);
6: **Output:** Target model $h(x)$.

C. Learn the Weight Parameters

When obtaining the cost matrix M from the data or model specifications, we can then learn the weight parameter $\boldsymbol{\alpha}$ in (1). Based on the above definition of CR, we obtain the weight vector $\boldsymbol{\alpha}$ by finding the projected coordinate of the new objective that has the highest CR or correspondingly smallest weighted correlated discrepancies to the objectives of base models so that the base models can be utilized as much as possible. Formally, we obtain $\boldsymbol{\alpha}$ by solving the following regularized optimization problem:

$$\begin{aligned}\boldsymbol{\alpha} &= \arg\max_{\boldsymbol{\alpha}\in\Delta_{N-1}} \mathrm{CR} + \frac{\lambda_1}{2}\|\boldsymbol{\alpha}\|_2^2 \\ &= \arg\min_{\boldsymbol{\alpha}\in\Delta_{N-1}} \sum_{i=1}^{N} \alpha_i \cdot \mathrm{OT}(\boldsymbol{e}_i, \boldsymbol{\alpha}, M) + \frac{\lambda_1}{2}\|\boldsymbol{\alpha}\|_2^2 \end{aligned} \tag{13}$$

where the extra regularization term $(\lambda_1/2)\|\boldsymbol{\alpha}\|_2^2$ avoids assigning too large weight on a single base model. Notice that when replacing the above CD_i by traditional discrepancy measures such as Euclidean norm, the above optimization problem reduces to a traditional quadratic programming (QP) problem, which finds the central point of several fixed points (which is exactly the base models in the objective space) according to square discrepancies and thus can be efficiently solved due to its convexity. However, when it comes to the CD, solving the optimization in (13) efficiently is not trivial. The key lies in the definition of CD that solving the optimal transport itself would be ill posed in stability. For fixed $\boldsymbol{\alpha}$, the corresponding transport polytope $U(\boldsymbol{e}_i, \boldsymbol{\alpha})$ defined in Definition 1 is fixed for fixed $\boldsymbol{e}_i$, and the optimal transport defined by Definition 2 can be efficiently found inside the convex polytope; when $\boldsymbol{\alpha}$ now turns into a variable, the transport polytope is changing with $\boldsymbol{\alpha}$, resulting in the nonconvexity of the corresponding optimal transport optimization.

To handle the difficulties in solving optimization (13), we propose the following approaches. We first separate the optimization into two phases and alternatively optimize one phase each time. The procedure repeats until converges. Specifically, the algorithm conducts the following procedures alternatively.

1) Fix $\boldsymbol{\alpha}$ and calculate CD CD_i for $i \in [N]$ with *Sinkhorn approximation* [27].
2) Fix CD_i and update $\boldsymbol{\alpha}$ by reducing the original optimization into a QP problem.

We formally propose the above approach as Alternative Sinkhorn-QP (ASQP).

At the beginning of the algorithm, a random $(N-1)$-dimensional simplex is initialized as $\boldsymbol{\alpha}$. Then, the algorithm enters the following two-step alternative optimization procedure until certain convergence is reached: the first step is to fix $\boldsymbol{\alpha}$ and optimize CD_i for $i \in [N]$, and the second step is to fix CD_i and optimize $\boldsymbol{\alpha}$.

In the first step, instead of directly optimizing CD_i by minimizing optimal transport $\mathrm{OT}(\boldsymbol{e}_i, \boldsymbol{\alpha}, M)$, which provides precise results but at high computational complexity, we solve a surrogate problem. This surrogate problem offers an approximation with bounded errors, significantly reducing computational complexity. Specifically, the surrogate is defined as

$$\mathrm{CD}_i^{\lambda_2} = \min_{P\in U(\boldsymbol{e}_i,\boldsymbol{\alpha})} \langle P, M\rangle - \frac{1}{\lambda_2} H(P) \tag{14}$$

where $H(P) = -P\log P = -\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij}\log p_{ij}$ is the element-wise entropy of P and $\lambda_2 > 0$ is the entropic regularization coefficient.

The introduction of the entropy term makes the surrogate problem strongly convex, ensuring fast convergence rates. In this case, we can apply the famous Sinkhorn algorithm [27] to solve the surrogate problem. The following theorem guarantees that the approximation error introduced by the extra regularization is bounded by a problem-related term, and the convergence rate of the surrogate problem is exponential.

Theorem 1: Let $\lambda_2 > 0$ be the entropic regularization parameter in the surrogate problem defined by (14). Let CD_i be the solution to the original optimal transport problem, and let $\mathrm{CD}_i^{\lambda_2}$ be the solution to the surrogate problem. In addition, let $\mathrm{CD}_i^{\lambda_2,k}$ denote the kth iterate of the Sinkhorn algorithm applied to the surrogate problem. Then, the following results hold.

1) *Proximity of Surrogate to Original:* There exists a constant $C > 0$, dependent on the specific problem parameters, such that the approximation error in terms of the Frobenius norm is bounded by

$$\|\mathrm{CD}_i^{\lambda_2} - \mathrm{CD}_i\|_F \le \frac{C}{\lambda_2}.$$

2) *Exponential Convergence Rate of the Surrogate:* The iterative solution $\mathrm{CD}_i^{\lambda_2,k}$ to the surrogate problem converges exponentially to $\mathrm{CD}_i^{\lambda_2}$. Specifically, there exists a constant $C' > 0$ such that the approximation error after k iterations is bounded by

$$\|\mathrm{CD}_i^{\lambda_2,k} - \mathrm{CD}_i^{\lambda_2}\|_F \le C'\exp(-\lambda_2 k).$$

Remark 2: The first result ensures that the solution of the surrogate problem defined in (14) is close to the original optimal transport solution. The second result guarantees that the surrogate problem can be solved efficiently using the Sinkhorn algorithm, with an exponential convergence rate. The overall computational complexity of solving the surrogate problem in (14) is thus $\mathcal{O}(N^2\log(N)/\lambda_2^2)$.

