Adaptive Entropy Guided Universal Domain Adaptation

Jawad Saeed
Department of Computer Science
Lahore University of Management Sciences
25100094@lums.edu.pk

Daanish Uddin Khan
Department of Computer Science
Lahore University of Management Sciences
25100004@lums.edu.pk

Muhammad Saad Haroon
Department of Computer Science
Lahore University of Management Sciences
25100147@lums.edu.pk

Abstract

Domain adaptation faces inherent challenges in reducing category gaps between labeled source domains and unlabeled target domains. This paper introduces Adaptive Entropy Guided Universal Domain Adaptation (AEG-UDA), a framework that builds upon the DANCE methodology to improve classification across known and unknown target categories. The proposed approach employs an adaptive thresholding mechanism to dynamically separate target samples into low-entropy (confident) and high-entropy (uncertain) groups, ensuring alignment with the evolving learning conditions. For low-entropy samples, we implement entropy-guided pseudo-labeling, which assigns soft labels based on the model's confidence, while high-entropy samples are handled using a novel Dynamic Rejection Loss (DRL) to reduce their impact on training. Evaluations on the Office-31 dataset across Closed Set, Open Set, Partial Set, and Open Partial Set scenarios demonstrate AEG-UDA's competitiveness, achieving mean per-class accuracy comparable to DANCE while reducing inference time in Closed and Partial domain adaptation scenarios. Visualizations using t-SNE further reveal improved cluster compactness and separation, highlighting AEG-UDA's ability to effectively align source and target domain features. Though challenges such as full-precision training variability remain, AEG-UDA establishes a robust foundation for addressing domain adaptation tasks. This work advances entropy-based methods and sets a pathway for adaptive mechanisms in universal domain adaptation.

GitHub Repository

1 Introduction

The rapid advancement of machine learning has led to significant interest in domain adaptation, with the goal of enabling models trained on labeled source domains to generalize to unlabeled target domains effectively. However, the semantic and co-variate shifts that may occur between the domains make this a challenging task due to learned features from the source domain not being sufficient enough for accurate target domain classification. Existing approaches in this field rely on the alignment of source and target domains using different approaches such as using adversial models You et al. [2019] and entropy based approaches Saito et al. [2020] for source and target feature alignment.

CS-6304 Advanced Topics in Machine Learning
Template from NeurIPS 2024

In a Domain Adaptation scenario let us assume that the label sets for our source and target domains are represents as follows. L_s represents the label set for the source domain while L_t represents the set for the target domain. We investigate four scenarios specifically which include the closed set ($L_s = L_t$) , open set ($L_s \subset L_t$) Busto and Gall [2017], partial set ($L_t \subset L_s$) Cao et al. [2018] and open-partial set You et al. [2019] which is a combination of open and partial set scenarios. Existing methods in the UDA space are designed to handle multiple scenarios but on the basis of hyperparameters that are derived through experimentation for specific adaptation problems. Due to the unlabeled nature of the target domain we cannot predict which of the possible scenarios can occur and therefore using specific hyperparameter values can lead to suboptimal results in varying scenarios.

We propose to overcome this issue by introducing AEG-UDA (Adaptive Entropy Guided Universal Domain Adaptation) framework which aims to address this problem through different approaches. Taking inspiration from the DANCE framework we use their "neighborhood clustering" technique that self-supervises feature learning in the target while preserving useful source features and class boundaries. Additionally, to refine the training process, we use an Entropy Guide Refined Pseudo-labeling approach, which assigns soft labels to low entropy target samples for fine-tuning the model iteratively. Lastly, we introduce a novel Dynamic Rejection Loss, which penalizes high entropy samples, forcing the model to learn to reject them in order to make better predictions in the context of adaptation.

2 Related Work

Our approach builds upon and is influenced by the DANCE architecture, Saito et al. [2020]. Since our approach focuses on integrating an adaptive threshold calculated through entropy, building on DANCE's entropy-based methodology feels like a logical progression to enhance the model's robustness. DANCE employs a static threshold to differentiate between low and high-entropy samples. However, we identified this as a potential area for exploration. The underlying premise is that as training progresses, the model's ability to distinguish between low and high-entropy samples improves. Yet, this enhanced capability is not captured in the initial entropy-based sample separation.

We hypothesize that introducing an adaptive threshold, evolving alongside the model's learning process, could address this limitation. As the model improves and generates more low-entropy samples, an adaptive threshold dynamically calculated based on the current entropy distribution could better guide the sample separation. This approach would align with the model's evolving capability, refining the learning process.

DANCE utilizes Neighborhood Clustering to align target samples. In contrast, we introduce an approach termed Entropy-Guided Pseudo-labeling (EGPL). Following the initial split into low and high-entropy samples, EGPL generates soft pseudo-labels for the low-entropy samples. Each sample is then assigned a weight based on its entropy value, with greater weight given to low-entropy samples that are further from the adaptive threshold, as these represent the most confident predictions. By integrating EGPL with KL divergence loss, the model can progressively align its predictions with the pseudo-label distribution, leveraging the certainty of low-entropy samples to guide the learning process effectively.

