

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

UES: AN ULTRA-EXPANDED SPACE FOR UNSUPER- VISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised Domain Adaptation (UDA) offers a promising solution to address label annotation costs and dataset bias by facilitating knowledge transfer from a label-rich source domain to a related but unlabeled target domain. While the FC+Softmax+Cross Entropy loss has become the de facto standard for classification under the IID assumption, its performance degrades significantly under UDA’s non-IID setting, where target domain features frequently violate decision boundaries, resulting in inter-class confusion. **To overcome this limitation, we propose an innovative Distance Margin-based Ultra-Expanded Space (UES) loss, which encourages features to occupy an expanded representation space, thereby maintaining a safer distance from decision boundaries. Designed as a plug-and-play regularization term, UES can be seamlessly integrated into various classification-based UDA frameworks, offering exceptional simplicity by requiring only few lines of code and minimal hyperparameter tuning while reducing computational overhead. Extensive experiments demonstrate that our method achieves performance improvements in nearly all tested cross-domain tasks.**

1 INTRODUCTION

Deep neural networks that rely on large-scale, well annotated training datasets have achieved significant success. In practical applications, sufficient training data is often time-consuming and costly, and due to domain transfer issues, the generalization ability to new datasets is poor (Pan & Yang, 2009). Therefore, Unsupervised Domain Adaptation (UDA) has emerged as a crucial research direction, focusing on leveraging knowledge from label rich source domains to facilitate learning of relevant but unlabeled target domains (Li et al., 2021b).

UDA has seen major progress in **theory** and algorithms, with most feature distribution alignment methods (Wang & Deng, 2018; Zhang et al., 2019; Liu et al., 2022; Li et al., 2020) following a common three-part architecture in Figure 1. The core component is the feature extractor, which learns domain-invariant representations. The source classifier (the last layer is FC with softmax) handles supervised prediction for the source domain. The domain confuser module implements the alignment strategy to reduce domain differences. During training, the feature extractor processes source data for source classifier while feeding both source and target features into the domain confuser. Through backpropagation, this dual-path setup helps the model learn transferable representations for target domain.

We note two key issues.

The first **challenge**. Target domain features exhibit significantly greater intra-class variation compared to source domain features in the first column of the Figure 2, leading to decision boundary violations and class confusion due to higher epistemic uncertainty in OOD data (Vu et al., 2019). The conventional **margin loss (FC+Softmax+Cross Entropy loss)** module effectively maximizes source

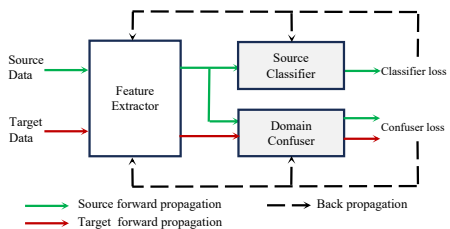


Figure 1: The generalized architecture for UDA. Different confusers represent different alignment algorithms.

domain log-likelihood in the probability space but fails to offer the generalization feature space required for target domain classification. Based on the foundation established in FC+Softmax+Cross Entropy loss, the enhanced margin loss (such as ArcFace (Deng et al., 2019) and Center loss (Wen et al., 2016)) is proposed, which provides a solution by enhancing intra-class compactness and inter-class separability. However, these supervised and IID margin methods does not provide a broad space for target domain features in Figure 2 and promotes the occurrence of negative transfer phenomena (Jiang et al., 2023; Zhang et al., 2022). Although margin methods shows promising performance in IID, its potential have never blossomed in the UDA.

The second challenge. There is an asymmetric determination mechanism: source feature distribution emerge from the combined parameters influence of feature extractor, source classifier, and domain confuser, while target lack classifier parameters guidance (Li et al., 2024).

Our core contribution lies in enhancing feature discriminability through positioning features away from decision boundaries to prevent Inter-class confusion, while comprehensively exploiting the parametric representation capacity of classifiers. This work first derives a Distance Margin from the geometric properties of the FC+Softmax+Cross-Entropy framework, then formulates the Ultra-Expanded Space (UES) loss as a theoretically-grounded regularization term pushing the target domain error further below the generalization bound. The proposed UES loss is universally applicable to any classification paradigm employing linear classifiers—a design pattern ubiquitous across diverse domains, with typical applications including semantic segmentation (pixel-wise classification), object detection (region-based classification) and language modeling (token-level classification) (Azuma et al., 2023).

Our contributions are summarized as follows:

- We propose a margin-based UES loss, representing the first margin method specifically designed for Unsupervised Domain Adaptation (UDA). It constructs an ultra-expanded feature space that effectively pushes features away from decision boundaries, thereby preventing inter-class feature boundary violations.
- The proposed UES loss serves as a simple yet plug-and-play regularization term that can be seamlessly integrated into existing UDA classification frameworks. It requires minimal implementation effort (just few lines of code), demands simplified hyperparameter tuning, and introduces lower training overhead by enhancing computational graph utilization efficiency.
- Extensive experiments demonstrate that the UES regularization term achieves performance improvements across diverse cross-domain tasks, exhibiting remarkable generalization capability.

2 RELATED WORK

UDA Distribution Alignment. UDA distribution alignment methods can be categorized into three primary strategies based on their operational spaces: Input space, the performance degradation on the target domain primarily stems from the distribution shift between the source and target data. Cross-domain data augmentation addresses this issue by synthesizing target-style representations from source samples(Hoffman et al., 2018; Li et al., 2021a; Chen et al., 2025; Dong et al., 2025); Output space, developed from self-supervised learning and self-training to provide additional supervisory signals (Mei et al., 2020; Liu et al., 2021; Li et al., 2022; Yuan et al., 2024; Ali et al., 2025); and Feature space-currently the most prevalent due to their simplicity and efficiency-focusing on

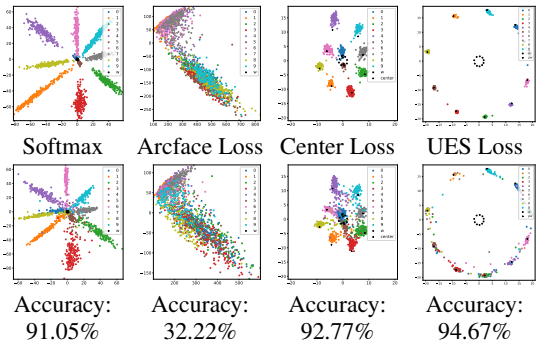


Figure 2: Feature visualization of source domain (MNIST, top) and target domain (USPS, bottom) under different margin losses and provide the accuracy of the target domain. Specifically, we appended a 2D bottleneck layer after the feature extractor and presented it by class. Softmax indicates only using UDA algorithm.

feature distribution alignment (Wu et al., 2021; Wei et al., 2021; Dan et al., 2023; Zhou et al., 2023; Fang et al., 2024; Na et al., 2025), with the first two approaches can be combined with this method (Shen et al., 2023).

