

E-MLNet: Enhanced Mutual Learning for Universal Domain Adaptation with Sample-Specific Weighting

Samuel Felipe dos Santos, Tiago Agostinho de Almeida, and Jurandy Almeida

Department of Computing, Federal University of São Carlos (UFSCar), Sorocaba, SP – Brazil

Emails: {samuel.felipe, talmeida, jurandy.almeida}@ufscar.br

Abstract—Universal Domain Adaptation (UniDA) seeks to transfer knowledge from a labeled source to an unlabeled target domain without assuming any relationship between their label sets, requiring models to classify known samples while rejecting unknown ones. Advanced methods like Mutual Learning Network (MLNet) use a bank of one-vs-all classifiers adapted via Open-set Entropy Minimization (OEM). However, this strategy treats all classifiers equally, diluting the learning signal. We propose the Enhanced Mutual Learning Network (E-MLNet), which integrates a dynamic weighting strategy to OEM. By leveraging the closed-set classifier’s predictions, E-MLNet focuses adaptation on the most relevant class boundaries for each target sample, sharpening the distinction between known and unknown classes. We conduct extensive experiments on four challenging benchmarks: Office-31, Office-Home, VisDA-2017, and ImageCLEF. The results demonstrate that E-MLNet achieves the highest average H-scores on VisDA and ImageCLEF and exhibits superior robustness over its predecessor. E-MLNet outperforms the strong MLNet baseline in the majority of individual adaptation tasks—22 out of 31 in the challenging Open-Partial DA setting and 19 out of 31 in the Open-Set DA setting—confirming the benefits of our focused adaptation strategy.

I. INTRODUCTION

Deep learning models have achieved remarkable success, yet their performance often relies on supervised learning, which assumes that training and testing data are drawn from the same distribution and share identical classes. These ideal conditions are rare in real-world applications, which frequently face challenges like costly data annotation and distribution shifts caused by dynamic environments where novel classes can emerge [1], [2].

Unsupervised Domain Adaptation (UDA) offers a strategy to overcome these issues by transferring knowledge from a label-rich source domain to an unlabeled target domain [3]. However, traditional UDA often operates under the restrictive Closed-Set Domain Adaptation (CDA) assumption, where both domains share the same set of classes [4]. As illustrated in Figure 1, real-world scenarios are more complex, including Partial Domain Adaptation (PDA), where the target label set is a subset of the source’s; Open-Set Domain Adaptation (ODA), where the target contains novel classes; and a mixture of both, called Open-Partial Domain Adaptation (OPDA).

Universal Domain Adaptation (UniDA) [5] addresses this challenge by making no assumptions about the label set relationship. A UniDA model must correctly classify target samples from shared classes while simultaneously identifying

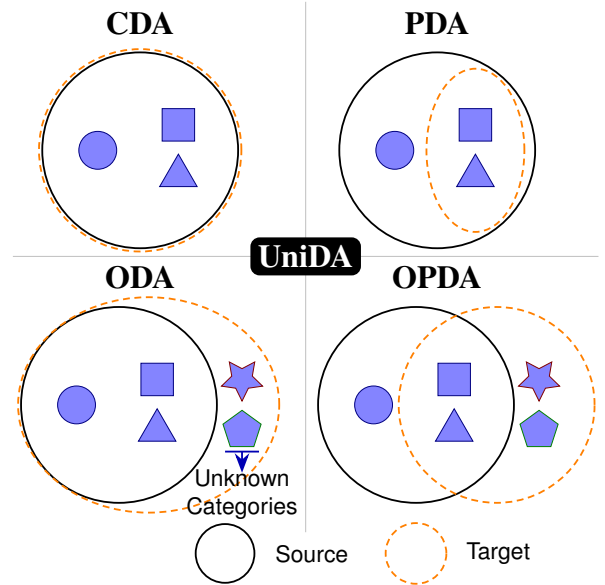


Fig. 1. Different settings for Unsupervised Domain Adaptation (UDA) based on the relationship between the class sets of the source (solid line) and target (dashed line) domains: Closed-set (CDA), Partial (PDA), Open-Set (ODA), and Open-Partial (OPDA). Universal Domain Adaptation (UniDA) generalizes these settings by assuming this relationship is unknown a priori [4].

samples from novel target classes as “unknown”. Recent methods like OVANet [6] and MLNet [7] tackle this by training a bank of one-vs-all (open-set) classifiers. These models learn class-specific decision boundaries and adapt them to the target domain using Open-set Entropy Minimization (OEM). MLNet further enhances this with Neighborhood Invariance Learning (NIL), Cross-domain Manifold Mixup (CMM), and a Consistency Constraint (CC) [7].

In both OVANet and MLNet, OEM is performed by averaging the entropy from all open-set classifiers, implicitly assuming that each one is equally important for any given sample. Often, only a small subset of class boundaries is truly relevant for making an accurate in-lie/out-lie decision. By treating all classifiers equally, the learning signal is diluted by the vast number of irrelevant classifiers, especially as the number of source classes grows. This can lead to suboptimal adaptation and less precise decision boundaries.