DANCE's entropy separation loss focuses on distinguishing known from unknown classes and then emphasizes either aligning or rejecting a sample based on a confidence margin. However, it employs a static threshold to determine whether a sample falls within this margin. Building on our hypothesis of an adaptive threshold, we propose a novel Dynamic Rejection Loss (DRL). DRL penalizes samples that exceed the adaptive threshold and fall outside a predefined static confidence margin. The degree of penalization is proportional to the sample's distance from the adaptive threshold. The objective of DRL is to minimize the uncertainty associated with high-entropy samples, thereby reducing classification ambiguity and improving the guidance of model predictions.

While DANCE processes low and high-entropy samples collectively, our approach treats them explicitly and independently. DANCE emphasizes categorization, aligning or rejecting samples, whereas we focus on refining the distinct influence of low and high-entropy samples throughout training. This allows us to address their unique characteristics more effectively, tailoring the learning process to maximize the utility of each sample type.

Although our architecture builds mainly on DANCE, the methodologies of DANN Ganin et al. [2016] and UAN You et al. [2019] provide valuable context for our work. DANN's adversarial approach to learning domain-invariant features informs our emphasis on robust entropy-based sample separation, while UAN's dynamic handling of uncertainty through a transferability criterion aligns conceptually with our adaptive threshold mechanism. These connections highlight the broader relevance of dynamically addressing uncertainty, which is integral to our Entropy-Guided Pseudo-labeling and Dynamic Rejection Loss strategies.

3 Methodology

3.1 Initial Setup: Universal Domain Adaptation

Considering that this is not a source free scenario we have access to both the source and and the target domain datasets. Given a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where $x_i^s \in \mathcal{X}_s$ and $y_i^s \in \mathcal{Y}_s$ we have a corresponding label set L_s. Additionally we have unlabeled target domain data $D_t = \{x_i^t\}_{i=1}^{N_t}$, where $x_i^t \in \mathcal{X}_t$ with a label set L_t where depending on the experiment the domains can share certain labels or have distinct labels. The goal here is assign the target samples either one of the labels from L_s or assign them a label corresponding to the "unknown" class.

Building on the DANCE framework, we make use of its **Neighborhood Clustering (NC)** approach, which essentially uses a prototype-based classifier that maps target samples close to their true class centroid (prototype) and far from samples of other classes. This is paired with an **Entropy Separation (ES)** loss that forces the classifier to align the sample with a source prototype or reject it as unknown. Additionally, we introduced a Pseudolabeling Mechanism that leverages low-entropy target samples, dynamically assigning soft labels based on the model's confidence. To ensure robustness, we incorporated an Adaptive Thresholding approach that dynamically adjusts the entropy threshold based on the distribution of target samples, effectively separating confident predictions from ambiguous ones. Furthermore, a **Dynamic Rejection Loss (DRL)** was added to penalize high-entropy samples, reducing their negative impact on training and improving alignment between source and target domains.

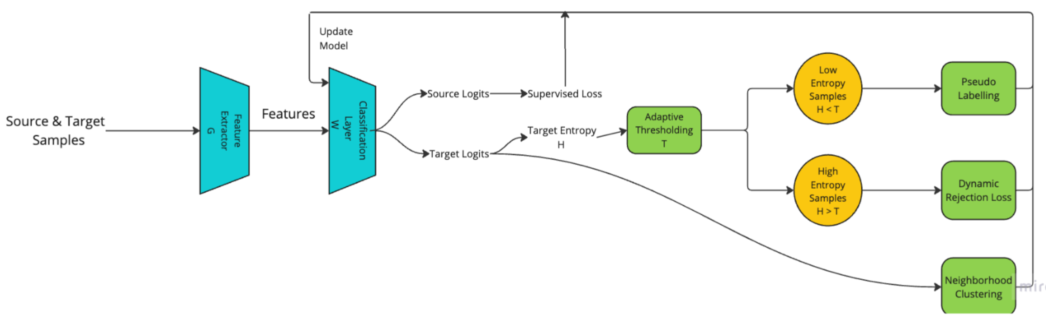


Figure 1: High Level Overview of the Model Architecture

3.2 Adpative Thresholding

Adaptive Thresholding plays a crucial part in ensuring our model's robustness by separating confident predictions from ambiguous ones. For each experiment a specific starting value is set for the threshold based on the values used by the authors of DANCE. The threshold is computed as follows using the entropy values of the current target batch. α and β represent weights that influence the effect of the upper and lower quartiles on the threshold calculation. Both values are initialized to 0.5 to give equal weight to both quartiles with the value adjusted every 100 epochs.