Generalization Error Bound. Theoretical frameworks that constrain target domain error within generalization bounds have driven significant algorithmic advances. Ben-David et al. (2006) established the first domain adaptation bound, which Ben-David et al. (2010) extended through analysis in a symmetric hypothesis space. Zhao et al. (2019) demonstrated that algorithms learning domain-invariant features based on such theories may fail in certain scenarios, and proposed an alternative bound based on label function discrepancy. Addressing the gap between binary theory and multi-class practice, Zhang et al. (2019) developed a margin loss-based bound. In source-free domain adaptation, Gezheng et al. (2024) applied contrastive learning to establish a unified upper bound for sample discrepancy metrics on outlier-robust risk. Shen et al. (2023) introduced a bias-variance trade-off approach to control generalization bounds in multi-source-free domain adaptation.

Margin Loss. In classification problems, margin refers to a function that characterizes the confidence disparity of predictions between different classes, and margin loss denotes an optimizable objective function that governs this margin. Among various approaches, the combination of FC+Softmax and Cross Entropy loss has emerged as the most widely adopted and effective paradigm. The former components quantify inter-class probability differences, whereas the latter optimizes these differences. To further enhance intra-class compactness and inter-class separability within this FC+Softmax+Cross-Entropy loss, two major methodological directions have been developed. First, the weight matrices of the FC layer and features are normalized and inner-producted to obtain the angular cosine, which is made directionally more discriminative by adjusting the angular margin of the different categories (Liu et al., 2017; Wang et al., 2018b;a; Kim et al., 2022; Xu et al., 2024). Second, a center is learned for each category, and all the features of each category are pulled towards the corresponding category center based on the Euclidean metric (Wan et al., 2018; Pang et al., 2019; Zhang et al., 2020; Yang et al., 2022; Peng et al., 2025). These methods fail to provide features with a more expansive representation space, consequently limiting their effectiveness in enhancing UDA performance.

3 PRELIMINARIES

This section will introduce the basic notations in UDA classification problems and a generalization bound.

Symbol. In UDA, there are two domains accessible: a labeled source domain containing N_s samples, denoted as $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where $y_i^s \in \{1, 2, \dots, K\}$ represents the label of the i -th source sample x_i^s ; and an unlabeled target domain comprising N_t samples, represented as $\mathcal{T} = \{(x_i^t)\}_{i=1}^{N_t}$. While both domains share the same label space, they exhibit divergence in their underlying probability distributions. This distributional discrepancy typically leads to performance degradation when a model trained solely on the source domain is directly applied to the target domain.

Margin generalization boundary. Zhang et al. (2019) presents a generalization bound based on the margin loss, extending previous bounds that were confined to binary classification to practical multi-class problems. The expected error $err_{\mathcal{T}}(f)$ on the target domain is bounded by the sum of four terms: the empirical error $err_{\mathcal{S}}^{(\rho)}(f)$ on the source domain, the empirical margin disparity discrepancy (MDD) $d_{f, \mathcal{F}}^{(\rho)}(\hat{\mathcal{S}}, \hat{\mathcal{T}})$, an ideal empirical error term λ^* and a complexity term. It is assumed that with a suitable architecture or design, the latter two terms can be made sufficiently small.

$$err_{\mathcal{T}}(f) \leq err_{\mathcal{S}}^{(\rho)}(f) + d_{f, \mathcal{F}}^{(\rho)}(\hat{\mathcal{S}}, \hat{\mathcal{T}}) + \lambda^* + complexity \quad (1)$$

Here, the hypothesis space \mathcal{F} consists of classifiers $f : x \rightarrow \mathbb{R}^{|y|} = \mathbb{R}^k$, where each dimension of the output represents the predictive confidence for a class. The superscript (ρ) denotes a custom margin loss. For instance, $err_{\mathcal{S}}^{(\rho)}(f)$ employs the Softmax+Cross-Entropy loss as the margin loss.

The problem then reduces to minimizing $d_{f,\mathcal{F}}^{(\rho)}(\hat{\mathcal{S}}, \hat{\mathcal{T}})$, formulated as follows:

$$d_{f,\mathcal{F}}^{(\rho)}(\hat{\mathcal{S}}, \hat{\mathcal{T}}) \triangleq \sup_{f' \in \mathcal{F}} \left(\text{disp}_{\hat{\mathcal{T}}}^{(\rho)}(f', f) - \text{disp}_{\hat{\mathcal{S}}}^{(\rho)}(f', f) \right) \quad (2)$$

$$\text{disp}_{\hat{\mathcal{D}}}^{(\rho)}(f', f) \triangleq \mathbb{E}_{\hat{\mathcal{D}}} \text{MarginLoss}_{f'}(h_f) = \frac{1}{n} \sum_i \text{MarginLoss}_{f'}(x_i, h_f(x_i)) \quad (3)$$

Here, h_f denotes the class predicted with the highest probability by f , where f is the source domain classifier and f' is an auxiliary target domain classifier. By introducing a feature extractor φ , the minimization of MDD is formulated as a minimax game:

$$\min_{f, \varphi} \max_{f'} \left(\text{disp}_{\varphi(\hat{\mathcal{T}})}^{(\rho)}(f', f) - \text{disp}_{\varphi(\hat{\mathcal{S}})}^{(\rho)}(f', f) \right) \quad (4)$$

4 METHOD DERIVATION AND ANALYSIS

This chapter is structured as follows. We commence by introducing a distance margin within the feature space in Section 4.1. Building upon this, Section 4.2 integrates the proposed distance margin with the generalization bound from Section 3.2, thereby formulating the UES loss. This UES loss is a margin-based regularization function designed to be universally applicable to UDA methods involving classification. Section 4.3 furnishes a theoretical demonstration that the UES loss surpasses the commonly employed entropy minimization in UDA. The overall formulation of the proposed approach is consolidated in Section 4.4.