Motivated by this observation, we propose the Enhanced Mutual Learning Network (E-MLNet) for UniDA, which

refines the adaptation process of MLNet by integrating a dynamic, sample-specific weighting strategy for OEM. Inspired by [8], E-MLNet leverages the predictions from the closed-set classifier to weight the contribution of each open-set classifier’s entropy term. This dynamic weighting strategy encourages the model to focus on the most informative open-set classifiers for each target sample, thereby sharpening the learning signal and improving adaptation.

The main contributions of this paper are:

- We introduce E-MLNet, which enhances MLNet with a dynamic weighting strategy to enable the OEM to focus adaptation on the most relevant class boundaries for each target sample, improving the model’s robustness and consistency across a wide variety of adaptation tasks.
- We conduct extensive experiments on four challenging and publicly available benchmarks for domain adaptation. Our results demonstrate that E-MLNet performs better than MLNet in most of the evaluated scenarios.

II. RELATED WORK

Universal Domain Adaptation (UniDA) addresses the challenge of transferring knowledge from a labeled source domain to an unlabeled target domain under the assumption that the relationship between their label sets is unknown. Consequently, the source and target domains may share a set of common classes, while each may also possess private classes. An effective UniDA model must not only correctly classify target samples belonging to the common classes but also reject samples from private target classes as “unknown”. A diverse range of strategies has been explored to tackle this problem.

One prevalent approach relies on adversarial training with a domain discriminator to learn domain-invariant features. The Universal Adaptation Network (UAN) [5] pioneered this by employing entropy to weight samples and quantify their ambiguity. CMU [9] extended this idea by using more refined techniques for uncertainty measurement and pseudo-labeling.

Another significant research direction focuses on exploiting the geometric structure of the feature space. Several methods leverage clustering techniques; for instance, Domain Consensus Clustering (DCC) [10] learns robust class representations through consensus. Others, like CPR [11], learn explicit prototypes for common classes and reciprocal points to explicitly model the “unknown” feature space, thereby pushing private target samples away from the known class prototypes.

Self-supervised learning has also emerged as a promising tool for UniDA. DANCE [4] employs self-supervision to cluster target samples while employing entropy minimization to align source and target features and reject unknown instances. Similarly, ROS [12] explored image rotation as a pretext task to help discriminate between known and unknown samples. Inspired by SwAV [13], UniOT [14] formulates UniDA as a unified optimal transport problem, seeking to learn an optimal mapping between target samples and source prototypes.

Differently, some methods integrate multiple strategies, designing sophisticated, multi-component loss functions to address the various facets of the UniDA problem simultaneously.

TNT [15] evaluates sample-level uncertainty with a mutual nearest neighbors contrastive loss. GATE [16] employs geometric adversarial learning for global distribution calibration and subgraph-level contrastive learning for local region aggregation. NCAL [17] employs credibility-weighted conditional adversarial learning to obtain class-invariant features.

Our work belongs to a promising family of methods based on one-vs-all (open-set) classifiers. OVA-Net [6] first introduced this strategy by training a separate binary classifier for each known class, enabling it to learn an adaptive threshold for rejecting unknowns. Building on this, MLNet [7] improves upon OVA-Net with three key components: Neighborhood Invariance Learning (NIL) for a more robust feature representation, Cross-domain Manifold Mixup (CMM) to synthesize features that simulate unknown samples, and a Consistency Constraint (CC) to foster mutual learning between classifiers.

III. OUR APPROACH

This section begins by formally defining the UniDA problem. We then review the architectures and learning objectives of OVA-Net and MLNet, which serve as the foundation for our method. Finally, we detail our enhancement: a weighting strategy for OEM that improves upon the MLNet framework. Figure 2 shows an overview of our proposed approach.

A. Problem Formulation

We are given a labeled source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and an unlabeled target domain $D_t = \{x_j^t\}_{j=1}^{N_t}$. Let L_s and L_t denote the label sets of the source and target domains, respectively. In the UniDA setting, the relationship between L_s and L_t is unknown. The goal is to train a model that can classify target samples x_j^t into one of the known classes present in the source domain (L_s) if $x_j^t \in L_s \cap L_t$, or classify it as “unknown” if $x_j^t \in L_t \setminus L_s$. For simplicity, we denote $K = |L_s|$ as the number of classes in L_s .

B. Preliminaries: From OVA-Net to MLNet

Our method builds directly upon the MLNet framework, which itself extends OVA-Net. The core architecture comprises three components: a feature extractor $\mathcal{F}$, a closed-set classifier $\mathcal{C}$, and a bank of open-set classifiers $\mathcal{O}$. The classifier $\mathcal{C}$ is a standard multi-class classifier trained on the K source classes. The bank of open-set classifiers $\mathcal{O} = \{\mathcal{O}^k\}_{k=1}^K$ consists of K one-vs-all (OVA) binary classifiers. Each classifier $\mathcal{O}^k$ corresponds to a source class k and is trained to distinguish samples of that class (in-liers) from all other classes (out-liers).