The idea behind this approach was based on the assumption that as the model is trained the entropy of the samples in each batch decreases, reflecting higher confidence in the model's classifications. By leveraging this behavior, the method dynamically adapts to the model's learning progress. Initially, when the model is less confident, the threshold allows a broader range of samples to be considered for pseudolabeling. As training progresses and confidence improves, the threshold tightens, focusing

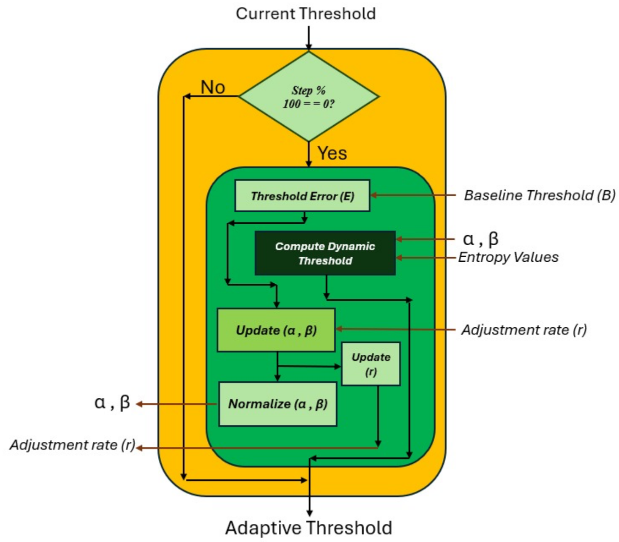


Figure 2: Overview of the Adaptive Thresholding Pipeline

on only the most reliable samples.

$$\tau = \alpha \cdot Q_1 + \beta \cdot Q_3$$

The lower and upper quartiles are Q_1 and Q_3 respectively which are normalized to be between 0 and 1 to ensure stable updates to the threshold.

$$Q_1 = \frac{\text{quantile}_{0.25}(H)}{\max(H)}, \quad Q_3 = \frac{\text{quantile}_{0.75}(H)}{\max(H)}$$

With the threshold computed the next step involves the updates of the α and β weights in the formula. This is done using an adversarial approach as follows where the weights are updated to ensure that their directions are opposite to each other. The idea behind this approach is to allow the model to learn the optimal weights for α and β to avoid over-reliance on either confident or uncertain samples.

$$\alpha = \frac{\alpha - \eta \cdot (\tau - \tau_T)}{\alpha + \beta}, \quad \beta = \frac{\beta + \eta \cdot (\tau - \tau_T)}{\alpha + \beta}$$

3.3 Entropy Guided Pseudolabeling

Traditional pseudo-labeling approaches in domain adaptation involve assigning the most likely class label (highest classifier output) to unlabeled target samples. However, in our approach, we exploit the entropy values of the classifier's output to identify reliable target samples for pseudo-labeling. Since Entropy is a measure of uncertainty in a probability distribution, we assume that low entropy indicates high confidence in the predicted class, while high entropy signifies uncertainty.

Based on these assumptions, we opted for a soft pseudo-label approach since previous studies have shown that such approaches significantly outperform hard pseudo-labels Saporta et al. [2020] due to the loss of information. Given $p(c_i|x_j)$ which represents the predicted probability of class c_i for target sample x_j we calculate the soft pseudo labels using the following formula which essentially normalizes the probabilities for each sample ensuring that they sum to 1.

$$\tilde{y}_{ij} = \frac{p(c_i|x_j)}{\sum_{k=1}^{C} p(c_k|x_j)}$$

With the pseudo labels generated, we assign weights to each sample based on their computed entropy values. This is done through the following formula where $H(x_j)$ represents the computed entropy for sample X_j and τ represents the current value of the Adaptive Threshold. The intuition behind assigning weights stems from the fact that we want to reward correct predictions by assigning greater

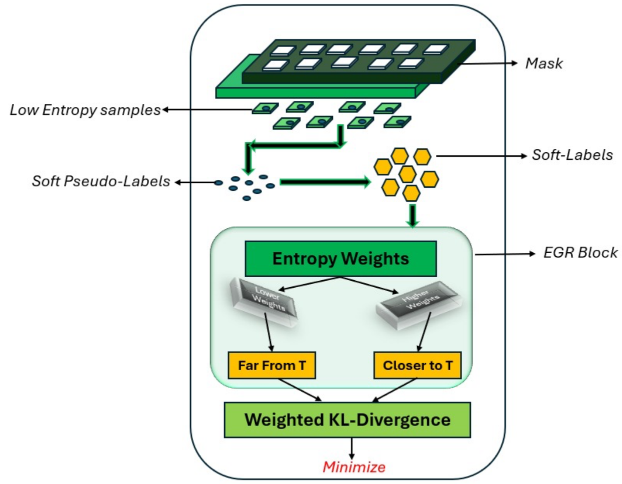


Figure 3: Overview of Entropy Guided Pseudolabeling

weight to low entropy predictions that are far from the threshold value. Similarly, samples with values close to the threshold are assigned lesser weights since the model is still uncertain about their labels.