4.1 DISTANCE MARGIN

To enhance the discriminability of sample features, it is imperative to position them as far as possible from the decision boundary. As illustrated in Figure 3, when analyzing with two adjacent templates W_1 and W_2 (where a template W_k denotes the class vector corresponding to label k in the final linear classification layer W), the decision boundary v between the two classes must satisfy $W_1^T v = W_2^T v$ (ignoring the bias term), which is equivalent to $(W_1^T - W_2^T)v = 0$. The distance from a sample feature a_i to this decision boundary is given by:

$$\text{Distance} = \frac{\|(W_1^T - W_2^T) \cdot a_i\|_2}{\|(W_1^T - W_2^T)\|_2} \quad (5)$$

To maximize this distance, an intuitive and elegant approach is to scale the sample feature a_i by a factor of ρ , yielding:

$$\text{Distance}_{\rho} = \frac{\|(W_1^T - W_2^T)(\rho a_i)\|_2}{\|(W_1^T - W_2^T)\|_2} = \rho \frac{\|(W_1^T - W_2^T) a_i\|_2}{\|(W_1^T - W_2^T)\|_2} = \rho \text{Distance} \quad (6)$$

To achieve this effect, we employ the classifier’s margin loss, which integrates a fully-connected layer, softmax activation, and cross-entropy loss (Appendix A):

$$\text{softmax}(W a_i + b)_{y_i} = \frac{e^{W_{y_i}^T a_i + b_{y_i}}}{\sum_k e^{W_k^T a_i + b_k}} = \frac{e^{-\|a_i - W_{y_i}\|_2^2 + \|W_{y_i}\|_2^2 - W_{y_i}^T a_i + b_{y_i} + \|a_i\|_2^2}}{\sum_k e^{-\|a_i - W_k\|_2^2 + \|W_k\|_2^2 - W_k^T a_i + b_k + \|a_i\|_2^2}} \quad (7)$$

We observe that while minimizing the margin loss reduces the Euclidean distance between the feature and its corresponding template, the complex functional relationship and variable dependencies between the margin loss and the distance $\|a_i - W_{y_i}\|_2^2$ prevent it from effectively enhancing intra-class compactness. Consequently, we adopt the distance $\|a_i - W_{y_i}\|_2^2$ directly as a feature-level

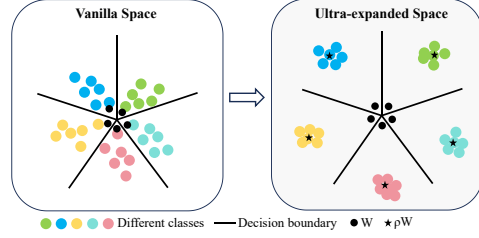


Figure 3: (Best viewed in color.) Schematic diagram of the Distance Margin mechanism.

supervisory signal. However, this alone does not strengthen inter-class separability. The aforementioned decision boundary analysis provides a solution. We thus propose a Distance Margin as follows:

$$Margin_{Distance} = \|a_i - \rho W_{y_i}\|_2^2 = \|\varphi(x_i) - \rho W_{y_i}\|_2^2 \quad (8)$$

where ρ is a Ultra-expanded hyperparameter, and the gradient computation for W_{y_i} is detached. Under this formulation, When ρ assumes large values while $Margin_{Distance}$ undergoes reduction, the transformed feature $\varphi(x_i)$ is compelled to approach ρW_{y_i} , thereby constructing a Ultra-expanded space that promotes both intra-class compactness and inter-class separability.

4.2 ULTRA-EXPANDED SPACE LOSS

Substituting the distance margin from Equation 8 into Equation 3 yields:

$$disp_{\mathcal{D}}^{(\rho)}(f', f) \triangleq \frac{1}{n} \sum_{i=1}^n \left\| \varphi(x_i) - \rho W_{h_f(x_i)}^{f'} \right\|_2^2 \quad (9)$$

Where $W^{f'}$ denotes the class weight matrix of the auxiliary target domain classifier f' . Since we detach the gradient computation for W , this constitutes a function independent of the optimization of classifiers (f', f) . Consequently, Equation 4 can be reformulated as follows:

$$\begin{aligned} & \min_{f, \varphi} \max_{f'} \left(disp_{\varphi(\hat{\mathcal{T}})}^{(\rho)}(f', f) - disp_{\varphi(\hat{\mathcal{S}})}^{(\rho)}(f', f) \right) \\ &= \min_{f, \varphi} \max_{f'} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \left\| \varphi(x_i^t) - \rho W_{h_f(x_i^t)}^{f'} \right\|_2^2 - \gamma \frac{1}{N_s} \sum_{i=1}^{N_s} \left\| \varphi(x_i^s) - \rho W_{h_f(x_i^s)}^{f'} \right\|_2^2 \right) \\ &= \min_{\varphi} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \left\| \varphi(x_i^t) - \rho W_{h_f(x_i^t)}^{f'} \right\|_2^2 - \gamma \frac{1}{N_s} \sum_{i=1}^{N_s} \left\| \varphi(x_i^s) - \rho W_{h_f(x_i^s)}^{f'} \right\|_2^2 \right) \end{aligned} \quad (10)$$

where γ is a coefficient that combines two terms. As $W^{f'}$ not undergo gradient-based optimization, we directly employ the parameters W^f from the source domain classifier. This implementation effectively reduces the framework to a single-classifier architecture. In practice, we observe that the latter term yields fluctuating performance gains. This is likely because the source domain features are already well-clustered by the supervised loss of the source classifier, rendering this additional regularization term redundant or even conflicting. Thus, we set $\gamma = 0$. The optimization objective is thus formulated as the minimization of the UES loss.

$$\min UES = \min_{\varphi} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \left\| \varphi(x_i^t) - \rho W_{h_f(x_i^t)}^f \right\|_2^2 \right) \quad (11)$$

Divested of its minimax game mechanism, the UES loss is functionally redefined as a model-agnostic regularizer, enabling direct incorporation into arbitrary UDA architectures. This adaptability makes it applicable to a wide range of UDA classification tasks, such as image classification, semantic segmentation and object detection.

Conventional UDA methods aim to bound the target domain error via a generalization bound and then minimize this bound. In contrast, our proposed margin-based regularization not only operates within this theoretical framework but also further pushes the target domain error away from the established generalization bound, thereby achieving additional performance gains.