Next, we proceed to the second step of the alternating optimization, where CD_i is fixed and we optimize $\boldsymbol{\alpha}$. Specifically, with CD_i fixed, the problem in (13) reduces to the following QP problem:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}} \mathbf{c}^\top\boldsymbol{\alpha} + \frac{\lambda_1}{2}\cdot\boldsymbol{\alpha}^\top\mathbf{I}_N\boldsymbol{\alpha} \\ &\text{s.t. } \mathbf{1}_N^\top\boldsymbol{\alpha} = 1 \\ &\quad \mathbf{I}_N\boldsymbol{\alpha} \ge \mathbf{0}_N\end{aligned} \tag{15}$$

where $\mathbf{c}$ is constructed by taking CD_i as the ith entry, i.e., $\mathbf{c}_i = \mathrm{CD}_i$. Meanwhile, $\mathbf{I}_N$ is the N-dimensional identity matrix; $\mathbf{1}_N$ and $\mathbf{0}_N$ is N-dimensional vector of ones and zeros, respectively. The constraints guarantee $\boldsymbol{\alpha}$ to be a valid distribution. We can prove that the above QP problem is strictly convex and thus can be efficiently solved by various methods. We take the interior point method in the implementation, which is provably guaranteed to find a global optimal solution within $\mathcal{O}(N^{3.5})$ complexity in the worst case.

By alternatively repeating the above two-step procedure, the algorithm repeatedly updates CR according to (5) until certain stopping condition is met; the algorithm stops and outputs the weight parameter. We summarize the above procedure as the ASQP algorithm in Algorithm 2. In the following, we focus on how to learn the δ-function based on the CR.

D. Learn the Auxiliary δ-Function

When the weight vector $\boldsymbol{\alpha}$ in (1) is determined, the next step is to learn the δ-function. Note that the new objective may not always fall within the objective space, causing the weighted ensemble of base models to fail if the new objective differs significantly from previous ones. The δ-function facilitates retraining, indicating that our constructed hypothesis does not always rely on the weighted ensemble. We perform the learning in an end-to-end manner, allowing simultaneous learning of the parameter $\boldsymbol{\beta}$ and the feature representation $\phi(\cdot)$. Specifically, for a given new objective ℓ^*, we directly learn the δ-function by minimizing the following structural risk:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} \ell^*(h(x_i), y_i) + \frac{\lambda_3}{2} \cdot \frac{\|\boldsymbol{\beta}\|_2^2}{-\mathrm{CR}}. \tag{16}$$

According to the definition in (5), CR measures how much previous information aids the current hypothesis, determined by the distance between the target model $h(x)$ in (1) and the weighted ensemble of base models $\sum_{i=1}^{N} \alpha_i f_i(x)$ in the objective space. This regularization, combined with the auxiliary δ-function, enables a balance between relying on original models and retraining a new model. For instance, if the new objective significantly deviates from previous ones, the correlated discrepancies CD_i will be large, resulting in a highly negative CR. Consequently, the regularization term becomes dominant, allowing $\boldsymbol{\beta}$ to scale significantly by reducing the penalty with $(1/(-\mathrm{CR}))$. In this scenario, the δ-function in (1) dominates the weighted ensemble, indicating that the final classifier $h(x)$ will prioritize retraining the δ-function over trusting the weighted ensemble. This approach is reasonable, as there is limited reusable information, and the previous weighted ensemble should contribute less.

E. Overall Procedure

Finally, we summarize the whole learning procedure as the OACR algorithm in Algorithm 3 and Fig. 2. At the beginning of a learning task, we first obtain several base models $f_1(x), f_2(x), f_3(x), \ldots$, each with corresponding specification or data; when the objective varied, we take the objectives of these base models as the bases and span the corresponding objective space, to which we project the original models. Meanwhile, we can learn a cost matrix M from the model specifications or data. By obtaining this cost matrix and the coordinates of base models on the objective space, we can find the coordinates of the target model by minimizing the weighted distance on the objective space by the ASQP algorithm, according to which we construct the target model by $h(x) = \mathrm{sign}(\sum_{i=1}^{N} \alpha_i f_i(x) + \boldsymbol{\beta}^\top \phi(x))$, where $\boldsymbol{\alpha}$ is the weight distribution found by ASQP, and $\boldsymbol{\beta}^\top \phi(x)$ is the δ-function trained by ERM. In the following, we empirically evaluate our approach to real-world OA tasks.

V. Experiments

In this section, we conduct two real-world applications on OA. We first conduct experiments on multilabel learning, stating the correlations within labels are essential; then, we conduct OA as in [3], which adapts original models to specific performance measures. All experiments were conducted on a system equipped with Intel Xeon E5-2640 v4 (10C/20T, 2.4 GHz), 32-GB DDR4-2666 RAM, and NVIDIA RTX 2080Ti (11-GB GDDR6). All cost matrices used in the experiments are learned purely from RKME specifications, assuming that no additional data are available.

A. Exemplary Case: Multilabel Learning

We first conducted experiments on real-world multilabel data to verify the importance of cross-original objective correlations in OA. In multilabel learning, each instance is associated with multiple labels that describe different aspects of the instance. For example, a dog can be labeled as both "mammal" and "animal" simultaneously. Clearly, in this example, "mammal" and "animal" are correlated, a relationship dictated by the intrinsic structure of the label space. Utilizing such label correlations, or in our approach, cross-original objective correlations are central to our method.