$$w_j = 1 - \text{clamp}\left(\frac{H(x_j)}{\tau}, 0, 1\right)$$

Lastly, we use Kullback Liebler Divergence Loss(KL) between the log probabilities of the model's output and the weighted soft pseudo labels. This essentially encourages the model to align its predictions with the computed soft pseudo labels and penalizes the model in case it starts to deviate from the pseudo label distribution.

$$\mathcal{L}_{pseudo} = \sum_j w_j \sum_i \tilde{y}_{ij} \log \frac{\tilde{y}_{ij}}{q(c_i|x_j)}$$

This strategy enables the model to gradually adapt to the target domain by incorporating reliable pseudo labels into the training process while minimizing the influence of uncertain samples.

3.4 Dynamic Rejection Loss

To mitigate the negative impact of high entropy samples, we introduced a **Dynamic Rejection Loss (DRL)**. This loss penalizes samples whose entropy values exceed the adaptive threshold by a predefined margin m. This value of this margin is fixed at 0.1 following experimentation to balance penalization of extreme high entropy uncertain samples and borderline samples. The reasoning behind this loss stems from the fact that samples within the margin m are still close enough to the threshold such that it is possible for them to be classified correctly in subsequent iterations. Therefore, such samples are not penalized due to their potential of being re-evaluated. However, those samples that exceed the margin are treated as highly ambiguous and are penalized to encourage the model to reduce uncertainty in its predictions.

This penalty basically influences the network to focus on confidently classifying samples or rejecting them outright as unknown. The loss is defined as follows:

$$\mathcal{L}_{\text{rejection}} = \frac{1}{|H_{\text{high}}|} \sum_{i \in H_{\text{high}}} \begin{cases} -\left|H(p(x_i)) - T_{\text{current}}\right|, & \text{if } |H(p(x_i)) - T_{\text{current}}| > m, \\ 0, & \text{otherwise.} \end{cases}$$

Here:

- H_{high}: Set of high-entropy target samples,
- T_{current}: Threshold used in the current iteration,
- m: Confidence margin (described below).

4 Experimental Design

4.1 Research Questions

The primary research questions guiding the experiment are as follows:

- How can we effectively handle domain shifts by dynamically adjusting thresholds using adaptive entropy-based mechanisms?
- Can pseudo labeling be used for low entropy samples to enhance the model's performance?
- Is it possible to reduce the impact of high entropy samples by training the model to reduce their uncertainty?

4.2 Experimental Setup

4.2.1 Dataset

The proposed framework was evaluated on the following dataset:

- **Office-31**: The Office-31 dataset Saenko et al. [2010] consists of three domains namely **Amazon, DSLR and Webcam** with 31 different classes.

Additionally, the source and target data loaders were both preprocessed to resize images and apply image net normalization and standard deviation. Moreover, we used data augmentation techniques **RandomHorizontalFlip** and **RandomCrop** to enhance the diversity of the data.

4.2.2 Experiments

We evaluated our framework on four different scenarios as follows:

1. **Closed Set**: 31 shared classes, 0 source private classes and 0 target private classes
2. **Open Set**: 10 shared classes, 0 source private classes and 11 target private classes
3. **Partial Set**: 10 shared classes, 21 source private classes and 0 target private classes
4. **Open Partial Set**: 10 shared classes, 10 source private classes and 11 target private classes

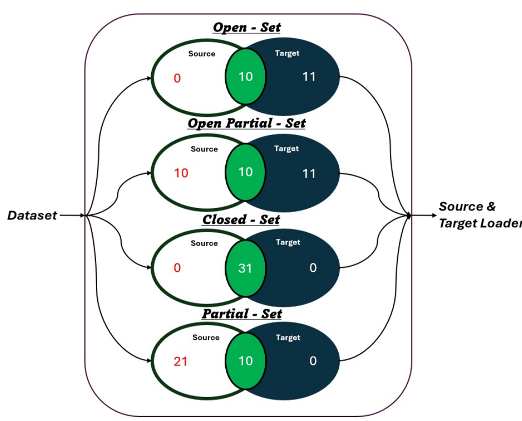


Figure 4: Experiment Scenarios

4.3 Plan for Analysis

Based on the DANCE paper and previous works, we attempt to follow similar evaluation metrics as follows:

1. **Mean Per Class Accuracy(MPCA)**: This essentially is the average of the accuracies of the shared classes plus the "unknown" class provided that the experiment has target private

classes. For each class, the number of correctly classified samples is divided by the total number of samples in that class, and these class-wise accuracies are averaged to obtain the MPCA.

2. **Test Set Accuracy**: This metric measures the overall accuracy of the model on the target dataset. It is computed as the ratio of correctly classified samples to the total number of samples in the test set.
3. **Inference Times**: We analyze the inference times between our model and the DANCE model across the four experimental scenarios to see if our modifications result in extra overhead. This analysis helps determine whether the introduced modifications, such as using an adaptive threshold, dynamic rejection mechanisms, or pseudo-labeling, introduce significant computational overhead.