4.3 DISTANCE MARGIN V.S. ENTROPY MINIMIZATION

We proceed with a formal analysis of the properties of the distance margin in Equation 8. When optimizing the standard FC + softmax + cross-entropy loss, the feature vector $a_i = \varphi(x_i)$ tends to distribute around a scaled version of the template vector $\rho_w W_{y_i}$, where the scaling factor ρ_w is not fixed (Figure 2 helps to understand qualitatively). By introducing and effectively optimizing the proposed loss from Equation 8, the feature a_i is constrained to a region ρW_{y_i} near, which is equivalent to scaling a_i by a factor of ρ . Therefore, Equation 7 becomes $softmax\left(W \frac{\rho}{\rho_w} a_i + b\right)$ to

produce a clearer output probability distribution. It is noteworthy that entropy minimization $H(p) = -\text{plogp}$, also promotes sharper probability distributions. The gradients of these two objectives with respect to the feature a_i are given by:

$$\frac{2(a_i - \rho W_{y_i})}{W(p(H(p) - \text{logp}))} \quad (12)$$

Obviously, the former has a more stable gradient that makes learning easier and prevents overfitting.

Entropy minimization is widely adopted in UDA, where it is generally understood to enhance the confidence of predictions on the target domain or to encourage more similar prediction distributions across domains, thereby promoting domain adaptation. Based on the theoretical analysis above, the proposed UES regularization—rooted in the distance margin—represents a theoretically superior alternative to conventional entropy minimization.

4.4 OVERALL FORMULATION

The overall objective function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{UDA} + \lambda \mathcal{L}_{UES} \quad (13)$$

The proposed margin regularization method can be directly incorporated into existing UDA frameworks. The complete implementation Pytorch code for \mathcal{L}_{UES} is provided in Figure 4(right). This work introduces only two hyperparameters: a standard trade-off parameter λ and the Ultra-expanded hyperparameter ρ inherent to the \mathcal{L}_{UES} formulation.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Dataset. The evaluation employs four different difficulty benchmarks for image classification tasks: (1) Digit [USPS (1.8k)/MNIST (2k images), 10 classes(M/U), 2 tasks]; (2) Office-31 [4.1k images, 31 classes (A/W/D), 6 tasks]; (3) Office-Home [15.5k images, 65 classes (Ar/Ci/Pr/Rw), 12 tasks]; and(4) DomainNet[600K images, 345classes (C/I/P/Q/R/S), 30 tasks].

Semantic segmentation task is evaluated via adaptation from the GTA5 dataset (24,966 images) to the CityScapes dataset (2,975 training and 500 validation images) across the 19 shared categories.

UDA Algorithm. We evaluate five UDA algorithms for UDA: DAN(Long et al., 2015), DeepCORAL(Sun & Saenko, 2016), DANN(Ganin & Lempitsky, 2015), DAAN(Yu et al., 2019) and DSAN(Zhu et al., 2020).

Margin loss. In addition to the proposed UES loss, the present study employs Center loss and ArcFace loss as baseline comparative methods, with all hyperparameters strictly following the configurations established in their respective original publications. This approach ensures a fair and reproducible experimental setup while maintaining methodological consistency with existing literature.

Training Details. For fair comparison with existing methods, we employed the standard ResNet-50 pretrained on ImageNet as the backbone network. Experiments were conducted with three different random seeds (0, 1, 2) and results were averaged. For Office-31 and Office-Home datasets, input images were resized to 224×224 , while Digit dataset images were resized to 16×16 . Notably, for the Digit dataset, we modified the ResNet-50 architecture by replacing its first convolutional and pooling layers with a 1×1 convolutional layer. Following Wang et al. (2025)’s optimization strategy, we used SGD with momentum 0.9, weight decay 5×10^{-4} , and an initial learning rate of 0.1 which was decayed according to $(1 + 0.0003 * epoch)^{-0.75}$.

5.2 RESULTS

UD Single-Source Image Classification. Experimental results of Digit,Office31 and OfficeHome are presented in appendix B and a more concise performance change chart is drawn in Figure 4 (a) (The x-axis data represents the baseline performance using only UDA algorithms, while the y-axis

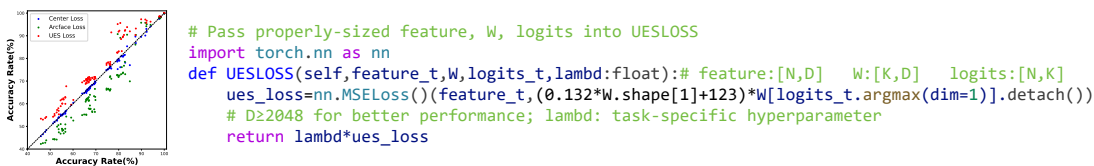


Figure 4: (left)Performance changes under different margin loss. (right) Plug and play Pytorch UES code.

UDA	Margin	R→C	R→P	P→C	P→R	C→S	S→P	Avg
DANN	Softmax	49.13	47.05	45.41	59.75	41.07	42.93	47.56
	UES	54.09	49.24	46.39	59.41	42.50	44.54	49.36
DSAN	Softmax	52.69	49.61	45.74	58.90	42.74	44.53	49.04
	UES	55.91	51.19	46.97	59.50	43.82	45.27	50.43

Table 1: Accuracy (%) on DomainNet for unsupervised domain adaptation

represents the performance after using margin regularization loss. The diagonal line serves as a reference, where points above the line indicate performance improvement and points below denote degradation. Data points are color-coded by their respective margin loss for clear visualization). The results demonstrate that: (1) Arcface loss only achieves accuracy improvement on the simpler Digit dataset; (2) Center loss exhibits accuracy fluctuations around the baseline performance across all three datasets; and (3) our proposed UES loss consistently improves accuracy across all datasets and feature alignment methods, significantly outperforming both Arcface and Center loss. A representative example is the A→W task in Table 5, where our UES loss achieves over 10% accuracy improvement while both ArcFace and Center loss show decreased performance.

The partial experimental results of the DomainNet dataset are shown in Table 1, which is a dataset with significant differences between domains. We found an example of a slight decrease in accuracy when using UES. UES leverages pseudo-labels generated to push target-domain features away from source decision boundaries. However, its effectiveness can be compromised when pseudo-labels are of low quality. The pseudo-label weighting technique, commonly employed in self-training, can be integrated into the UES framework and is expected to yield further performance gains.