OVA-Net Framework. OVA-Net [6] trains its components using two primary constraints on source data. The closed-set classifier $\mathcal{C}$ is trained with a standard cross-entropy loss, $\mathcal{L}_{cls}$. The open-set classifiers $\mathcal{O}$ are trained using Hard Negative Classifier Sampling (HNCS), which encourages each classifier $\mathcal{O}^k$ to learn a tight boundary between its corresponding class k and the most confusing negative class. This loss, $\mathcal{L}_{ova}$, for a source sample (x_i^s, y_i^s) is given by [6]:

$$\mathcal{L}_{ova}(x_i^s, y_i^s) = -\log p_o(y_i^s | x_i^s) - \min_{k \neq y_i^s} \log(1 - p_o(k | x_i^s)), \quad (1)$$

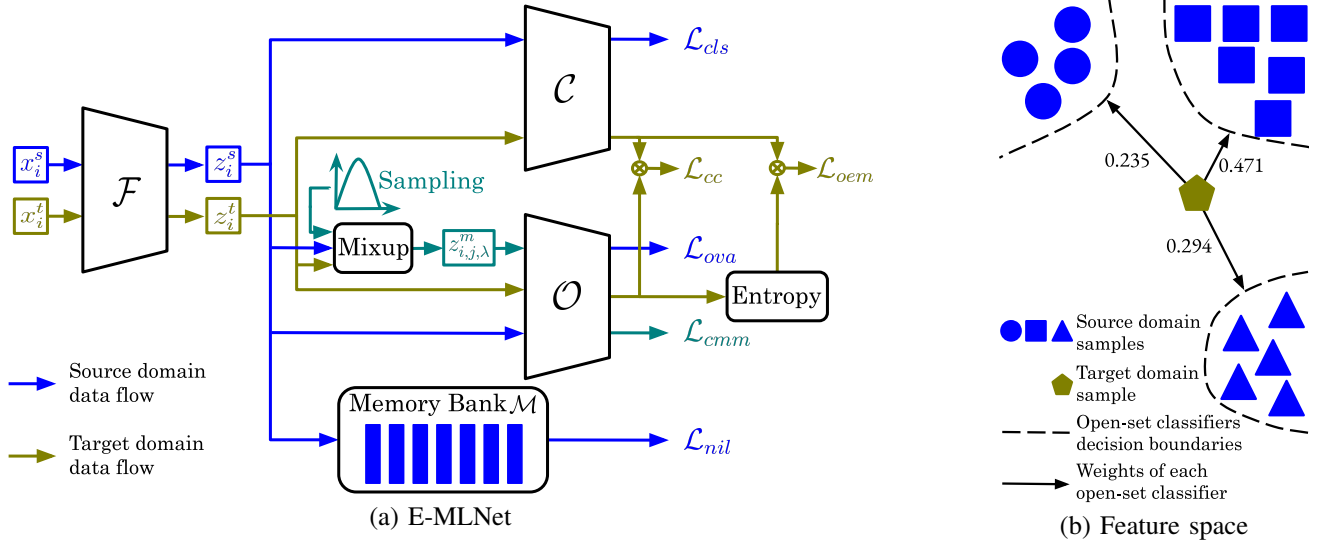


Fig. 2. Overview of E-MLNet. (a) Schematic of our proposed E-MLNet, highlighting the data flow for the source and target domains and the application of loss functions. (b) Illustration of the feature space with one-vs-all decision boundaries. For a given target sample, its proximity to different class boundaries varies, making some classifiers more relevant than others. Following [8], E-MLNet leverages the closed-set classifier’s predictions to weight the importance of each open-set classifier during adaptation.

where $p_o(k|x_i^s)$ is the score from the k -th open-set classifier, $\mathcal{O}^k$, indicating the probability that x_i^s is an in-liner for class k . For adaptation, OVANet uses Open-set Entropy Minimization (OEM) on target samples to encourage confident predictions from the open-set classifiers [6]:

$$\mathcal{L}_{oem}(x_j^t) = -\frac{1}{K} \sum_{k=1}^K [p_o(k|x_j^t) \log p_o(k|x_j^t) + (1 - p_o(k|x_j^t)) \log(1 - p_o(k|x_j^t))]. \quad (2)$$

MLNet Extensions. MLNet [7] enhances OVANet with three additional constraints.

- 1) **Neighborhood Invariance Learning ($\mathcal{L}_{nil}$):** This constraint reduces intra-domain variations in the target feature space. For a target sample x_j^t , it encourages its feature representation to be similar to its neighbors N_j found in a memory bank $\mathcal{M}$ and is given by [7]:

$$\mathcal{L}_{nil}(x_j^t) = -\frac{1}{|N_j|} \sum_{k \in N_j} w_{jk} \log p_{jk}, \quad (3)$$

where p_{jk} is the softmax-normalized similarity (i.e., softmax of a dot product) between the l_2 -normalized features of $z_j^t = \mathcal{F}(x_j^t)$ and its neighbor $z_k^t = \mathcal{F}(x_k^t)$, and $w_{jk} = \frac{|N_j \cap N_k|}{|N_j \cup N_k|}$ is a confidence weight based on the Jaccard similarity of their respective neighborhoods. By pulling a sample’s feature representation closer to its high-confidence neighbors, which are likely to share the same class, the feature cluster for that class becomes tighter, making classes more separable.