To illustrate this, we consider a binary 2-label multilabel classification task. The learner is initially provided with four base models $f_1(x), \ldots, f_4(x)$, where $f_k(x)$ for $k = 1, 2, 3, 4$ are trained to maximize the accuracy for the kth label, denoted by ℓ_k. Suppose that now, the learning objective shifts to the Hamming loss, which evaluates the accuracy across all labels. The Hamming loss is formally defined as follows:

$$\ell^* := \frac{1}{ml} \sum_{i=1}^{m} \sum_{j=1}^{l} \mathbb{I}_{\hat{y}_{ij} \neq y_{ij}} = \frac{1}{4} \sum_{k=1}^{4} \ell_k$$

where m is the total number of instances, l is the number of labels, and $\hat{y}_{ij}$ is the prediction for the jth label of the ith sample. In this case, since the varied objective can be written as a weighted combination of the original objectives, the weights can be treated as a valid criterion for the previous-varied objective correlation.

If equal weights are assigned to each of the base models, it suggests that the average ensemble of base models may be a promising model for the new objective ℓ^*. However, this assumption does not hold in practice. One key difference between multilabel learning and single-label learning is the existence of label correlations [35], which implies a more complex internal structure within the base models. Therefore, determining model reusability solely based on weights may

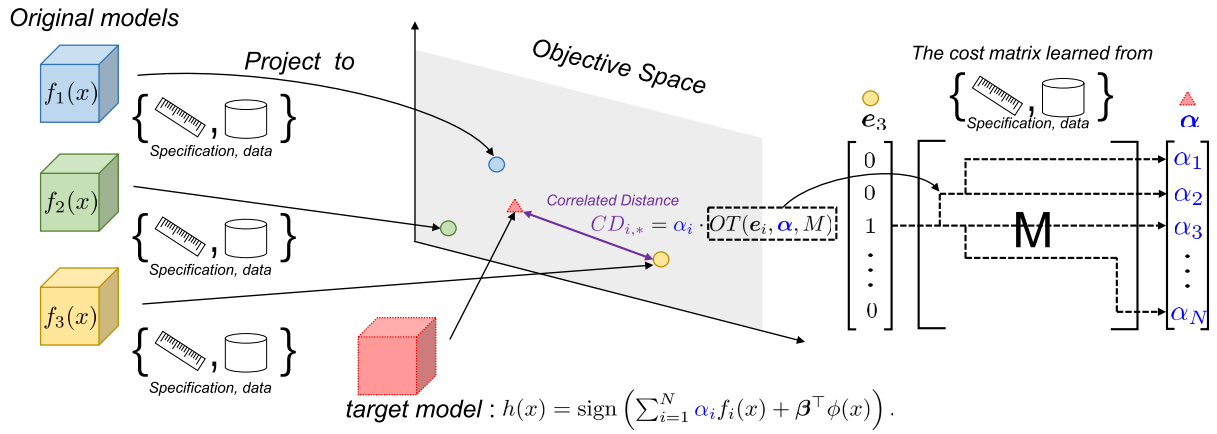
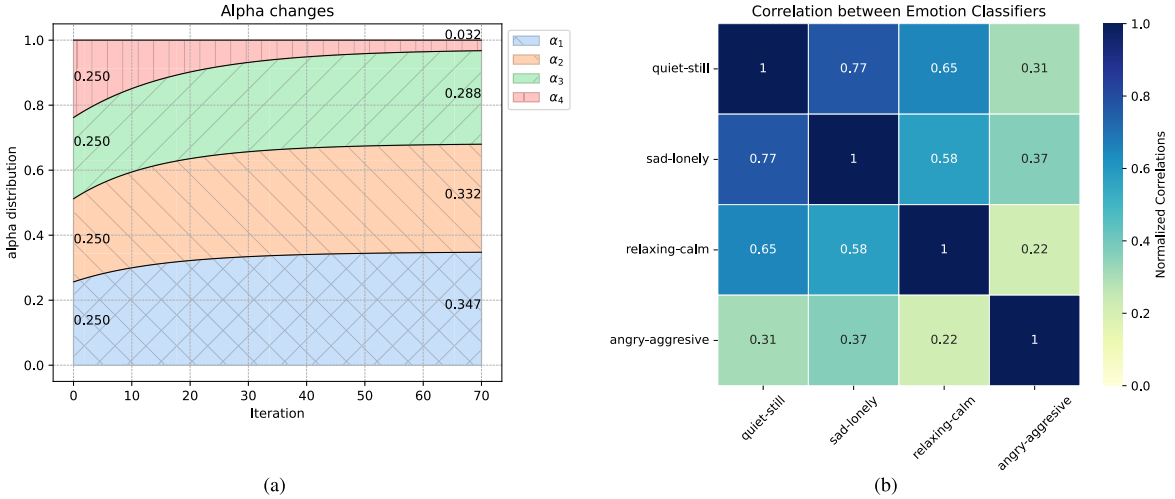


Fig. 2. Illustration of the OACR. We first transform the objectives of the base models onto an objective space, and then, by constructing the target model as a weighted ensemble of base models plus an extra disturb function, we can measure the correlated discrepancies between the objective of a base model and the new objective (represented by the purple double-headed arrow), as the weighted optimal transport (represented by dashed lines), where the optimal transport is the shortest transport from the objective of the base model (in this example, represented by a one-hot vector $\boldsymbol{e}_3$) to the new objective (the $\boldsymbol{\alpha}$ marked in blue) on a cost matrix M that is learned from model specifications or data.