UDA Multi-Source Image Classification. We evaluated the OfficeHome dataset using the multi-

	Cl,Pr,Rw→Ar	Ar,Pr,Rw→Cl	Ar,Ci,Rw→Pr	Ar,Ci,Pr→Rw	Avg
MFSAN	71.24	60.8	80.58	81.66	73.57
MFSAN+UES	72.68	62.1	81.48	82.71	74.74

Table 2: Accuracy (%) on OfficeHome for multi-source unsupervised domain adaptation

source unsupervised domain adaptation algorithm MFSAN (Zhu et al., 2019), directly applying the hyperparameters from Figure 4(right) without any modifications. The results in Table 2 demonstrate that UES loss consistently enhances performance across every experimental configuration.

UDA Semantic Segmentation. We evaluate the GTA5→CityScapes adaptation using the Advent algorithm with DeepLabV3 as the baseline. In DeepLabV3, the output is generated through a final convolutional layer followed by interpolation-based upsampling. This convolutional layer is treated as a linear layer for applying the UES loss, and we adjust its hidden dimension to 2048. All trade-off hyperparameters of Advent were set to 0.001, and the same trade-off parameter was applied to the UES without any modifications. The results are illustrated in Figure 5.

5.3 ANALYSIS

Convergence and Runtime. As illustrated in Figure 6(left), the well-behaved landscape of the loss function, due to its convexity and smoothness, results in a smoother trajectory of accuracy convergence, making the learning process more robust.

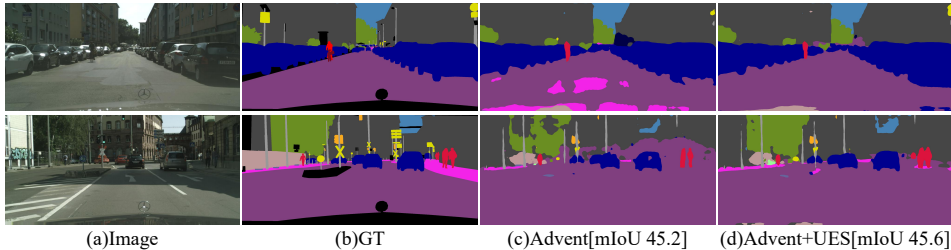


Figure 5: Results in the GTA5→Cityscapes .

A counterintuitive observation is that although the UES loss introduces an additional tensor operation—which would theoretically lead to a slight increase in computational time—the empirical results indicate an actual reduction in runtime. This improvement can be attributed to the fact that the UES loss may enable the framework’s Pytorch computational graph optimizer to identify and exploit more opportunities for operator fusion, thereby enhancing overall execution efficiency.

Hyperparameter. As shown in Figure 6, we conduct sensitivity analysis on the two newly introduced hyperparameters, λ and ρ . λ serves as a conventional balancing coefficient and demonstrates stable performance when values remain below 1. We recommend adjusting this parameter according to different task types—for instance, setting $\lambda = 0.1$ has proven effective for image classification tasks. More importantly, the Ultra-expanded parameter ρ is tested across an extensive range from 10 to 3000, where it consistently yields stable and satisfactory results.

To further investigate its scalability across different feature dimensions, we examine the relationship between the optimal ρ and the dimensionality of W (Figure 6, right). While previous experiments were conducted using 2048-dimensional features from ResNet, we observe that as the dimensionality increases, the optimal ρ value exhibits an approximately linear growth trend. By taking the average of the top three ρ values corresponding to the highest accuracy at each dimensionality and applying least squares fitting, we derive the relationship $\rho = 0.132\text{dim}(W) + 123$. This establishes ρ as an automatically configurable parameter, effectively eliminating the need for manual tuning in practical applications. Based on observations, we find that smaller dimensions of the W tend to lead to significant performance fluctuations across different values of the hyperparameter ρ . Therefore, we recommend that the dimensionality of the linear projection W should not be set too small in practice.

UES v.s. Entropy Minimization. We evaluate the performance of the DANN algorithm combined with UES and Entropy Minimization respectively across three distinct tasks in Table 3. The results demonstrate that UES consistently achieves superior performance.

	Digit M→U	Office31 A→W	OfficeHome Rw→Pr
Entropy Minimization	94.11	89.55	81.48
UES	96.64	92.24	82.61

Table 3: Comparison of accuracy between UES and Entropy Minimization in three tasks

6 EFFECTIVENESS ANALYSIS OF THE UES

To elucidate the underlying mechanisms for the effectiveness of UES, we proceed with a systematic analysis from two perspectives: the optimization behavior of the loss, and the enhancement of intra-class compactness alongside inter-class separability.

Optimization behavior of the loss. As the domain confuser becomes increasingly optimized, the influence of its loss signal on the encoder gradually weakens (Zhang et al., 2023), thereby impeding effective domain adaptation. As shown in Figure 6(middle), the proposed UES loss not only introduces auxiliary supervision to facilitate model optimization but also further reduces the domain confuser loss. A comparative analysis with two alternative margin-based approaches—Center loss and ArcFace loss—reveals distinct behavioral differences: Center loss exhibits a negligible reg-

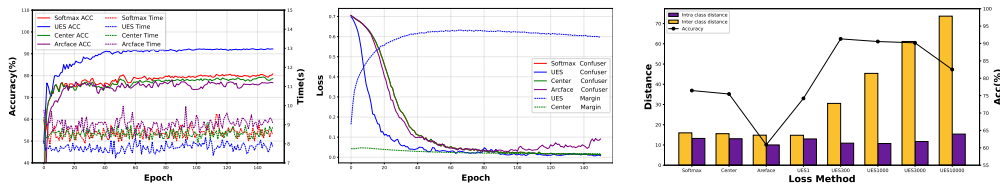


Figure 6: Schematic illustration of Office31 A→W task performance evolution across training epochs (left) Accuracy and runtime. (middle) Domain confuser and margin regularization term loss. ArcFace constitutes a direct variant of the Softmax, wherein neither formulation incorporates an explicit margin regularization term. (right) Intra-class and Inter-class distance in epoch 40. The notation UES ρ denotes the method utilizing different Ultra-expanded hyperparameters ρ .