- 2) **Cross-domain Manifold Mixup ($\mathcal{L}_{cmm}$):** This constraint simulates “unknown” samples by creating a mixed feature $z_{i,j,\lambda}^m = \lambda z_i^s + (1 - \lambda) z_j^t$ through a linear combination of a source feature $z_i^s = \mathcal{F}(x_i^s)$ and a target

feature $z_j^t = \mathcal{F}(x_j^t)$ with a weight $\lambda \sim \text{Beta}(\alpha, \alpha)$, $\alpha = 2.0$. The act of mixing features across different domains is sufficient to generate a challenging out-of-distribution sample for the open-set classifier related to the source class y_i^s , which is then trained to reject this sample [7]:

$$\mathcal{L}_{cmm}(x_i^s, y_i^s, x_j^t) = -\log(1 - p_o(y_i^s | z_{i,j,\lambda}^m)). \quad (4)$$

- 3) **Consistency Constraint ($\mathcal{L}_{cc}$):** This constraint counteracts the tendency of $\mathcal{L}_{cmm}$ to misclassify knowns as unknown. It identifies target samples where the closed-set confidence, $p_c(k|x_j^t)$, is high, but the open-set score, $p_o(k|x_j^t)$, is low. The loss is defined as [7]:

$$\mathcal{L}_{cc}(x_j^t) = -\frac{1}{K} \sum_{k=1}^K p_c(k|x_j^t) \cdot p_o(k|x_j^t). \quad (5)$$

This pushes the open-set score $p_o(k|x_j^t)$ to increase by a magnitude proportional to the closed-set confidence $p_c(k|x_j^t)$, thus correcting misidentified known samples. The overall training loss of MLNet combines all of these components, including the $\mathcal{L}_{cls}$ and $\mathcal{L}_{oem}$ constraints of OVANet.

C. Enhanced MLNet

The standard OEM constraint, $\mathcal{L}_{oem}$, averages the binary entropy over all K open-set classifiers, $\mathcal{O}$, assuming that all source classes are equally relevant for any given target sample. This overlooks the fact that a sample’s feature representation will be close to some decision boundaries but far from others, thereby diluting the learning signal by including irrelevant classifiers in the adaptation.

Motivated by this observation, we propose to enhance MLNet by using the predictions of the closed-set classifier $\mathcal{C}$ as dynamic, sample-specific weights for the open-set entropy terms. This allows the model to focus adaptation on the most

relevant classes. Inspired by [8], we reformulate the $\mathcal{L}_{oem}$ constraint for a target sample x_j^t as:

$$\mathcal{L}_{oem}(x_j^t) = -\frac{1}{K} \sum_{k=1}^K p_c(k|x_j^t) \cdot [p_o(k|x_j^t) \log p_o(k|x_j^t) + (1 - p_o(k|x_j^t)) \log(1 - p_o(k|x_j^t))], \quad (6)$$

where $p_c(k|x_j^t)$ is its prediction probability for class k from the closed-set classifier $\mathcal{C}$. This formulation acts as a dynamic weighting strategy. If the closed-set classifier $\mathcal{C}$ is confident that x_j^t likely belongs to class k , its prediction $p_c(k|x_j^t)$ will be high. Consequently, the loss will be dominated by the term that encourages a confident in-lier/out-lier decision from the corresponding open-set classifier, $\mathcal{O}^k$. Entropy terms for other, less relevant classifiers will be down-weighted, preventing them from diluting the adaptation signal.

Following MLNet, the overall loss $\mathcal{L}_{all}$ is a weighted sum of all components [7]:

$$\begin{aligned} \mathcal{L}_{all} = & \mathbb{E}_{(x_i^s, y_i^s) \sim D_s} [\mathcal{L}_{cls}(x_i^s, y_i^s) + \mathcal{L}_{ova}(x_i^s, y_i^s)] \\ & + \mathbb{E}_{x_j^t \sim D_t} [\beta_1 \mathcal{L}_{nil}(x_j^t) + \eta \mathcal{L}_{cc}(x_j^t) + \gamma \mathcal{L}_{oem}(x_j^t)] \quad (7) \\ & + \mathbb{E}_{(x_i^s, y_i^s) \sim D_s, x_j^t \sim D_t} [\beta_2 \mathcal{L}_{cmm}(x_i^s, y_i^s, x_j^t)], \end{aligned}$$

where β_1, β_2, η , and γ are hyperparameters that balance the contributions of the different loss components. For a fair comparison with MLNet, we adopt their settings for β_1, β_2 , and η , and set γ following OVANet.

IV. EXPERIMENTS AND RESULTS

This section presents our experiments and their results. We first detail the experimental setup and then report results for different UniDA settings on four well-known benchmarks.