5 Results and Findings

5.1 Performance

Table 1: Results on Open-Set Domain Adaptation (ODA) for Office-31.

Method	**Office (10 / 0 / 11)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
DANCE(Emulated)	**81/93.6**	**82/96.0**	**90/97.1**	**83/94.8**	**77/91.0**	70/**90.1**	**80/93.7**
AEG-UDA(Ours)	75/89.3	76/93.3	72/93.5	73/90.4	65/79.4	**73**/87.4	72.3/88.9

Table 2: Results on Open Partial Domain Adaptation (OPDA) for Office-31.

Method	**Office (10 / 10 / 11)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
DANCE(Emulated)	**79/92.2**	**92/97.0**	**90/97.1**	**80/91.7**	**83/91.0**	**79/90.3**	**83.8/93.2**
AEG-UDA(Ours)	73/89.0	81/94.5	80/94.1	74/89.6	76/86.5	75/86.6	76.5/88.4

Table 3: Results on Closed Domain Adaptation (CDA) for Office-31.

Method	**Office (31 / 0 / 0)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
DANCE(Emulated)	**85/86.0**	**98/97.0**	98/97.1	**87/89.7**	**62/62.0**	**61/60.9**	**81.9/82.1**
AEG-UDA(Ours)	67/66.3	82/82.0	**98/97.6**	72/74.2	52/51.5	52/51.1	70.5/70.5

Table 4: Results on Partial Domain Adaptation (PDA) for Office-31.

Method	**Office (10 / 21 / 0)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
DANCE(Emulated)	**29/97.5**	**29/99.5**	**31/100**	**29/92.0**	**22/73.7**	**22/74.1**	**27/89.4**
AEG-UDA(Ours)	22/73.1	27/93.1	**31/100**	20/58.8	18/61.5	15/51.1	22.2/80.0

The tables above present results comparing the emulated **DANCE** baseline with our proposed *AEG-UDA* model across four domain adaptation settings: *Open-Set Domain Adaptation (ODA)*, *Open Partial Domain Adaptation (OPDA)*, *Closed Domain Adaptation (CDA)*, and *Partial Domain Adaptation (PDA)*. In each cell, the left value represents the **Test Accuracy**, which includes classes from unknown domains, while the right value is the **Mean Per-Class Accuracy**, an important metric referenced in prior works, such as the **DANCE** paper. Although test accuracy is not a standard metric in prior benchmarks, it was included in our evaluation to provide a more comprehensive understanding of the models' behavior across known and unknown classes.

The results reported here are not directly derived from any prior work but are based on experiments we conducted ourselves. Both models, **DANCE** and **AEG-UDA**, were evaluated for 2000 steps under identical controlled environments. The hyperparameter settings were determined after exhaustive testing. While more optimal configurations may exist, we report the settings that yielded the best results in our experiments. The hyperparameters used are as follows:

- $\alpha = 0.5$
- $\beta = 0.5$
- Baseline thresholds:
  - 1.15 for **ODA**
  - 1.49 for **OPDA**
  - 1.71 for **CDA** and **PDA**
- Confidence margin: 0.1
- Initial adjustment rate for dynamic thresholding: 0.01

Our experimentation compares **AEG-UDA**, which incorporates dynamic thresholding and entropy-guided learning, against **DANCE**, which uses fixed thresholds and memory-based alignment mechanisms. These methods were tested on the Office-31 dataset under all four adaptation scenarios.

While our model, **AEG-UDA**, does not explicitly outperform **DANCE** on the reported metrics, it comes significantly close in a fair comparison. A key factor contributing to this difference is that **DANCE** employs mixed-precision training, which enhances stability and mitigates fluctuations in performance. In contrast, **AEG-UDA** uses full-precision training, which, while effective in some cases, results in performance variability across different seeds. Notably, although **AEG-UDA** outperforms **DANCE** on certain seeds, it does not achieve higher average performance across the four seeds reported. This highlights the trade-off between stability and adaptability, with mixed-precision training providing a consistent advantage to **DANCE**. Despite this, **AEG-UDA** demonstrates its potential by achieving competitive results in several scenarios, validating its design and approach for handling diverse domain adaptation challenges.

5.2 Inference

Table 5: Inference Average Time Results for Different Adaptation Scenarios.