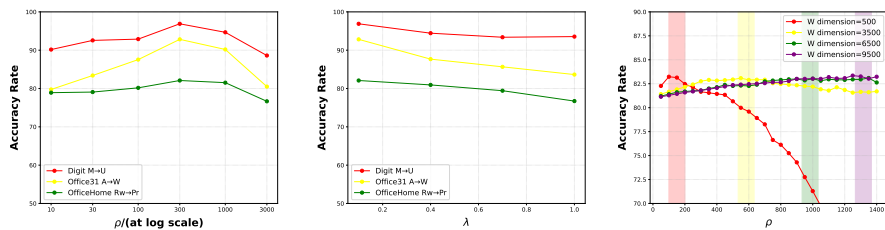


Figure 7: Analysis of hyperparameter sensitivity. (left) Fixed $\lambda = 0.1$. (middle) Fixed $\rho = 300$. (right) Performance Variation with Respect to ρ and $\dim(W)$.

ularization effect, manifested by virtually identical domain confuser loss curves compared to the baseline Softmax (evidenced by the overlapping green and red solid lines); ArcFace, as a Softmax variant, induces oscillatory and persistently increasing confuser loss. (Liang et al., 2021)

Intra-class compactness and Inter-class separability Intra-class compactness reflects the degree of aggregation of features within each class, and Inter-class separability quantifies the differences between different classes. We conduct qualitative and quantitative analyses on intra-class compactness and inter-class separability.

Qualitative Analysis: As shown in Figure 8, T-SNE is used to visualize the feature distribution. The proposed UES successfully forms 31 well-separated clusters with clear decision boundaries, leaving almost no scattered data points in the gaps between clusters.

Quantitative Analysis: Currently, there is no established method to quantify Intra-class compactness and Inter-class separability. We propose a novel approach based on their inherent properties: Inter-class separability is measured by the average distance between class centers (termed inter-class distance), and intra-class compactness is evaluated by the average distance from features to their corresponding class centers based on ground-truth labels (termed intra-class distance). As illustrated in Figure 6 (right), the intra-class distance exhibits no significant correlation with the final accuracy, whereas the inter-class distance within a certain range can substantially enhance the accuracy. Notably, a larger Intra-class distance is not always beneficial; an excessively large intra-class distance leads to performance degradation, as it disrupts the semantic information among features. Both Center loss and ArcFace loss have minimal impact on inter-class and Intra-class distances,

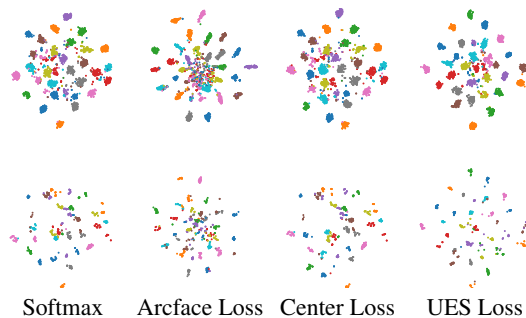


Figure 8: In the transfer task Office 31 A→W, T-SNE visualization of learning features is performed, with the first row representing the source domain and the second row representing the target domain, with each column representing a different loss.

486 while our proposed method significantly increases the Inter-class distance with flexible controllabil-
487 ity.

488 The remarkable achievement in Inter-class separability highlights the extraordinary capability of
489 UES in constructing an Ultra-expanded Space for feature.
490

491 7 CONCLUSION

492 This paper presents the first distance margin-based regularization loss specifically designed to ad-
493 dress the issue of target domain data violating decision boundaries in Unsupervised Domain Adap-
494 tation (UDA). In contrast to prevailing margin-based methods, our approach achieves performance
495 improvements by substantially enhancing Inter-class separability. The proposed regularization tech-
496 nique is universally applicable to any classification-based UDA framework and requires minimal
497 hyperparameter tuning.
498

500 ETHICS STATEMENT

501 The text of this manuscript was polished for English language using Deepseek. The authors are
502 solely responsible for the final content.
503

505 REFERENCES

- 506 Eman Ali, Sathira Silva, and Muhammad Haris Khan. Dpa: Dual prototypes alignment for unsuper-
507 vised adaptation of vision-language models. In *2025 IEEE/CVF Winter Conference on Applica-*
508 *tions of Computer Vision (WACV)*, pp. 6083–6093. IEEE, 2025.
- 509 Chiori Azuma, Tomoyoshi Ito, and Tomoyoshi Shimobaba. Adversarial domain adaptation using
510 contrastive learning. *Engineering Applications of Artificial Intelligence*, 123:106394, 2023.
- 511 Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations
512 for domain adaptation. *Advances in neural information processing systems*, 19, 2006.
- 513 Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wort-
514 man Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175,
515 2010.
- 516 Zhuangzhuang Chen, Bin Pu, Lei Zhao, Jie He, and Pengchen Liang. Divide and augment: Su-
517 pervised domain adaptation via sample-wise feature fusion. *Information Fusion*, 115:102757,
518 2025.
- 519 Jun Dan, Tao Jin, Hao Chi, Shunjie Dong, Haoran Xie, Keying Cao, and Xinjing Yang. Trust-aware
520 conditional adversarial domain adaptation with feature norm alignment. *Neural Networks*, 168:
521 518–530, 2023.
- 522 Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin
523 loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision*
524 *and pattern recognition*, pp. 4690–4699, 2019.
- 525 Wenyong Dong, Zhixue Liang, Liping Wang, Gang Tian, and Qianhui Long. Unsupervised domain
526 adaptation segmentation algorithm with cross-domain data augmentation and category contrast.
527 *Neurocomputing*, 623:129393, 2025.
- 528 Yuchun Fang, Chen Chen, Wei Zhang, Jiahua Wu, Zhaoxiang Zhang, and Shaorong Xie. Prototype
529 learning for adversarial domain adaptation. *Pattern Recognition*, 155:110653, 2024.
- 530 Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In
531 *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.
- 532 Xu Gezheng, Hui Guo, Li Yi, Charles Ling, Boyu Wang, and Grace Yi. Revisiting source-free
533 domain adaptation: a new perspective via uncertainty control. In *The Thirteenth International*
534 *Conference on Learning Representations*, 2024.