A. Datasets

Our experiments are conducted on four widely-used domain adaptation benchmarks.

- **Office-31** [18] is a standard, medium-scale dataset containing 4,110 images across 31 categories, collected from three distinct domains: Amazon (A), DSLR (D), and Webcam (W).
- **Office-Home** [19] is a more challenging dataset comprising 15,500 images across 65 categories. It features four domains with significant visual shifts: Art (A), Clipart (C), Product (P), and Real-World (R).
- **VisDA-2017** [20] is a large-scale benchmark for simulation-to-real adaptation, consisting of 152,397 synthetic images and 55,388 real-world images distributed across 12 categories.
- **ImageCLEF**¹ is a dataset of 2,400 images across 12 common categories, sourced from four publicly available datasets: Bing (B), Caltech-256 (C), ImageNet (I), and PASCAL VOC-2012 (P).

¹<http://imageclef.org/2014/adaptation/> (As of September 10, 2025)

We follow the standard UniDA protocols established in prior work [5], [7] and evaluate on various category-shift scenarios. Due to space constraints, we focus our reporting on the OPDA and ODA settings. We omit CDA and PDA results, as our proposed modification to the OEM loss does not affect that specific scenario. For fair comparison, the class splits for each scenario are kept consistent with prior work [5], [7]. In the results section, each table specifies the class split using the format: $|L_s \cap L_t| / |L_s \setminus L_t| / |L_t \setminus L_s|$ (shared / source-private / target-private classes).

B. Evaluation Metrics

In UniDA, a model's performance must reflect a balance between correctly classifying known-class samples and accurately identifying unknown ones. A single accuracy metric can therefore be misleading. We adopt the standard metric for this task, the H-score (HSC), which is the harmonic mean of the known-class and unknown-class accuracies [12]:

$$\text{HSC} = \frac{2 \times \text{OS}^* \times \text{UNK}}{\text{OS}^* + \text{UNK}}. \quad (8)$$

where OS* is the average per-class accuracy on the set of shared categories ($L_s \cap L_t$) and UNK is the accuracy on the unknown categories ($L_t \setminus L_s$). This metric yields a high score only when both of its components are high, providing a balanced measure of performance on the UniDA task.

C. Implementation Details

For a fair comparison, our implementation² is built directly upon the official MLNet framework³. We use a ResNet-50 model pre-trained on ImageNet as the backbone feature extractor $\mathcal{F}$, replacing its final fully-connected layer with new classification heads. The network is trained using an SGD optimizer with Nesterov momentum of 0.9 and a weight decay of 5×10^{-4} . We adopt an inverse learning rate decay schedule, with an initial learning rate of 1×10^{-3} for the backbone and 1×10^{-2} for the new layers. All experiments were run for 50 epochs with a batch size of 36.

The hyperparameters for the MLNet loss components were set according to the original study to ensure a direct comparison [7]: $\beta_1 = 0.5$ for $\mathcal{L}_{nil}$, $\beta_2 = 0.1$ for $\mathcal{L}_{cmm}$, and $\eta = 0.16$ for $\mathcal{L}_{cc}$ (doubled for the VisDA-2017 dataset). For the OEM loss hyperparameter, γ , we follow OVANet [6] and set it to 0.1 across all experiments.

All experiments were conducted on a single machine with an AMD Ryzen Threadripper PRO 5975WX 32-core CPU, 256 GB of DDR4 memory, and four NVIDIA RTX 5000 Ada Generation GPUs. The system runs Ubuntu 22.04.4 LTS with Linux kernel 6.8.0 and the ext4 file system.

D. Results on UniDA

We first evaluate all methods in the OPDA setting, the most challenging UniDA scenario, which includes both source-private and target-private classes. The results are presented in

²Our code is available at <https://github.com/jurandy-almeida/E-MLNet>

³<https://github.com/YanzuoLu/MLNet> (Accessed: September 10, 2025)

TABLE I
H-SCORE (%) COMPARISON IN THE OPDA SETTING FOR OFFICE-31, OFFICE-HOME, AND VISDA.