(a) (b)

Fig. 3. (a) Weights assigned to base models by ASQP over iterations, where each color represents the weight assigned to a specific base model. (b) Label-correlation heatmap, with darker blocks indicating higher correlations. In the left plot, unlike the averaged ensemble that assigns equal weights of 0.25 to each model, ASQP assigns greater weights to the first three base models (with higher α_1–α_3) and the least weight to the fourth base model. This observation is explained by the right plot: the high correlation between "quiet-still," "sad-lonely," and "relaxing-calm" aligns with intuition as these emotional states are associated with a peaceful and calm atmosphere. Conversely, the low correlation between "angry-aggressive" and the other states is expected as it is characterized by high energy and intensity, sharply contrasting with the calmer emotions.

fail due to the presence of multicollinearities among the labels. This highlights the necessity of considering label correlations to accurately estimate model reusability and improve adaptation to new objectives.

To illustrate the above observation, we simulate the following scenarios on real-world multilabel data emotions [36], which consists of a collection of texts, each labeled with six different emotions: amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, and angry-aggressive. The dataset reflects the multilabel nature of human emotions, acknowledging that the emotional tones conveyed by the text can be correlated. We select the last four emotions as the original objectives in multilabel learning, denoted by ℓ_1–ℓ_4, and train their corresponding classifiers $f_1(x)$–$f_4(x)$. The varied objective is the Hamming loss $\ell^* = (1/4)\sum_{k=1}^{4} \ell_k$, which concerns the averaged accuracies over the four labels. We compare the proposed OACR (Algorithm 3) with the following contenders.

1) *Retrain From Scratch:* Without utilizing the original models, we directly optimize hamming loss on the available data. This method is implemented with an eight-layer multilayer perceptron (MLP).
2) *Averaged Ensemble:* Given that the varied objective can be represented as the weighted combination of original objectives with equal weight for each original objective, we can construct the ensemble of original classifiers according to (1), with an equal weight of 0.25 assigned to each of the original classifier.

All the auxiliary δ-functions are trained with the same structure of MLP and the same set of parameters as retrain from scratch, including optimizer and learning rate.

Experimental Result: We illustrate the weights assigned to each base model and the label-correlation heatmap in Fig. 3. Meanwhile, we present a plot of training epochs versus averaged accuracy in Fig. 4. Focusing first on Fig. 3, the left plot displays the weights assigned to base models by ASQP

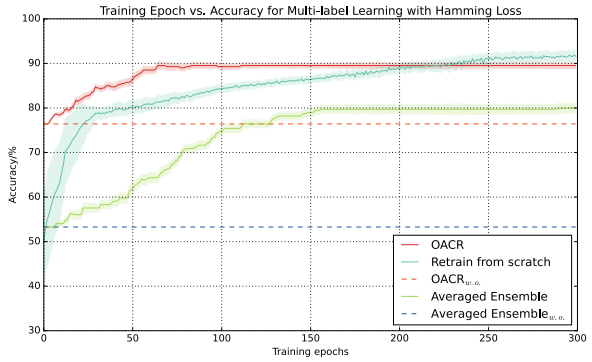


Fig. 4. Training epoch versus averaged accuracy. The plot compares the proposed OACR (red solid line) and OACR without δ-function (OACR$_{w.o.}$, red dashed line) with averaged ensemble methods: with δ-function (turquoise solid line) and without δ-function (turquoise dashed line). Retraining from scratch (green line) is also included. Performance is measured on the test set and averaged over ten trials with randomly initialized MLPs, with shaded areas indicating one standard deviation. OACR shows faster convergence (stabilizing around 100 epochs) and strong final performance, while the averaged ensemble achieves the worst results despite an improvement from the δ-function. Retraining achieves the best final performance but requires significantly more epochs, highlighting its higher computational cost compared to OACR.

over iterations. Each color represents the weight assigned to one base model. As we can observe, in contrast to the averaged ensemble, which assigns equal weights of 0.25 to each model, our proposed method assigns more weight to the first three base models (with higher α_1–α_3) and the least weight to the fourth base model. The right plot in Fig. 3 showcases the label-correlation heatmap, where a deeper block indicates a higher correlation between emotions. We find that the unbalanced weight assignment comes from varying correlation levels: the "quiet-still," "sad-lonely," and "relaxing-calm" emotions exhibit higher correlations, while the "angry-aggressive" emotion is more isolated, showing lower correlation levels with the other emotions. This weight distribution, differing from an equal assignment, arises from our approach's modeling of cross-original objective correlations. Turning our attention to Fig. 4, we present a training epoch versus averaged accuracy plot, with the training epoch corresponding to the training of the auxiliary δ-function. We present the performance of our proposed OACR and OACR$_{w.o.}$ (without δ-function) using a red solid line and a dashed line, respectively. Meanwhile, we illustrate the performance of averaged ensemble and averaged ensemble$_{w.o.}$ (without δ-function) with the greenish-blue solid and dashed lines, respectively. The green line represents retraining from scratch. All performances are measured on the test set and averaged over ten independent trials with randomly initialized MLPs. The shadow represents a one-standard deviation region. With examining the dashed lines, we observe that the weight distribution assigned by OACR has a relatively better initialization than the averaged ensemble. As for the solid lines, OACR reaches a stationary phase with commendable performance within a relatively small number of epochs (approximately 100 epochs). In contrast, while the auxiliary δ-function significantly enhances the averaged ensemble's performance from a suboptimal initialization, this improvement is limited. Consequently, the averaged ensemble exhibits the lowest final performance. Meanwhile, even though retraining from scratch achieves the highest final performance, it takes longer to reach the stationary phase (approximately 250 epochs), implying a potentially higher computational cost.