- 540 Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros,
541 and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International*
542 *conference on machine learning*, pp. 1989–1998. Pmlr, 2018.
- 543
- 544 Janguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, Jie Jiang, and Mingsheng
545 Long. Forkmerge: Mitigating negative transfer in auxiliary-task learning. *Advances in neural*
546 *information processing systems*, 36:30367–30389, 2023.
- 547 Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recog-
548 nition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*,
549 pp. 18750–18759, 2022.
- 550
- 551 Junjie Li, Yixin Zhang, Zilei Wang, Saihui Hou, Keyu Tu, and Man Zhang. Probabilistic contrastive
552 learning for domain adaptation. In *Proceedings of the Thirty-Third International Joint Conference*
553 *on Artificial Intelligence*, pp. 1001–1009, 2024.
- 554 Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport
555 distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on*
556 *computer vision and pattern recognition*, pp. 13936–13944, 2020.
- 557
- 558 Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-
559 level self-labeling for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF*
560 *conference on computer vision and pattern recognition*, pp. 11593–11603, 2022.
- 561 Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable
562 semantic augmentation for domain adaptation. In *Proceedings of the IEEE/CVF conference on*
563 *computer vision and pattern recognition*, pp. 11516–11525, 2021a.
- 564
- 565 Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic
566 concentration for domain adaptation. In *Proceedings of the IEEE/CVF international conference*
567 *on computer vision*, pp. 9102–9111, 2021b.
- 568 Jian Liang, Kaixiong Gong, Shuang Li, Chi Harold Liu, Han Li, Di Liu, Guoren Wang, et al. Pareto
569 domain adaptation. *Advances in Neural Information Processing Systems*, 34:12917–12929, 2021.
- 570
- 571 Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Ad-*
572 *vances in Neural Information Processing Systems*, 34:22968–22981, 2021.
- 573 Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep
574 hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer*
575 *vision and pattern recognition*, pp. 212–220, 2017.
- 576
- 577 Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye
578 Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives.
579 *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.
- 580 Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with
581 deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR,
582 2015.
- 583
- 584 Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsu-
585 pervised domain adaptation. In *European conference on computer vision*, pp. 415–430. Springer,
586 2020.
- 587 Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Bridging domain spaces for
588 unsupervised domain adaptation. *Pattern Recognition*, 164:111537, 2025.
- 589
- 590 Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge*
591 *and data engineering*, 22(10):1345–1359, 2009.
- 592 Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-
593 entropy loss for adversarial robustness. In *International Conference on Learning Representations*,
2019.

- 594 Wenjie Peng, Quhui Ke, Jinglin Liang, Shuangping Huang, and Tianshui Chen. Proxy-an loss for
595 deep metric learning. *Neural Networks*, pp. 108254, 2025.
- 596
- 597 Maohao Shen, Yuheng Bu, and Gregory W Wornell. On balancing bias and variance in unsuper-
598 vised multi-source-free domain adaptation. In *International conference on machine learning*, pp.
599 30976–30991. PMLR, 2023.
- 600 Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In
601 *European conference on computer vision*, pp. 443–450. Springer, 2016.
- 602
- 603 Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adver-
604 sarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of
605 the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2517–2526, 2019.
- 606 Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for
607 loss functions in image classification. In *Proceedings of the IEEE conference on computer vision
608 and pattern recognition*, pp. 9117–9126, 2018.
- 609
- 610 Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verifica-
611 tion. *IEEE Signal Processing Letters*, 25(7):926–930, 2018a.
- 612 Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei
613 Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE
614 conference on computer vision and pattern recognition*, pp. 5265–5274, 2018b.
- 615 Jindong Wang et al. Everything about transfer learning and domain adaption. [http://
616 transferlearning.xyz](http://transferlearning.xyz), 2025.
- 617
- 618 Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:
619 135–153, 2018.
- 620 Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: Task-
621 oriented alignment for unsupervised domain adaptation. *Advances in Neural Information Pro-
622 cessing Systems*, 34:13834–13846, 2021.
- 623
- 624 Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach
625 for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer,
626 2016.
- 627
- 628 Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Instance-invariant domain adaptive object
629 detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine
630 Intelligence*, 44(8):4178–4193, 2021.
- 631 Jiamu Xu, Xiaoxiang Liu, Xinyuan Zhang, Yain-Whar Si, Xiaofan Li, Zheng Shi, Ke Wang, and
632 Xueyuan Gong. X2-softmax: Margin adaptive loss function for face recognition. *Expert Systems
633 with Applications*, 249:123791, 2024.
- 634 Lu Yang, Yunlong Wang, Lingqiao Liu, Peng Wang, and Yanning Zhang. Center prediction loss for
635 re-identification. *Pattern Recognition*, 132:108949, 2022.
- 636
- 637 Chaohui Yu, Jindong Wang, Yiqiang Chen, and Meiyu Huang. Transfer learning with dynamic
638 adversarial adaptation network. In *2019 IEEE international conference on data mining (ICDM)*,
639 pp. 778–786. IEEE, 2019.
- 640 Jin Yuan, Feng Hou, Ying Yang, Yang Zhang, Zhongchao Shi, Xin Geng, Jianping Fan, Zhiqiang
641 He, and Yong Rui. Domain-aware graph network for bridging multi-source domain adaptation.
642 *IEEE Transactions on Multimedia*, 26:7210–7224, 2024.
- 643
- 644 Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA
645 Journal of Automatica Sinica*, 10(2):305–329, 2022.
- 646
- 647 Xiao Zhang, Rui Zhao, Yu Qiao, and Hongsheng Li. Rbf-softmax: Learning deep representative
prototypes with radial basis function softmax. In *European Conference on Computer Vision*, pp.
296–311. Springer, 2020.

- 648 Yifan Zhang, Xue Wang, Jian Liang, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Free
649 lunch for domain adversarial training: Environment label smoothing. In *ICLR*, 2023.
650
- 651 Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm
652 for domain adaptation. In *International conference on machine learning*, pp. 7404–7413. PMLR,
653 2019.
- 654 Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant
655 representations for domain adaptation. In *International conference on machine learning*, pp.
656 7523–7532. PMLR, 2019.
- 657 Lihua Zhou, Mao Ye, Xiatian Zhu, Siying Xiao, Xu-Qian Fan, and Ferrante Neri. Homeomorphism
658 alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International
659 Conference on Computer Vision*, pp. 18699–18710, 2023.
660
- 661 Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and clas-
662 sifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI confer-
663 ence on artificial intelligence*, 2019.
- 664 Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong,
665 and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions
666 on neural networks and learning systems*, 32(4):1713–1722, 2020.
667
668
669

670 A CLASS WEIGHT MATRIX

671

672 The inner product operation between sample features and the class weight matrix (within the fully-
673 connected layer), followed by projection into the probability space via the softmax function, enables
674 the computation of cross-entropy loss with respect to the ground-truth labels. The class weight
675 matrix, learned through the training of the entire neural network on the dataset, can be regarded
676 as a high-order statistical representation of the data distribution across different classes. However,
677 this information remains underutilized in the target domain. A natural intuition therefore suggests
678 clustering features in the vicinity of their predicted class weight vectors.