Method	Office-31 (10/10/11)							Office-Home (10/5/50)													VisDA (6/3/3)
	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	
UAN [5]	59.7	58.6	60.1	70.6	60.3	71.4	63.5	51.6	51.7	54.3	61.7	57.6	61.9	50.4	47.6	61.5	62.9	52.6	65.2	56.6	30.5
CMU [9]	68.1	67.3	71.4	79.3	72.2	80.4	73.1	56.0	56.6	59.2	67.0	64.3	67.8	54.7	51.1	66.4	68.2	57.9	69.7	61.6	34.6
DANCE [4]	79.6	75.8	82.9	90.9	77.6	87.1	82.3	61.0	60.4	64.9	65.7	58.8	61.8	73.1	61.2	66.6	67.7	62.4	63.7	63.9	42.8
DCC [10]	88.5	78.5	70.2	79.3	75.9	88.6	80.2	58.0	54.1	58.0	74.6	70.6	77.5	64.3	73.6	74.9	81.0	75.1	80.4	70.2	43.0
GATE [16]	87.7	81.6	84.2	94.8	83.4	94.1	87.6	63.8	75.9	81.4	74.0	72.1	79.8	74.7	70.3	82.7	79.1	71.5	81.7	75.6	56.4
TNT [15]	85.7	80.4	83.8	92.0	79.1	91.2	85.4	61.9	74.6	80.2	73.5	71.4	79.6	74.2	69.5	82.7	77.3	70.1	81.2	74.7	55.3
UniOT [14]	87.0	88.5	88.4	98.8	87.6	96.6	91.2	67.3	80.5	86.0	73.5	77.3	84.3	75.5	63.3	86.0	77.8	65.4	81.9	76.6	57.3
CPR [11]	84.4	81.4	85.5	93.4	91.3	96.8	88.8	59.0	77.1	83.7	69.7	68.1	75.4	74.6	56.1	78.9	80.5	63.0	81.0	72.3	58.2
NCAL [17]	85.3	85.3	88.0	94.0	87.9	95.5	89.3	59.1	88.3	87.3	72.1	73.2	81.0	76.3	57.4	88.4	81.1	62.0	85.4	75.9	62.9
OVANet [6]	83.9	78.5	81.3	95.4	83.2	96.4	86.5	62.2	79.0	80.0	69.2	70.5	76.5	70.9	59.5	80.9	76.8	62.8	79.6	72.3	49.6
MLNet [7]	90.4	93.7	89.7	96.2	88.4	98.3	92.8	68.2	83.8	85.0	73.6	78.2	82.2	75.2	64.7	85.1	78.8	69.9	83.9	77.4	69.9
E-MLNet	91.8	91.8	89.4	98.3	88.2	96.5	92.6	68.6	82.8	84.2	73.8	78.2	82.2	75.4	65.2	84.4	77.8	69.6	84.0	77.2	70.3

TABLE II
H-SCORE (%) COMPARISON IN THE OPDA SETTING FOR IMAGECLEF.

Method	ImageCLEF (6/3/3)												
	B2C	B2I	B2P	C2B	C2I	C2P	I2B	I2C	I2P	P2B	P2C	P2I	Avg
OVANet [6]	68.9	70.4	63.4	62.0	81.3	74.3	59.5	73.4	66.8	54.4	83.1	66.9	68.7
MLNet [7]	<u>96.7</u>	<u>77.0</u>	<u>64.6</u>	<u>68.8</u>	<u>87.7</u>	<u>66.6</u>	<u>65.9</u>	<u>93.7</u>	<u>80.1</u>	<u>64.7</u>	<u>89.8</u>	<u>81.8</u>	<u>78.1</u>
E-MLNet	97.3	79.5	66.1	70.6	89.0	69.3	67.1	94.3	81.4	64.9	90.2	88.3	79.8

TABLE III
H-SCORE (%) COMPARISON IN THE ODA SETTING FOR OFFICE-31, OFFICE-HOME, AND VISDA.

Method	Office-31 (10/0/11)							Office-Home (25/0/40)													VisDA (6/0/6)
	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	
UAN [5]	38.9	46.8	68.0	68.8	54.9	53.0	55.1	40.3	41.5	46.1	53.2	48.0	53.7	40.6	39.8	52.5	53.6	43.7	56.9	47.5	51.9
CMU [9]	52.6	55.7	76.5	75.9	65.8	64.7	65.2	45.1	48.3	51.7	58.9	55.4	61.2	46.5	43.8	58.0	58.6	50.1	61.8	53.3	54.2
DANCE [4]	84.9	78.8	79.1	78.8	68.3	88.9	79.8	61.9	61.3	63.7	64.2	58.6	62.6	67.4	61.0	65.5	65.9	61.3	64.2	63.1	67.5
DCC [10]	58.3	54.8	67.2	89.4	85.3	80.9	72.7	56.1	67.5	66.7	49.6	66.5	64.0	55.8	53.0	70.5	61.6	57.2	71.9	61.7	59.6
GATE [16]	88.4	86.5	84.2	95.0	86.1	96.7	89.5	63.8	70.5	75.8	66.4	67.9	71.7	67.3	61.5	76.0	70.4	61.8	75.1	69.0	70.8
TNT [15]	85.8	82.3	80.7	91.2	81.5	96.2	86.3	63.4	67.9	74.9	65.7	67.1	68.3	64.5	58.1	73.2	67.8	61.9	74.5	67.3	71.6
NCAL [17]	84.0	93.4	93.4	85.4	89.0	87.2	88.7	64.2	74.1	80.5	68.1	72.5	77.0	66.9	58.1	79.1	74.6	63.5	79.6	71.5	69.1
OVANet [6]	88.2	88.7	86.9	97.8	89.4	98.8	91.6	58.7	66.5	70.4	61.5	65.4	68.4	60.4	53.5	70.0	68.2	59.1	67.0	64.1	61.6
MLNet [7]	93.0	91.9	86.9	98.1	87.5	99.5	92.8	61.3	69.9	74.4	63.1	68.2	70.4	62.0	59.9	72.4	69.1	62.6	71.1	67.0	63.9
E-MLNet	93.6	92.4	88.6	98.2	87.9	98.6	93.2	61.6	69.2	73.1	63.0	67.6	70.2	61.7	59.2	71.9	68.3	62.1	70.6	66.6	66.6

TABLE IV
H-SCORE (%) COMPARISON IN THE ODA SETTING FOR IMAGECLEF.