In conclusion, our proposed OACR outperforms the averaged ensemble due to the cross-original objective correlations and demonstrates a slightly lower averaged accuracy than retraining from scratch. However, it benefits from a much faster rate of convergence. This tradeoff that slightly sacrifices performance for significantly enhanced efficiency represents the central advantage of OACR, which we aim to emphasize.

TABLE I
STATISTICS OF REAL-WORLD DATASETS

Dataset	# Features	# Train Data	# Test Data	Imbalance Ratio (IR)
vehicle	18	446	400	0.98
usps	256	7291	2007	0.96
covertype	54	440000	141012	0.94
splice	60	1000	2175	0.92
reuters	8315	7770	3299	0.92
phishing	68	6055	5000	0.79
rna	8	59535	271617	0.77
letter	16	10500	5000	0.37
mitfaces	361	6977	24045	0.02

B. Model Adaptation With Specific Performance Measures

In this section, we conduct the OA task according to specific performance measures [3]. Specifically, we focus on the following four commonly used performance measures: accuracy, $F1$-score, area under the ROC curve (AUC), and average precision (AP). For each task, we select one of these measures as the varied new objective, while the remaining three measures are treated as the original objectives.

For instance, consider a scenario where accuracy is chosen as the new objective. In this case, $F1$-score, AUC, and AP are treated as the original objectives. We train base models to optimize these original objectives and then adapt these models to the new objective of accuracy. Similarly, if $F1$-score is selected as the new objective, the base models are initially trained to optimize accuracy, AUC, and AP, and then adapted to optimize $F1$-score. By systematically varying the new objective and keeping the others as original objectives, we aim to evaluate the effectiveness of our approach in adapting to different performance measures as the varied new objectives.

It is important to note that adapting from one to another is in fact nontrivial, even for balanced datasets [37]. The main reasons for this are differences in optimization compatibility, threshold dependency, and varying emphases on different error types. Specifically, accuracy, due to its alignment with cross-entropy loss, can be efficiently optimized via gradient descent. In contrast, $F1$-score and AUC are nondifferentiable, preventing direct optimization through similar methods. Furthermore, accuracy, $F1$-score, and AP values vary with classification thresholds, whereas AUC remains threshold-independent. In addition, accuracy treats all errors equally, precision focuses more on positive predictions, $F1$-score balances precision and recall, and AP considers the ranking of positive instances. These factors collectively highlight the complexity and challenges of adapting to these metrics interchangeably.

TABLE II
EXPERIMENTAL RESULTS OF THE OA TASKS. "↑" INDICATES "THE HIGHER THE BETTER." THE RESULTS ARE GIVEN IN THE MEAN ± STD (RANK) FORMAT. THE BEST RESULTS ARE EMPHASIZED WITH BOLD

