Rapid Disaster Damage Assessment Using Deep Adversarial Sliced Wasserstein Domain Adaptation

Fatma AlNaimi[1], Abdulaziz Al-Homaid[2], Ferda Ofli[2], Abdelkader Baggag[2*]

[1]Information and Computing Technology Division, Hamad Bin Khalifa University, Education City, Doha, 34110, Qatar.
[2]Qatar Computing Research Institute, Hamad Bin Khalifa University, HBKU Research Complex, Doha, 34110, Qatar.

*Corresponding author(s). E-mail(s): abaggag@hbku.edu.qa;
Contributing authors: fat.nuaimi@gmail.com; abalhomaid@hbku.edu.qa; fofli@hbku.edu.qa;

Abstract

Disasters vary greatly in their nature, severity and underlying distribution. A streamlined model is crucial for assessing the impact of unexpected disasters. Previous methods rely on training with historical disaster data to evaluate the damage caused. However, in practice, this approach rarely achieves acceptable performance due to domain shift. Therefore, existing models need to be adapted to the emergent disaster quickly. A promising way to achieve this goal is through unsupervised domain adaptation (UDA). To this end, many advancements have been made toward better transferability of the model, including the Domain Adversarial Neural Network (DANN), which utilizes adversarial training to attain a domain-invariant feature extractor. The Adversarial sliced Wasserstein Domain Adaptation Network (AWDAN) further improves on DANN by using sliced Wasserstein distance as a measure between the features extracted from the source and target domains. Inspired by these advancements, we explore the utility of UDA approaches in the disaster response domain. Specifically, we perform extensive experiments on real-world images collected from Twitter during four major disasters. We train a total of 216 models to benchmark six methods across all possible source–target domain combinations. We improve the performance of the previous state-of-the-art DANN-based method for rapid damage assessment by enhancing it with a deeper backbone architecture to learn better feature representations. Furthermore, we adopt AWDAN to more effectively mitigate the distribution shift in data obtained from different disaster events.

Experimental results demonstrate that the proposed approach achieves statistically significant performance gains, with up to 11.4% improvement in F1-score and 8.9% improvement in accuracy over the source-only model, consistently outperforming several state-of-the-art domain adaptation frameworks, including DANN, CORAL, MMD, and CDAN.

Keywords: Domain adaptation, Adversarial learning, Sliced Wasserstein, Distribution shift, Disaster datasets, Damage assessment

1 Introduction

Rapid disaster damage assessment is a critical component of effective humanitarian response. It enables organizations to prioritize resources, plan rescue operations, and allocate aid where it is needed most. Traditional damage assessment methods, such as ground surveys or manual analysis of aerial images, are often slow, resource-intensive, and infeasible during large-scale disasters. However, when a disaster occurs, abundant information becomes available on social media in the form of text messages and images. If captured and analyzed efficiently, such information can be extremely valuable for rapid disaster response. Manually studying this information can be time- and resource-consuming whereas machine learning-based approaches can automate the process and alleviate human effort. Some studies process social media text data to identify disaster footprints and hot spots, whereas others analyze social media images to assess disaster impact.

Most of these studies work well on the datasets that they were trained on in a supervised manner. However, a model trained on past disaster data does not generalize well on data coming from a new disaster. Even if the disaster type is the same, there is usually a shift in data characteristics (i.e., distribution) that causes existing models to underperform on unseen events. This shift can be due to many factors including differences in scale of the impact, time of the year, or geographic location. In this situation, it is unrealistic to expect to collect and annotate additional data manually from the emergent disaster event and train new models, especially when a rapid assessment of the impact is vital in the early hours of the disaster. Therefore, a solution based on unsupervised domain adaptation is more plausible since it can deal with the domain shift problem by leveraging available datasets and models from past disasters.