Method	ImageCLEF (6/0/6)												
	B2C	B2I	B2P	C2B	C2I	C2P	I2B	I2C	I2P	P2B	P2C	P2I	Avg
OSNN [21]	74.9	69.1	63.5	54.9	60.4	57.3	49.8	57.0	54.2	52.2	67.3	64.1	60.4
STA [22]	66.5	71.2	59.8	65.2	77.2	65.7	57.9	68.4	68.2	51.0	63.2	65.1	65.0
OSBP [23]	83.9	74.3	66.5	59.9	84.3	66.7	58.1	86.3	70.1	56.3	78.9	72.6	71.5
DAOD [24]	80.7	84.3	76.3	49.1	83.6	76.7	55.7	81.2	76.9	51.3	80.5	83.9	73.3
ROS [12]	83.8	74.6	62.9	63.3	80.6	73.3	58.8	90.6	77.0	52.7	79.7	80.5	73.1
OVANet [6]	74.1	79.6	68.7	66.5	86.3	76.8	64.9	86.0	74.2	55.7	65.7	61.5	71.7
MLNet [7]	<u>88.8</u>	<u>78.6</u>	<u>69.0</u>	<u>59.6</u>	82.4	63.8	60.4	<u>95.6</u>	<u>80.3</u>	<u>58.9</u>	<u>89.1</u>	<u>86.3</u>	<u>76.1</u>
E-MLNet	89.1	80.4	70.1	61.6	<u>85.1</u>	<u>65.4</u>	<u>62.2</u>	96.3	81.3	60.4	89.8	87.3	77.4

Tables I and II. For the family of methods based on one-vs-all classifiers (OVANet, MLNet, and E-MLNet), the best results are in **bold** and the second-best ones are in underline.

Our analysis first focuses on the family of one-vs-all methods: OVANet, MLNet, and our E-MLNet. Both MLNet and E-MLNet significantly outperform OVANet across all datasets, with average H-score gains of at least 6.1% on Office-31, 4.9% on Office-Home, 20.3% on VisDA, and 9.4% on ImageCLEF.

The primary comparison is between E-MLNet and its direct predecessor, MLNet. While both methods achieve similar

average H-scores, E-MLNet demonstrates superior robustness, outperforming MLNet in 22 of the 31 individual adaptation tasks. This highlights the benefit of using the weighted strategy: by dynamically focusing the adaptation on relevant class boundaries, E-MLNet achieves more consistent performance across a wider range of scenarios—a critical attribute for a truly “universal” method. Notably, E-MLNet obtains the highest average H-score on the challenging VisDA and ImageCLEF benchmarks, underscoring the effectiveness of our approach. We hypothesize that the few cases where MLNet performs

better occur in tasks with severe domain shifts where the closed-set classifier’s initial predictions are less reliable. In such cases, uniform weighting of the standard OEM loss is a ‘safer’ albeit less efficient strategy.

Next, we analyze the ODA setting, where the target domain contains unknown classes but all source classes are shared. The results are detailed in Tables III and IV.

E-MLNet and MLNet again surpass OVA-Net on all datasets, confirming the strength of the MLNet framework. Crucially, even without source-private classes, E-MLNet maintains an edge in consistency, achieving a higher H-score than MLNet in 19 out of 31 tasks. This result confirms that our focused adaptation strategy is beneficial even when the open-set recognition task is simpler, leading to more robust decision boundaries. When compared to all methods, the OVA-based family is highly competitive, though methods like GATE and NCAL also show strong results on the Office-Home dataset.

V. CONCLUSION

In this paper, we proposed the Enhanced Mutual Learning Network (E-MLNet), a method that refines the adaptation process of MLNet by integrating a dynamic, sample-specific weighting strategy for OEM. Inspired by [8], E-MLNet employs the predictions from the closed-set classifier to modulate the entropy contribution of each open-set classifier. This simple yet effective modification sharpens the learned feature representations, improving the model’s ability to distinguish between known shared classes and unknown private classes in the target domain. Our extensive experiments on four challenging benchmarks demonstrated the tangible benefits of this approach. E-MLNet showed superior robustness and consistency over its strong baseline, outperforming MLNet in the majority of individual tasks (22 of 31 in the OPDA setting and 19 of 31 in the ODA setting) and achieved the highest average H-scores on VisDA and ImageCLEF.

Despite its good performance, E-MLNet has limitations. Its effectiveness is highly dependent on the accuracy of the closed-set classifier. In cases of severe domain shift where the closed-set classifier’s initial predictions are unreliable, our method could potentially focus adaptation on incorrect class boundaries, leading to error propagation. This dependency suggests that performance could be hindered in the very early stages of training before the model has stabilized.