680 B EXPERIMENTAL RESULTS

681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724

UDA	Margin	M→U	U→M	Avg
DAN	Softmax	93.98	83.70	88.84
	Arcface	94.83	90.46	92.65
	Center	94.12	85.35	89.73
	UES	96.22	93.81	95.01
DANN	Softmax	92.29	87.53	89.91
	Arcface	95.07	90.38	92.72
	Center	91.74	86.78	89.26
	UES	96.64	93.18	94.91
DeepCORAL	Softmax	91.68	80.71	86.20
	Arcface	95.72	89.08	92.40
	Center	90.09	76.11	83.10
	UES	97.16	95.63	96.40
DAAN	Softmax	89.87	79.56	84.71
	Arcface	95.75	90.73	93.24
	Center	89.53	75.28	82.41
	UES	97.27	95.15	96.21
DSAN	Softmax	92.12	87.35	89.73
	Arcface	94.20	86.08	90.14
	Center	93.01	89.63	91.32
	UES	97.68	95.31	96.49

Table 4: Accuracy (%) on Digit for unsupervised domain adaptation

725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751

UDA	Margin	A→W	W→A	A→D	D→A	D→W	W→D	Avg
DAN	Softmax	82.72	65.74	82.32	67.41	97.82	100.00	82.67
	Arcface	78.49	61.68	76.77	63.94	97.90	99.93	79.78
	Center	81.55	65.47	81.32	66.73	97.65	99.93	82.11
	UES	92.20	70.87	90.36	71.37	98.44	100.00	87.21
DANN	Softmax	80.33	65.80	81.92	66.53	98.11	100.00	82.11
	Arcface	72.45	61.59	73.76	62.45	97.69	100.00	77.99
	Center	78.99	65.47	80.78	66.14	98.02	100.00	81.57
	UES	92.24	69.47	89.69	71.54	98.28	100.00	86.87
DeepCORAL	Softmax	79.95	66.05	81.79	66.89	97.98	100.00	82.11
	Arcface	71.78	61.88	73.42	62.90	97.90	99.86	77.96
	Center	79.11	65.74	80.92	66.47	97.73	99.82	81.63
	UES	91.82	69.12	89.35	70.95	98.61	100.00	86.64
DAAN	Softmax	76.77	65.53	81.05	66.25	97.82	100.00	81.23
	Arcface	71.02	60.91	73.02	62.32	97.94	100.00	77.53
	Center	75.51	65.11	80.92	65.68	97.48	100.00	80.78
	UES	91.48	68.76	88.55	70.37	98.53	100.00	86.28
DSAN	Softmax	84.90	66.86	82.86	69.21	98.53	99.93	83.71
	Arcface	69.89	61.25	73.02	63.34	98.11	99.86	77.58
	Center	77.02	66.11	78.11	68.59	98.36	99.86	81.34
	UES	91.94	69.38	92.50	70.80	98.69	100.00	87.22

Table 5: Accuracy (%) on office-31 for unsupervised domain adaptation

752
753
754
755

756									
757									
758									
759									
760									
761									
762									
763	UDA	Margin	Ar→Cl	Cl→Ar	Ar→Pr	Pr→Ar	Pr→Cl	Cl→Pr	
764		Softmax	52.80	56.58	69.29	56.62	48.24	65.86	
765		Arcface	44.18	48.96	57.53	47.61	43.85	55.08	
766		Center	52.76	56.11	68.76	56.61	48.36	64.78	
767		UES	56.15	62.94	71.81	62.69	54.18	70.47	
768		Softmax	51.88	55.25	68.44	55.29	47.80	63.99	
769		Arcface	44.30	48.83	56.32	48.05	42.80	54.96	
770		Center	52.05	55.21	68.19	54.89	47.58	64.51	
771		UES	55.47	60.07	71.29	60.05	53.67	69.42	
772		Softmax	51.89	55.72	69.58	55.89	46.78	65.72	
773		Arcface	44.29	48.35	56.39	46.91	42.60	55.25	
774		Center	51.96	55.54	68.99	55.50	46.52	65.39	
775		UES	54.76	62.38	71.81	61.87	52.99	70.48	
776		Softmax	51.24	55.12	68.39	54.52	45.90	64.87	
777		Arcface	44.36	48.96	56.65	46.79	42.27	55.25	
778		Center	51.34	54.99	68.32	55.23	45.68	64.66	
779		UES	54.25	61.92	71.53	61.50	53.40	70.04	
780		Softmax	53.69	57.02	69.72	60.11	53.76	67.40	
781		Arcface	48.75	54.34	63.03	52.57	47.39	59.22	
782		Center	53.90	56.90	69.43	60.44	54.04	67.31	
783		UES	56.88	67.71	71.27	61.68	55.32	70.55	
784	Alignment	Margin	Ar→Rw	Rw→Ar	Cl→Rw	Rw→Cl	Pr→Rw	Rw→Pr	Avg
785		Softmax	76.03	67.05	67.28	54.58	75.39	80.05	64.15
786		Arcface	66.14	61.51	57.63	51.01	66.34	72.44	56.02
787		Center	75.74	66.99	67.17	54.60	75.21	79.71	63.90
788		UES	77.43	71.81	70.67	58.92	78.25	83.14	68.20
789		Softmax	76.14	66.39	66.89	56.62	74.67	79.40	63.56
790		Arcface	65.78	61.90	57.88	50.88	66.17	72.56	55.87
791		Center	75.90	66.08	66.61	55.61	74.94	78.60	63.35
792		UES	77.02	70.04	69.78	59.96	77.14	82.61	67.21
793		Softmax	76.43	66.29	67.70	53.56	75.39	79.55	63.71
794		Arcface	66.41	62.07	57.74	50.54	66.61	72.44	55.80
795		Center	76.16	66.39	67.72	53.70	75.29	79.47	63.55
796		UES	77.28	70.95	70.63	57.99	78.11	82.59	67.65
797		Softmax	76.10	65.95	67.50	53.01	74.69	79.34	63.05
798		Arcface	66.03	62.10	57.75	49.88	66.71	72.73	55.79
799		Center	75.80	65.99	67.07	52.54	74.76	78.83	62.93
800		UES	77.33	70.93	70.50	57.25	77.84	82.56	67.42
801		Softmax	76.15	69.59	69.81	59.17	76.31	81.36	66.17
802		Arcface	71.74	66.54	61.64	53.88	72.13	76.13	60.61
803		Center	76.24	69.88	69.54	59.38	76.56	81.05	66.22
804		UES	77.69	71.81	71.19	59.56	78.44	83.21	68.78

Table 6: Accuracy (%) on office-Home for unsupervised domain adaptation

805
806
807
808
809