Unsupervised Domain Adaptation (UDA) is a field in machine learning where the target domain labels are not needed during training, reflecting a real-life situation where classifying many unlabeled images from a new domain using available models. Figure 1 illustrates a typical UDA scenario in the context of disaster response. On the left hand side, we have images shared on social media during Hurricane Matthew, categorized into two groups: “damage” and “no damage.” This collection is referred to as the Hurricane dataset. On the right, we have a similar set of images shared during the Ecuador Earthquake. Unlike the Hurricane dataset, these images are unlabeled and unorganized, forming what we call the Earthquake dataset. In this setting, the

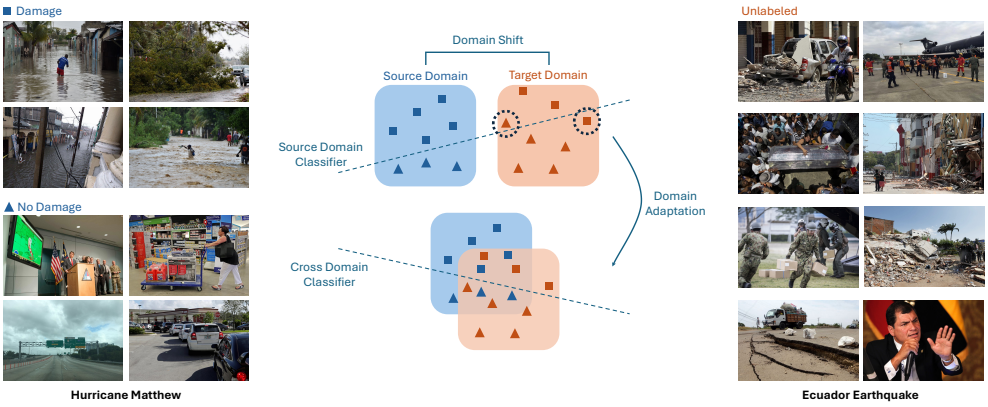

Fig. 1: An example of domain adaptation setting for rapid damage assessment

Hurricane dataset serves as the source domain, providing labeled data, while the Earthquake dataset represents the target domain, which lacks labels. UDA seeks to leverage the labeled source data to train a model that can generalize effectively to the target domain. By training the model on both the source and target datasets—using only the source labels—it becomes capable of predicting labels for the target dataset. This approach offers a practical and scalable solution for rapid disaster damage assessment across different disaster scenarios.

Many algorithms have been proposed for UDA following different approaches. For instance, divergence-based methods aim to align the distributions of source and target domain features by measuring the statistical discrepancy between them [e.g. see 1, 2] while reconstruction-based methods typically use an auxiliary reconstruction network to learn an invariant representation between source and target domains [e.g. see 3, 4]. On the other hand, adversarial-based methods employ an auxiliary domain discriminator network to measure domain discrepancy and learn domain-invariant feature representations by forcing the domain discriminator to fail at distinguishing the source and target domains [e.g. see 5, 6]. While the divergence-based approaches require determining the distance metric to consider the additional loss terms and tuning-related hyper-parameters explicitly, the reconstruction-based methods necessitate training of computationally expensive reconstruction sub-networks. And finally, adversarial-based approaches combine domain adaptation and deep feature learning within one training process which is susceptible to mode collapse.

These UDA algorithms have been successfully adopted in numerous fields such as healthcare [e.g. see 7, 8], agriculture [e.g. see 9] and autonomous driving [e.g. see 10] among others. Specifically in the crisis informatics field, UDA methods have been predominantly explored in NLP-based [e.g. see 11] and remote sensing-based studies [e.g. see 12]. However, their adoption for analysis of social media images in the disaster context has been limited to only a few examples [13–15]. For instance, [14] explored a divergence-based UDA method using first-order statistics (i.e., maximum mean discrepancy (MMD) [16]) for classifying social media images as *informative* and *not-informative* for humanitarian purposes, and later, they extended their work to

multi-source domain adaptation [15]. Whereas [13] applied an adversarial-based UDA method called Domain Adversarial Neural Network (DANN) [6] to adapt model features from source (past) disaster to target (emergent) disaster for damage severity classification (i.e., *damage* and *no-damage*).

While these studies reduce the distribution shift across domains by learning domain-invariant feature representations, they cannot guarantee that the category distributions are also aligned (i.e., features belonging to the same class are mapped nearby) across domains. To tackle this limitation, the Adversarial Sliced Wasserstein Domain Adaptation Network (AWDAN) [17] improves DANN [6] by adding a new error measure based on the Wasserstein distance between the label classification output of source and target features so that the learned features are more discriminative with better inter-class separability and intra-class compactness characteristics. Inspired by this trend, in this study we perform a comprehensive experimentation with several state-of-the-art UDA methods including DANN and AWDAN to explore the utility of UDA for rapid damage assessment during an emergent disaster. Put specifically, we extend the prior work [13] by (i) employing a ResNet [18] backbone instead of a VGG [19] backbone in DANN, (ii) further adopting a more recent and sophisticated approach, i.e., AWDAN, and (iii) providing baseline comparisons with other popular UDA methods such as CORAL [20], MMD [21], and CDAN [22] for the task at hand. Our results show that we can improve the overall accuracy on the disaster damage assessment task against prior art [13] by 5.9% when we employ DANN with a ResNet backbone, and by 7.6% when we employ AWDAN with a loss based on optimal transport, computed efficiently using the sliced Wasserstein approach [23]. Additionally, the resulting model achieves roughly 1-3% improvement in F1-score and accuracy over the benchmark UDA methods (i.e., CORAL, MMD, CDAN), thanks to its ability to seek alignment *both* in the feature space and in the label space.