For future work, this dynamic weighting strategy could be explored further. Mitigating the identified limitations by introducing a confidence threshold for applying the weights, or developing more robust guidance mechanisms that are less sensitive to initial classification errors, presents a promising research direction. In addition, strategies like Outlier Exposure [25] could be explored as an alternative to OEM.

ACKNOWLEDGMENT

This research was supported by São Paulo Research Foundation - FAPESP (grants 2023/17577-0, 2024/04500-2, and 2024/22985-3) and National Council for Scientific and Technological Development - CNPq (grants 315220/2023-6, 420442/2023-5, and 444982/2024-8).

REFERENCES

- [1] L. F. A. Silva, N. Sebe, and J. Almeida, “Tightening classification boundaries in open set domain adaptation through unknown exploitation,” in *Conf. Graphics, Patterns and Images – SIBGRAPI*, 2023, pp. 157–162.
- [2] L. F. A. Silva, S. F. dos Santos, N. Sebe, and J. Almeida, “Beyond the known: Enhancing open set domain adaptation with unknown exploration,” *Pattern Recognit. Lett.*, vol. 189, pp. 265–272, 2025.
- [3] L. F. A. Silva, D. C. G. Pedronette, F. A. Faria, J. P. Papa, and J. Almeida, “Improving transferability of domain adaptation networks through domain alignment layers,” in *Conf. Graphics, Patterns and Images – SIBGRAPI*, 2021, pp. 168–175.
- [4] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, “Universal domain adaptation through self supervision,” *NeurIPS*, vol. 33, pp. 16282–16292, 2020.
- [5] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Universal domain adaptation,” in *CVPR*, 2019, pp. 2720–2729.
- [6] K. Saito and K. Saenko, “Ovanet: One-vs-all network for universal domain adaptation,” in *ICCV*, 2021, pp. 9000–9009.
- [7] Y. Lu, M. Shen, A. J. Ma, X. Xie, and J.-H. Lai, “Mlnet: Mutual learning network with neighborhood invariance for universal domain adaptation,” in *AAAI*, vol. 38, no. 4, 2024, pp. 3900–3908.
- [8] G. Yu, J. Zhu, J. Yao, and B. Han, “Self-calibrated tuning of vision-language models for out-of-distribution detection,” *NeurIPS*, vol. 37, pp. 56322–56348, 2024.
- [9] B. Fu, Z. Cao, M. Long, and J. Wang, “Learning to detect open classes for universal domain adaptation,” in *ECCV*, 2020, pp. 567–583.
- [10] G. Li, G. Kang, Y. Zhu, Y. Wei, and Y. Yang, “Domain consensus clustering for universal domain adaptation,” in *CVPR*, 2021, pp. 9757–9766.
- [11] S. Hur, I. Shin, K. Park, S. Woo, and I. S. Kweon, “Learning classifiers of prototypes and reciprocal points for universal domain adaptation,” in *WACV*, 2023, pp. 531–540.
- [12] S. Bucci, M. R. Lohmani, and T. Tommasi, “On the effectiveness of image rotation for open set domain adaptation,” in *ECCV*, 2020, pp. 422–438.
- [13] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *NeurIPS*, vol. 33, pp. 9912–9924, 2020.
- [14] W. Chang, Y. Shi, H. Tuan, and J. Wang, “Unified optimal transport framework for universal domain adaptation,” *NeurIPS*, vol. 35, pp. 29512–29524, 2022.
- [15] L. Chen, Y. Lou, J. He, T. Bai, and M. Deng, “Evidential neighborhood contrastive learning for universal domain adaptation,” in *AAAI*, vol. 36, no. 6, 2022, pp. 6258–6267.
- [16] —, “Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation,” in *CVPR*, 2022, pp. 16134–16143.
- [17] W. Su, Z. Han, R. He, B. Wei, X. He, and Y. Yin, “Neighborhood-based credibility anchor learning for universal domain adaptation,” *Pattern Recognition*, vol. 142, p. 109686, 2023.
- [18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *ECCV*, 2010, pp. 213–226.
- [19] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, “Deep hashing network for unsupervised domain adaptation,” in *CVPR*, 2017, pp. 5018–5027.
- [20] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, “Visda: The visual domain adaptation challenge,” *CoRR*, abs/1710.06924, 2017.
- [21] P. R. Mendes Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. De Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha, “Nearest neighbors distance ratio open-set classifier,” *Machine Learning*, vol. 106, no. 3, pp. 359–386, 2017.
- [22] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, “Separate to adapt: Open set domain adaptation via progressive separation,” in *CVPR*, 2019, pp. 2927–2936.
- [23] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, “Open set domain adaptation by backpropagation,” in *ECCV*, 2018, pp. 153–168.
- [24] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, “Open set domain adaptation: Theoretical bound and algorithm,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4309–4322, 2020.
- [25] D. Hendrycks, M. Mazeika, and T. Dietterich, “Deep anomaly detection with outlier exposure,” in *ICLR*, 2019.