Datasets	Algorithm	*Learning Objective*			
		accuracy/%↑	F1-scores/%↑	AUC/%↑	avg. precision/%↑
splice	OACR_{cvm}	**91.19 ± 0.11 (1)**	**91.41 ± 0.14 (1)**	**95.11 ± 0.02 (1)**	**95.55 ± 0.02 (1)**
	CAPO_{cvm}	89.52 ± 0.01 (2)	90.24 ± 0.01 (2)	94.01 ± 0.01 (2)	93.66 ± 0.02 (2)
	$\text{Retrain}_{\text{es}}$	87.21 ± 0.07 (3)	87.27 ± 0.06 (3)	92.52 ± 0.04 (3)	93.38 ± 0.04 (3)
letter	OACR_{cvm}	**87.08 ± 0.05 (1)**	**90.37 ± 0.04 (1)**	**86.13 ± 0.01 (1)**	85.88 ± 0.02 (2)
	CAPO_{cvm}	86.17 ± 0.01 (3)	89.72 ± 0.01 (3)	85.74 ± 0.02 (2)	**85.91 ± 0.03 (1)**
	$\text{Retrain}_{\text{es}}$	86.71 ± 0.07 (2)	90.09 ± 0.06 (2)	83.29 ± 0.04 (3)	84.46 ± 0.04 (3)
mitfaces	OACR_{cvm}	**81.73 ± 0.56 (1)**	68.34 ± 1.75 (2)	**91.21 ± 0.22 (1)**	**85.26 ± 0.22 (1)**
	CAPO_{cvm}	79.77 ± 0.01 (3)	**70.52 ± 0.03 (1)**	89.98 ± 0.02 (2)	82.89 ± 0.01 (2)
	$\text{Retrain}_{\text{es}}$	81.48 ± 0.07 (2)	67.28 ± 1.17 (3)	89.64 ± 0.35 (3)	82.53 ± 0.34 (3)
reuters	OACR_{cvm}	**98.16 ± 0.01 (1)**	**98.17 ± 0.01 (1)**	**98.35 ± 0.15 (1)**	97.49 ± 0.15 (2)
	CAPO_{cvm}	93.47 ± 0.01 (3)	96.59 ± 0.01 (3)	98.20 ± 0.01 (2)	**97.97 ± 0.03 (1)**
	$\text{Retrain}_{\text{es}}$	97.93 ± 0.08 (2)	97.85 ± 0.08 (2)	90.76 ± 2.41 (3)	87.18 ± 2.06 (3)
rna	OACR_{cvm}	**84.97 ± 0.40 (1)**	**82.40 ± 0.76 (1)**	**91.09 ± 0.34 (1)**	**86.61 ± 0.34 (1)**
	CAPO_{cvm}	N/A (3)	N/A (3)	N/A (3)	N/A (3)
	$\text{Retrain}_{\text{es}}$	84.23 ± 0.36 (2)	80.81 ± 1.05 (2)	86.45 ± 1.87 (2)	82.23 ± 1.65 (2)
usps	OACR_{cvm}	**98.09 ± 0.05 (1)**	**98.04 ± 0.06 (1)**	**99.14 ± 0.03 (1)**	**99.01 ± 0.03 (1)**
	CAPO_{cvm}	97.76 ± 0.01 (2)	97.61 ± 0.01 (2)	98.70 ± 0.01 (2)	97.98 ± 0.01 (2)
	$\text{Retrain}_{\text{es}}$	97.56 ± 0.08 (3)	97.48 ± 0.09 (3)	97.99 ± 0.23 (3)	97.61 ± 0.23 (3)
phishing	OACR_{cvm}	**93.66 ± 0.21 (1)**	**94.31 ± 0.23 (1)**	**98.12 ± 0.09 (1)**	**98.37 ± 0.09 (1)**
	CAPO_{cvm}	91.19 ± 0.01 (3)	91.64 ± 0.02 (3)	96.67 ± 0.01 (3)	97.19 ± 0.01 (3)
	$\text{Retrain}_{\text{es}}$	93.22 ± 0.05 (2)	93.87 ± 0.06 (2)	97.55 ± 0.09 (2)	97.85 ± 0.09 (2)
covertype	OACR_{cvm}	**79.31 ± 0.22 (1)**	**78.33 ± 0.68 (1)**	**83.08 ± 0.16 (1)**	**81.65 ± 0.16 (1)**
	CAPO_{cvm}	72.85 ± 0.01 (3)	75.76 ± 0.02 (2)	81.82 ± 0.01 (2)	81.16 ± 0.01 (2)
	$\text{Retrain}_{\text{es}}$	74.54 ± 0.12 (2)	71.27 ± 0.29 (3)	78.12 ± 0.04 (3)	80.72 ± 0.04 (3)
vehicle	OACR_{cvm}	**96.30 ± 0.31 (1)**	**96.38 ± 0.32 (1)**	**98.8 ± 0.01 (1)**	**99.22 ± 0.01 (1)**
	CAPO_{cvm}	94.96 ± 0.01 (2)	95.08 ± 0.02 (2)	97.68 ± 0.01 (2)	98.85 ± 0.01 (2)
	$\text{Retrain}_{\text{es}}$	93.61 ± 1.44 (3)	93.52 ± 1.58 (3)	77.65 ± 0.01 (3)	80.43 ± 0.01 (3)

TABLE III
AVERAGE PERFORMANCE RANK OVER ALL EXPERIMENTS

Algorithm	OACR_{cvm}	CAPO_{cvm}	$\text{Retrain}_{\text{es}}$
avg. rank	**1.08**	2.31	2.61

1) Data Description: Table I provides an overview of key statistics of the dataset we use from the LIBSVM datasets,[3] where the imbalance ratio (IR) is defined as the ratio of the number of samples in the minority class to those in the majority class. A value of IR close to 1 indicates a well-balanced dataset, typically resulting in strong correlations between various performance metrics. In contrast, an IR approaching 0 represents a highly imbalanced dataset, where the correlation between metrics weakens (e.g., accuracy and $F1$-score may diverge significantly in cases of extreme imbalance), reflecting a scenario in which the learning objectives vary considerably from their original ones.

[3]https://www.csie.ntu.edu.tw/~cjlin/libsvm/

In addition, we briefly introduce the machine learning tasks on these datasets as follows.

1) *Splice:* This dataset focuses on *bioinformatics*, where the task is to classify DNA sequences as either splice junctions—where DNA is split and recombined—or nonsplice regions, based on sequence information.
2) *Letter:* In the domain of optical character recognition (OCR), this dataset involves classifying 26 capital letters of the English alphabet into two categories: "A to M" and "N to Z," using pixel images of the letters.
3) *Mitfaces:* Pertaining to *facial recognition* task, this dataset is used to categorize face data into two predefined categories, utilizing face detection data.
4) *Reuters:* This dataset is from the field of text mining and natural language processing (NLP), where the task is to classify news articles into different predefined topics based on their textual content.
5) *RNA:* In *bioinformatics and genomics*, this dataset is used to classify RNA sequences into noncoding and coding categories based on free energy changes and their secondary structure formation as features.

6) *Usps:* This dataset focuses on OCR, where the goal is to classify handwritten digits "01234" apart from digits "56789" using images of the digits.
7) *Phishing:* From the domain of *cybersecurity*, this dataset aims to identify and classify websites as either phishing or legitimate based on various features.
8) *Covertype:* In *forestry*, this dataset is used for predicting the forest cover type based on cartographic information.
9) *Vehicle:* This dataset pertains to *computer vision and transportation*, where the task is to classify vehicles into two predefined categories based on their silhouettes.

2) Contenders: We compared the following approaches.