In summary, the main contributions of the paper are as follows:

1. We improve the performance of the previous state-of-the-art DANN-based method [13] by replacing its backbone with a deeper network architecture to learn better feature representations. This leads to an average improvement of 6.4% in F1-score (i.e., 77.7 vs. 82.7) and 5.9% in accuracy (i.e., 76.1 vs. 80.6).
2. We adopt the Adversarial Sliced Wasserstein Domain Adaptation Network (AWDAN) for rapid disaster damage assessment for the first time to better mitigate the distribution shift in data obtained from different disaster events. This leads to a further improvement of 1.8% in F1-score (i.e., 82.7 vs. 84.2) and 1.6% in accuracy (i.e., 80.6 vs. 81.9).
3. The proposed method achieves the best performance with an average improvement of 11.4% in F1-score (i.e., 75.6 vs. 84.2) and 8.9% in accuracy (i.e., 75.2 vs. 81.9) over the source-only (lower bound) model. It outperforms the prior art [13] by 8.4% in F1-score and 7.6% in accuracy as well as other strong baselines such as CORAL, MMD, and CDAN by 1.2-2.9% in F1-score and 1.0-3.0% in accuracy.

This manuscript extends the work in [24], which initially investigated the use of adversarial Sliced Wasserstein domain adaptation for disaster image analysis. Building

upon that foundation, the present study performs extensive experiments on real-world images collected from Twitter (now X) during four major disasters. A total of 216 models are trained to benchmark six domain adaptation methods across all possible source-target domain combinations, offering a broad empirical assessment of cross-domain performance. In addition, the study broadens the scope of prior work by employing a deeper backbone architecture to enhance feature representation learning and conducting an extensive comparative analysis against six state-of-the-art domain adaptation frameworks. These advancements collectively strengthen the methodological rigor and empirical significance of the research.

The rest of the paper is organized as follows. Section 2 provides a brief overview of the existing work. Section 3 introduces the methodology used in this study. Section 4 presents the experimental results and analyses. Section 5 provides a discussion on the results and findings. Finally, we conclude the paper in Section 6.

Reproducibility. To support independent replication and further research, we publicly release the source code, datasets, trained model weights, and detailed reproduction instructions. The code is available at https://github.com/abalhomaid/disaster-assesment, and the datasets/model weights are available at https://huggingface.co/datasets/abalhomaid/disaster-damage-assessment.

2 Related Work

This study explores the use of unsupervised domain adaptation for crisis informatics. Hence, we present a brief overview of the literature on these two topics separately.

2.1 Crisis Informatics

Crisis informatics is regarded as the combination of computation and social science to address crisis-related problems [e.g. see 25]. After the widespread adoption of social media platforms, a major focus of research has been on processing the social media data streams during emergencies. Early studies in this domain relied predominantly on analyzing the *textual* social media content to accomplish tasks such as event detection, eyewitness identification, crisis communication, and gathering actionable information [26–30]. Furthermore, many notable systems have been developed in this space [31–33].

Later on, there has been a growing interest in the analysis of social media *visual* content during crisis. Many studies have shown that social media images can play a significant role in reducing information overload, detecting disaster events, and damage assessment [34–36]. Such models were also deployed in many real-time systems used by emergency responders [e.g. see 37]. Most recently, multimodal learning frameworks have been adopted to gain information about disasters by jointly analyzing social media text messages and images [e.g. see 38, 39].