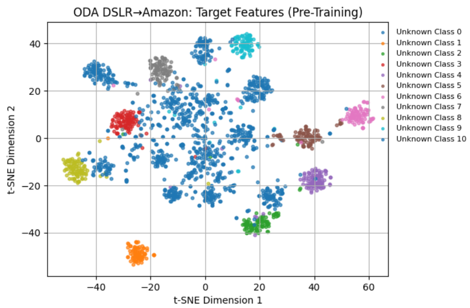
Type	Method	A2W	D2W	W2D	W2A	A2D	D2A	Avg(s)
ODA	DANCE	0.013884	**0.014578**	**0.014724**	**0.014680**	**0.014213**	**0.013742**	**0.0143035**
	AEG-UDA	**0.013448**	0.014882	0.015083	0.014827	0.014521	0.013882	0.0144405
OPDA	DANCE	0.013639	**0.014323**	**0.014388**	**0.014107**	**0.014083**	**0.013194**	**0.0139557**
	AEG-UDA	**0.013390**	0.014605	0.014514	0.014446	0.014392	0.013389	0.0141227
CDA	DANCE	0.012987	**0.014058**	**0.014025**	0.013585	0.013428	0.013048	0.0135218
	AEG-UDA	**0.012894**	0.014121	0.014308	**0.013506**	**0.013343**	**0.012805**	**0.0134962**
PDA	DANCE	0.013076	**0.014096**	0.014169	**0.013729**	0.013652	**0.012975**	0.0139495
	AEG-UDA	**0.012973**	0.014210	**0.014142**	0.013765	**0.013483**	0.013254	**0.0136825**

Upon analyzing the average inference time over 2000 steps (Table 5), our proposed model, AEG-UDA, outperforms DANCE in two scenarios: CDA and PDA, while achieving comparable results with a minor difference in ODA and OPDA. The improved inference times in CDA and PDA can be attributed to the dynamic threshold mechanism introduced in AEG-UDA, which updates adaptively every 100th step. This flexibility reduces computational overhead by minimizing the number of unnecessary updates, resulting in a more efficient processing pipeline. Despite the fluctuating nature of AEG-UDA, the ability to dynamically adjust thresholds ensures better optimization of resources, leading to a reduction in overall inference time compared to DANCE, which relies on fixed thresholds.

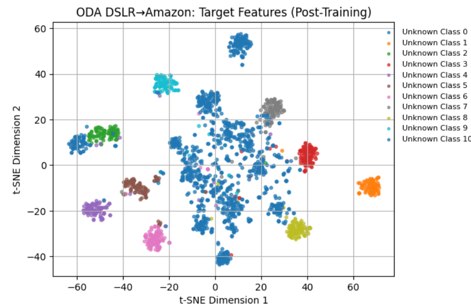
5.3 T-SNE Visualization

5.3.1 Cluster Compactness

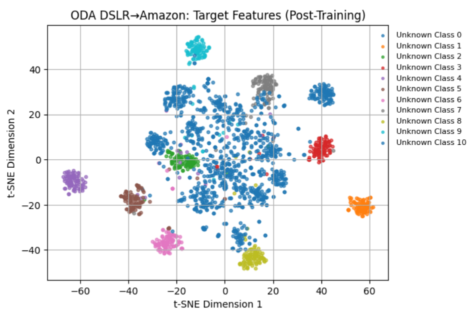
In Figure 5, (c), it can be seen that the clusters are more tightly packed as compared to DANCE-emulated (b). This can be attributed to the entropy guided pseudo-labeling which helps reduce intra-cluster variance, resulting in more compact clusters.



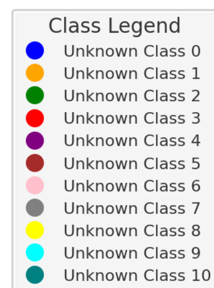
(a) Baseline (Pre-Training).



(b) Post-training (DANCE-emulated).



(c) Post-training (AEG-UDA).



(d) Legend for t-SNE visualizations.

Figure 5: t-SNE visualizations of target features (across 11 classes) in the DSLR $\rightarrow$ Amazon ODA setting. (a) Baseline pre-training, (b) DANCE-emulated post-training, (c) AEG-UDA post-training, and (d) the legend for visualizing the classes. AEG-UDA demonstrates better performance in feature alignment and class separation.

5.3.2 Cluster Separation

Figure 5, (c) also shows less overlap for clusters as compared to DANCE-emulated, where more of these clusters tend to blend into each other. This would be a direct result of adaptive thresholding and DRL. DRL would help pull high entropy samples away from cluster boundaries thus reducing overlap and the adaptive threshold adjusts to the target domains entropy distribution, which enables better discrimination between low and high entropy samples, preventing mixing of uncertain samples into multiple clusters.

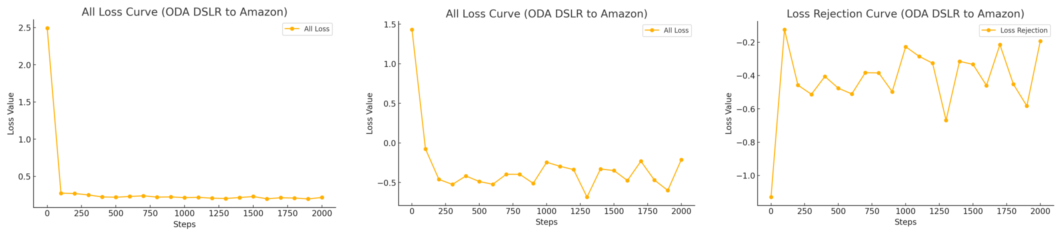
5.3.3 Outliers

The AEG-UDA visualization, (c), in figure 5 shows fewer scattered outliers than DANCE-emulated (b). Both DRL and EGPL help to achieve this as they both work to distinctly refine the contribution of low and high entropy samples, providing a more robust setup than DANCE-emulated's collective approach.