1) $OACR_{cvm}$ *(Algorithm 3):* For our proposed approach, we first train three core vector machine (CVM) [38] as the base models according to the original objectives and then we implement the δ-function in (1) by a eight-layer MLP.
2) $CAPO_{cvm}$ *[3]:* CAPO handles the OA problem in the function-level adaptation framework, except that it takes various trained models, which are not necessarily designed for the original objectives, and thus, the objective correlation is not taken into consideration. This approach can be seen as the ablation method without objective correlations. Different from the original CAPO that takes CVMs, RBF-kernelled neural network [39] and C4.5 decision tree [40] as its base models, we take five CVMs instead as its base models for $\mathsf{CAPO}_{\mathsf{cvm}}$, to get rid of the differences brought by the base models. We use the original codes and default parameters.
3) $Retrain_{es}$*:* Though retrain from scratch always guarantees promising performances, the cost is additional training epochs. To this end, we compare the proposed approach with a variant of retrain, that puts additional restriction on the number of training epochs. Specifically, $\mathsf{Retrain}_{\mathsf{es}}$ will conduct early stopping when our approach converges, and thus, $\mathsf{Retrain}_{\mathsf{es}}$ represents the performance of retraining a new model from scratch given (almost) same amount of computational resources. The implementation uses exactly the same eight-layer MLP with the same set of parameters, optimizer, and learning rate.

3) Experimental Results: Table II presented the performance on the varied objective by the above approaches, and we additionally present the ranks averaged on all tasks in Table III. The results are represented in percentile, with the best results emphasized by bold, and the N/A result stands for the algorithm does not output results within 24 h. As we can see, $\mathsf{OACR}_{\mathsf{cvm}}$ reaches the best overall performance. Compared with $\mathsf{Retrain}_{\mathsf{es}}$, which does not utilize any of the obtained models, our proposed approach achieves better performances in most tasks, the main reason is the faster convergence brought by the weighted ensemble base models, validating the rationality of the model reusing strategy; comparing $\mathsf{OACR}_{\mathsf{cvm}}$ with $\mathsf{CAPO}_{\mathsf{cvm}}$, which does not take the objective correlation into consideration, we state that the reusing strategies with explicitly learned previous-varied and cross-original objective correlations are both helpful for OA tasks.

VI. Conclusion and Future Work

This article addresses the challenge of learning with varied objectives by adapting preexisting models to new objectives without starting from scratch. Recognizing that smaller objective variation discrepancies correspond to more related models, we emphasize the importance of integrating objective correlations into the adaptation process. Unlike previous methods that overlook these correlations, we utilize a cost matrix with optimal transport techniques to accurately estimate objective variation discrepancies and guide hypothesis adaptation. Experiments on synthetic and real-world data show the effectiveness of our approach, validating the significance of considering objective correlations in OA.

While our framework demonstrates strong potential in adapting to varied learning objectives, it assumes a fixed data distribution with well-trained base models. In an open environment, distribution shifts in data, causing well-trained base models to become suboptimal and put for greater robustness in handling such cases. Future work will focus on improving the robustness of our proposed approach to data distribution shifts in a weakly trained model reuse framework for open ML.

References

[1] Z.-H. Zhou, "Open-environment machine learning," *Nat. Sci. Rev.*, vol. 9, no. 8, Aug. 2022, Art. no. nwac123.
[2] Z.-H. Zhou, "Learnware: On the future of machine learning," *Frontiers Comput. Sci.*, vol. 10, no. 4, pp. 589–590, Aug. 2016.
[3] N. Li, I. W. Tsang, and Z.-H. Zhou, "Efficient optimization of performance measures by classifier adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1370–1382, Jun. 2013.
[4] Y.-X. Ding and Z.-H. Zhou, "Preference based adaptation for learning objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 7839–7848.
[5] X.-Z. Wu, S. Liu, and Z.-H. Zhou, "Heterogeneous model reuse via optimizing multiparty multiclass margin," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6840–6849.
[6] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 16, 2003, pp. 313–320.
[7] T. Joachims, "A support vector method for multivariate performance measures," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 377–384.
[8] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3780–3788.
[9] P. Zhao, L.-W. Cai, and Z.-H. Zhou, "Handling concept drift via model reuse," *Mach. Learn.*, vol. 109, no. 3, pp. 533–568, Mar. 2020.
[10] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision Sciences*. Boca Raton, FL, USA: CRC Press, 2016, pp. 161–200.
[11] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 525–536.
[12] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1310–1322, Apr. 2020.
[13] A. Prajapati, A. Parashar, and A. Rathee, "Multi-dimensional information-driven many-objective software remodularization approach," *Frontiers Comput. Sci.*, vol. 17, no. 3, Jun. 2023, Art. no. 173209.
[14] Z.-H. Zhou, Y. Yu, and C. Qian, *Evolutionary Learning: Advances in Theories and Algorithms*. Berlin, Germany: Springer, 2019.
[15] T. Wu, H. Qian, Z. Liu, J. Zhou, and A. Zhou, "Bi-objective evolutionary Bayesian network structure learning via skeleton constraint," *Frontiers Comput. Sci.*, vol. 17, no. 6, Dec. 2023, Art. no. 176350.
[16] H. Daume III and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Intell. Res.*, vol. 26, pp. 101–126, Jun. 2006.

[17] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21, 2008, pp. 1041–1048.

[18] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[19] T. Long, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Domain adaptation as optimal transport on Grassmann manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7196–7209, Oct. 2023.

[20] Q. Da, Y. Yu, and Z.-H. Zhou, "Learning with augmented class by exploiting unlabeled data," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1760–1766.

[21] Y.-J. Zhang, P. Zhao, L. Ma, and Z.-H. Zhou, "An unbiased risk estimator for learning with augmented classes," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 10247–10258.

[22] Z. Zhang, P. Zhao, Y. Jiang, and Z. Zhou, "Learning with feature and distribution evolvable streams," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 11317–11327.

[23] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Learning with feature evolvable streams," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2602–2615, Jun. 2021.