It is important to note that most of the aforementioned studies follow a supervised learning paradigm. However, gathering annotations through crowd-sourcing for training supervised machine learning models remains to be challenging, especially during emergencies, in contrast to the ease of collecting large volumes of social media images. On the text analysis side, there are a few approaches in crisis informatics aiming to

automatically adapt the existing models to new datasets with limited labels or without labels [11, 40–43]. However, on the image analysis side, research has been limited to a handful of studies [13–15]. For instance, [14, 15] presented a divergence-based UDA method based on maximum mean discrepancy (MMD) to identify *informative* social media images whereas [13] used adversarial domain adaptation in the feature space to classify damage severity in images collected from four natural disasters including earthquakes and hurricanes. Unlike these methods, this study minimizes the discrepancy between source and target domains by aligning distributions both in the feature and label space using an efficient optimal transport metric such as sliced Wasserstein distance [23].

Although the above studies motivate image-based disaster assessment, our focus is on *cross-event transfer*. We therefore connect this line of work to generic UDA frameworks that can be adapted to noisy, rapidly evolving social-media imagery, and evaluate them under a protocol tailored to disaster damage classification.

2.2 Unsupervised Domain Adaptation

Machine learning models trained in a supervised fashion on a (seen) source domain tend to perform worse on a new (unseen) target domain due to domain shift [44]. To remedy this, unsupervised domain adaptation (UDA) aims to enable cross-domain learning without target domain labels by transferring knowledge from a labeled source domain. There are three main class of UDA approaches: divergence, reconstruction, and adversarial.

Divergence-based methods align the distributions of source and target domains by the distance between their data distributions using first-order (mean) [1, 45, 46], second-order (covariance) [20, 44, 47, 48], or higher-order statistics [2, 49, 50]. These methods depend on selection of the appropriate distance metric. To this end, maximum mean discrepancy (MMD) [16] has been widely used to check if two sample points belong to the same distribution [21, 51], but other distance metrics such as Wasserstein distance [23] has also been used to guide the domain alignment process [52, 53]. For instance, Correlation Alignment (CORAL) [20] minimizes domain shift by using linear transformation of its original distributions to align the second-order statistics of the source and target feature distributions whereas Maximum Mean Discrepancy (MMD) [21] is used as a regularizer to align the distributions among different domains to an arbitrary prior distribution.

Reconstruction-based methods use an auxiliary reconstruction network to learn a shared (i.e., invariant) representation between source and target domains. These models are simultaneously optimized for both classification of the source data and reconstruction of the unlabeled target data [3, 4, 54, 55]. Alternatively, image-to-image translation [56, 57] can be employed to align the source and target domain feature representations [58–60].

Adversarial-based methods train a generator and a discriminator to differentiate between source and target datasets to learn invariant features for both domains, and thus, improve the model accuracy [5, 6, 22, 61–65]. As one of the earlier methods, Domain Adversarial Neural Network (DANN) [6] performs adversarial training of the feature extractor to confuse the discriminator, aiming to align the features

of source and target domains. Alternatively, Conditional Adversarial Domain Adaptation (CDAN) [22] employs conditional GANs to impose multilinear conditioning on the discriminator and generator on discriminative information for a better match across different domains. Adversarial Sliced Wasserstein Domain Adaptation Network (AWDAN) [17] improves DANN by simultaneously minimizing the sliced Wasserstein distance in the label space to enforce the generated features to be discriminative, so that to guarantee the transfer performance.

While DANN and AWDAN are general-purpose UDA frameworks, our interest lies in their application to *image-based disaster assessment*, where models must transfer across events quickly and reliably under social-media noise and class imbalance. We therefore adapt and assess these methods under a unified backbone and evaluation protocol tailored to cross-event damage classification. Under the same ResNet50 backbone and evaluation protocol, we additionally include CORAL [20], MMD [21], and CDAN [22] as representative baselines for moment-matching and conditional adversarial alignment.

Within image-based disaster assessment on ground-level/social-media imagery, we prioritize methods that are directly comparable to our setting: cross-event damage classification with label-preserving shifts. In this context, we focus on DANN [6] and AWDAN [17] because they offer well-established, computationally efficient alignment mechanisms (adversarial invariance and sliced-Wasserstein geometry) that are practical for rapid deployment during emergencies.

3 Methodology

This section presents the approach of domain adaptation neural networks for data from different-but-close distributions. We elaborate particularly on two techniques, namely, Domain Adversarial Neural Networks (DANN) [6] and Adversarial sliced Wasserstein Domain Adaptation Network (AWDAN) [17].