5.4 Loss Curves

The Dance-emulated framework uses neighborhood clustering and entropy separation loss which contribute to its loss curve, they directly align target features to source features. This approach lacks mechanisms to explicitly handle high-entropy samples.

AEG-UDA builds on this and adds dynamic rejection loss. The oscillations seen in its combined loss curve are introduced from the dynamic rejection loss. It penalizes samples with high entropy values exceeding the adaptive threshold. Its behavior is expected because as the threshold changes during training, high-entropy samples vary in frequency and uncertainty as the model learns, leading to fluctuations. This is unique to AEG-UDA and allows for robust handling of ambiguous samples.



(a) Dance-emulated All Loss. (b) AEG-UDA All Loss. (c) Dynamic Rejection Loss.

Figure 6: Comparison of loss curves in the ODA DSLR to Amazon setting. (a) Dance-emulated All Loss, (b) AEG-UDA All Loss, and (c) AEG-UDA Dynamic Rejection Loss. These curves highlight the distinct training dynamics of the models.

The continuous oscillations reflect the ongoing refinement of the model's ability to separate known from unknown classes.

6 Discussion

The results presented in this study show the potential of the Adaptive Entropy Guided Universal Domain Adaptation (AEG-UDA) framework in addressing the inherent challenges of domain adaptation. By dynamically adjusting entropy thresholds, leveraging entropy-guided pseudo-labeling, and incorporating a novel Dynamic Rejection Loss (DRL), our approach refines the alignment between source and target domains while reducing the impact of high-entropy samples. These innovations demonstrate promise, as evidenced by the results achieved across multiple domain adaptation scenarios, including Closed Set, Open Set, Partial Set, and Open Partial Set.

6.1 Strengths and Contributions

A key strength of AEG-UDA lies in its ability to dynamically adapt to the target domain's entropy distribution, a shift from the static thresholds used by approaches like DANCE, upon which we built. This adaptive mechanism improves the model's robustness by guiding its behavior to evolving learning conditions, improving class separation, and reducing the influence of ambiguous samples. Additionally, the entropy-guided pseudo-labeling strategy allows for effective utilization of low-entropy samples, while DRL minimizes the negative impact of high-entropy samples, allowing for more confident predictions. These contributions not only extend the capabilities of existing domain adaptation methods but also provide a more nuanced approach to handling uncertainty in unlabeled target domains.

6.2 Limitations

While the AEG-UDA framework demonstrates competitive performance, its reliance on full-precision training introduces variability in results, as observed in the sensitivity to random seeds. In contrast, DANCE's mixed-precision training yields more stable performance, highlighting an area for potential refinement. Additionally, the current implementation of AEG-UDA does not consistently surpass DANCE across all metrics, suggesting that further optimization of hyperparameters, particularly for the adaptive threshold and DRL, could enhance overall performance. Computational constraints limited the exploration of all alternative configurations and additional datasets, which would provide a more comprehensive evaluation of the framework's generalizability.

6.3 Future Work

Future research could aim to integrate mixed-precision training into the AEG-UDA framework to improve stability and reduce performance variability. Expanding the scope of the framework to incorporate larger and more diverse datasets would provide deeper insights into its generalizability across real-world scenarios. Evaluating AEG-UDA in dynamic environments, such as time-varying domains or streaming data, could open avenues for extending its adaptive capabilities.

Another potential direction involves refining the Dynamic Rejection Loss (DRL) to handle highly uncertain samples more effectively. Introducing more sophisticated weighting schemes for DRL, driven by uncertainty estimation or external guidance mechanisms, could enhance its ability to minimize the impact of high-entropy samples.

The exploration of hybrid entropy mechanisms, combining global and local entropy measures, could further improve the separation of confident and ambiguous predictions. Additionally, advancing the pseudo-labeling process by leveraging semi-supervised or self-supervised learning techniques might result in more reliable predictions for low-entropy samples.

7 Conclusion

This project introduced the Adaptive Entropy Guided Universal Domain Adaptation (AEG-UDA) framework, which builds upon the DANCE architecture by incorporating multiple techniques to address the challenges of Universal Domain Adaptation. Through the use of adaptive thresholding, entropy-guided pseudo-labeling, and a novel Dynamic Rejection Loss (DRL), AEG-UDA effectively separates confident predictions from ambiguous samples.