[24] P. Zhao, Y.-F. Xie, L. Zhang, and Z.-H. Zhou, "Efficient methods for non-stationary online learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 11573–11585.

[25] P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou, "Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization," *J. Mach. Learn. Res.*, vol. 25, no. 98, pp. 1–52, 2024.

[26] L. Ma, Z.-Y. Zhang, Y.-X. Ding, and Z.-H. Zhou, "Handling varied objectives by online decision making," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 2130–2140.

[27] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2292–2300.

[28] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Scaling algorithms for unbalanced optimal transport problems," *Math. Comput.*, vol. 87, no. 314, pp. 2563–2609, Feb. 2018.

[29] G. Peyré, L. Chizat, F.-X. Vialard, and J. Solomon, "Quantum entropic regularization of matrix-valued optimal transport," *Eur. J. Appl. Math.*, vol. 30, no. 6, pp. 1079–1102, Dec. 2019.

[30] G. Peyré and M. Cuturi, "Computational optimal transport: With applications to data science," *Found. Trends Mach. Learn.*, vol. 11, nos. 5–6, pp. 355–607, 2019.

[31] M. M. Drugan and A. Nowé, "Designing multi-objective multi-armed bandits algorithms: A study," in *Proc. 22nd Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8.

[32] Z.-H. Zhou and Z.-H. Tan, "Learnware: Small models do big," *Sci. China Inf. Sci.*, vol. 67, no. 1, Jan. 2024, Art. no. 112102.

[33] Z.-H. Tan et al., "Beimingwu: A learnware dock system," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 5773–5782.

[34] X.-Z. Wu, W. Xu, S. Liu, and Z.-H. Zhou, "Model reuse with reduced kernel mean embedding specification," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 699–710, Jan. 2023.

[35] S.-J. Huang, Y. Yu, and Z.-H. Zhou, "Multi-label hypothesis reuse," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 525–533.

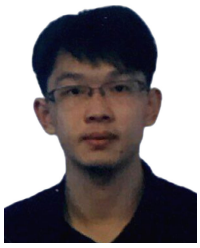
[36] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions," in *Proc. 9th Int. Soc. Music Inf. Retr. (ISMIR)*, 2008, pp. 325–330.

[37] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.

[38] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *J. Mach. Learn. Res.*, vol. 6, pp. 363–392, Apr. 2005.

[39] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1995.

[40] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.



Lanjihong Ma received the B.Sc. degree from Tsinghua University, Beijing, China, in 2017, and the M.Sc. degree from the University of California at San Diego, La Jolla, CA, USA, in 2019. He is currently pursuing the Ph.D. degree with Nanjing University, Nanjing, China, under the supervision of Prof. Zhi-Hua Zhou.

His research interests are mainly on open-environment machine learning, online learning, and preference learning with human feedback.

Yao-Xiang Ding received the B.Sc. degree in mechanical engineering from the University of Science and Technology Beijing, Beijing, China, in 2011, the M.Sc. degree in computer science from Peking University, Beijing, in 2014, and the Ph.D. degree from Nanjing University, Nanjing, China, in 2020.

He is currently a tenure-track Assistant Professor with the State Key Laboratory of Computer Aided Design and Computer Graphics and a member of the GAPS Lab, Zhejiang University, Hangzhou, China. He is currently working on special theories and algorithms of interactive learning, decision-making, and model reuse. His research interests are mainly machine learning and artificial intelligence.

Dr. Ding served as the Publicity Co-Chair for SIAM International Conference on Data Mining (SDM) 2023.

Peng Zhao (Member, IEEE) received the B.Sc. degree from Tongji University, Shanghai, China, in 2016, and the Ph.D. degree from Nanjing University, Nanjing, China, in 2021.

He is currently an Assistant Professor at the School of Artificial Intelligence, Nanjing University. He is now working on open-environment machine learning, online learning, and optimization. He has published over 50 papers in top-tier journals, such as *Journal of Machine Learning Research* (JMLR) and IEEE/ACM Transactions, and conferences, such as International Conference on Machine Learning (ICML), Conference on Neural Information Processing Systems (NeurIPS), and Conference on Learning Theory (COLT). His research interest is mainly in machine learning.

Dr. Zhao served as the Area Chair for ICML and NeurIPS.

Zhi-Hua Zhou (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees (Hons.) in computer science from Nanjing University, Nanjing, China, in 1996, 1998, and 2000, respectively.

He joined Nanjing University as an Assistant Professor in 2001, where he is currently a Professor and the Vice President. He is also the Founding Director of the LAMDA Group. He has authored the books *Ensemble Methods: Foundations and Algorithms*, *Evolutionary Learning: Advances in Theories and Algorithms*, *Machine Learning* and has published more than 200 papers in top-tier international journals or conference proceedings. He holds more than 30 patents. His research interests are mainly in artificial intelligence, machine learning, and data mining.

Dr. Zhou is a fellow of ACM, AAAI, AAAS, IAPR, IET, IEE, CCF, and CAAI. He received various awards/honors, including the National Natural Science Award of China, the IEEE Computer Society Edward J. McCluskey Technical Achievement Award, and the CCF-ACM Artificial Intelligence Award. He is the President of IJCAI Trustee, a Series Editor of *Lecture Notes in Artificial Intelligence* (Springer), an Advisory Board Member of AI Magazine, the Editor-in-Chief of *Frontiers of Computer Science*, the Associate Editor-in-Chief of *Science China Information Sciences*, and an Associate Editor of *Artificial Intelligence* and *Machine Learning*. He founded Asian Conference on Machine Learning (ACML).