Our experiments across the Open, Closed, Partial, and Open Partial set scenarios demonstrated the framework's ability to dynamically adjust to the target domain's entropy distribution, resulting in improved feature alignment and class separation. While AEG-UDA achieves competitive performance, it highlights the trade-offs between adaptability and stability, especially when compared to static-threshold-based models like DANCE. Despite limitations such as reliance on full-precision training and sensitivity to random seeds, the framework showcases the potential of dynamically addressing uncertainty in domain adaptation tasks.

Future work could focus on refining the adaptive threshold mechanism, exploring alternative pseudo-labeling strategies, and integrating mixed-precision training to enhance both performance stability and computational efficiency.

References

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 754–763, 2017. doi: 10.1109/ICCV.2017.88.

Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation, 2018. URL https://arxiv.org/abs/1808.04205.

Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation, 2019. URL https://arxiv.org/abs/1903.12230.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, page 213–226, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 364215560X.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision, 2020. URL https://arxiv.org/abs/2002.07953.

Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Esl: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation, 2020. URL https://arxiv.org/abs/2006.08658.

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2715–2724, 2019. doi: 10.1109/CVPR.2019.00283.

Table 6: Paper Reported Results on Open-Set Domain Adaptation (ODA). SO Saenko et al. [2010]. DANN Ganin et al. [2016]. ETN Cao et al. [2019]. STA Saporta et al. [2020]. UAN You et al. [2019]. DANCE Saito et al. [2020]. AEG-UDA is our proposed method.

Method	**Office (10 / 0 / 11)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
SO	83.8	95.3	95.3	89.6	86.5	82.9	89.1
DANN	87.6	90.5	91.2	88.7	87.1	86.4	88.7
ETN	86.7	90.0	90.1	89.1	87.6	88.3	88.2
STA	91.7	94.1	94.8	91.8	88.1	88.4	91.5
UAN	88.0	95.8	94.8	91.6	88.0	89.4	91.0
DANCE	**93.6**	**97.0**	**97.1**	**95.7**	90.1	**90.3**	**94.1**
DANCE(Emulated)	**93.6**	96.0	**97.1**	94.8	**91.0**	90.1	93.7
AEG-UDA (Ours)	89.3	93.3	93.5	90.4	79.4	87.4	88.9

Table 7: Paper Reported Results on Open Partial Set Domain Adaptation (OPDA).

Method	**Office (10 / 10 / 11)**						**Avg**
	A2W	**D2W**	**W2D**	**A2D**	**D2A**	**W2A**	
SO	75.7	95.4	95.2	83.4	84.1	86.4	86.4
DANN	87.6	90.5	91.2	88.7	87.4	87.0	88.7
ETN	89.1	90.6	90.9	89.1	88.6	88.3	88.3
STA	85.2	96.3	95.1	88.5	87.1	87.9	89.8
UAN	76.2	82.0	82.0	84.6	85.2	84.1	84.1
DANCE	**92.8**	**97.8**	**97.7**	91.6	**92.2**	**91.4**	**93.9**
DANCE(Emulated)	92.2	97.0	97.1	**91.7**	91.0	90.3	93.2
AEG-UDA (Ours)	89.0	94.5	94.1	89.6	86.5	88.6	88.4

A Appendix

Ablation Studies

1. Adaptive Thresholding

In our ablation studies for adaptive thresholding, we began with a **basic mean and standard deviation approach**, assigning weights α and β to these metrics. To improve stability, we introduced a **momentum factor** that prevented extreme values by smoothing updates over time. Building on this, we implemented a **decay rate** to allow α and β to vary exponentially based on the number of steps. Subsequently, we experimented with a **small neural network** to predict α and β, enabling a more dynamic and data-driven approach. Finally, we arrived at the **entropy quartiles approach**, which computes the adaptive threshold using quartiles of entropy values. This method was selected as it consistently provided the best results across our experiments.

2. Pseudolabeling of Low-Entropy Samples

For the pseudolabeling of low-entropy samples, we initially adopted a **basic argmax approach**, directly using hard labels for these samples. We then transitioned to an **entropy-aware pseudolabeling approach**, utilizing KL divergence to measure the distance of each sample from the threshold. Exploring further, we investigated **cluster-based entropy** and **centroid-based cluster approaches**, inspired by the DANCE protocol, to measure divergence or distance from entropic clusters. Ultimately, we converged on the **entropy-guided refinement approach**, which assigns different weights to samples based on their distance from the threshold. This approach was chosen as it demonstrated the most robust performance and provided the best results for fine-tuning low-entropy samples.

3. Dynamic Rejection Loss

In our application studies for dynamic rejection loss, we began with a **basic approach** of determining the confidence margin and assigning a zero loss for samples within this margin. We then explored various strategies, including **limiting the effect of the loss**, **preventing its influence under specific conditions**, and **decaying the confidence margin** over a set number of calculations. However, through rigorous experimentation, we found that the **initial approach of determining the confidence margin and assigning a zero loss** consistently yielded the best results, effectively balancing stability and adaptability.