To enhance clarity and reproducibility, we provide below a detailed formulation of each domain adaptation component along with explicit optimization objectives and gradient update rules. The notation is standardized across all subsections, and each loss term is defined in full. Furthermore, the corresponding training procedure, summarized in Algorithm 1, specifies projection sampling, loss computation, and parameter updates for complete reproducibility. These additions ensure that both the theoretical foundations and the implementational details are fully transparent.

3.1 Domain Adversarial Neural Networks

Given labeled samples from a source distribution $\boldsymbol{\mathcal{D}}_s$ and unlabeled samples from a target distribution $\boldsymbol{\mathcal{D}}_t$, the goal of unsupervised domain adaptation is to learn a function that solves the task for both the source and target domains. In this paper, we consider classification tasks $\varphi\colon \boldsymbol{\mathcal{X}} \to \mathcal{Y}$, where $\boldsymbol{\mathcal{X}}$ is the input space and $\mathcal{Y} = \{0, 1, ..., L-1\}$ is the set of L possible labels. In particular, the proposed model is trained on both source and target data jointly over $\boldsymbol{\mathcal{X}} \times \mathcal{Y}$, i.e., with the assumption that both domains share the label space and have different distributions, which lead to domain shift. The model,

therefore, aims to directly learn an aligned representation of the domains, in the feature space, while retaining meaningful information with respect to the source labels. The ultimate goal is to train a cross-domain model $p(y \mid \boldsymbol{\mathcal{X}}; \theta)$ with parameters θ that can predict the label of any data point in the target dataset $\boldsymbol{\mathcal{D}}_t$ without having any information about class labels in $\boldsymbol{\mathcal{D}}_t$, simply by using the knowledge from the source distribution $\boldsymbol{\mathcal{D}}_s$. In this setting, the hypothesis class is instantiated by a deep neural network composed of a feature extractor G_f and a label classifier G_y, which together implement the conditional distribution $p(y \mid \mathcal{X}; \theta)$ on both domains. The central difficulty lies in learning representations for which the induced feature distributions of $\mathcal{D}_s$ and $\mathcal{D}_t$ are sufficiently close, while still preserving discriminative information for the supervised task on the source domain. This intuition is formalized by domain adaptation theory, which relates the target risk to the source risk and a discrepancy term between $\mathcal{D}_s$ and $\mathcal{D}_t$, and motivates the adversarial alignment strategies developed in the following subsections.

Several theoretical studies of the domain adaptation problem have proposed upper bounds on the risk of the target domain, involving the risk on the source domain and the distance between the source and target distributions, $\boldsymbol{\mathcal{D}}_s$ and $\boldsymbol{\mathcal{D}}_t$. We justify these methods intuitively by assuming that we expect the source risk to be a good indicator of the target risk if both distributions are similar, in some sense. To this end, one of the popular methods has been proposed by Ganin et al. [6] where they tackle the unsupervised domain adaptation problem by introducing a domain adversarial training scheme. Here, we specifically consider the seminal works of Ben-David et al. [66, 67] where the $\mathcal{H}$-divergence between two distributions is defined.

Definition 1 ($\mathcal{H}$-divergence [66, 67]) *Given two domain distributions $\boldsymbol{\mathcal{D}}_s$ and $\boldsymbol{\mathcal{D}}_t$ over $\boldsymbol{\mathcal{X}}$, and a hypothesis class $\mathcal{H}$ (assumed to be binary), the $\mathcal{H}$-divergence between $\boldsymbol{\mathcal{D}}_s$ and $\boldsymbol{\mathcal{D}}_t$ is*

$$d_{\mathcal{H}}(\boldsymbol{\mathcal{D}}_s, \boldsymbol{\mathcal{D}}_t) = 2 \sup_{h \in \mathcal{H}} \Big| \operatorname*{Prob}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{D}}_s} (h(\boldsymbol{x}) = 1) - \operatorname*{Prob}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{D}}_t} (h(\boldsymbol{x}) = 1) \Big|, \tag{1}$$

$$= 2 \sup_{h \in \mathcal{H}} \Big| \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{D}}_s} \Big[\mathbb{1}[h(\boldsymbol{x}) = 1] \Big] - \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{D}}_t} \Big[\mathbb{1}[h(\boldsymbol{x}) = 1] \Big] \Big|, \tag{2}$$

where $\mathbb{1}[a]$ is the indicator function which is 1 if the predicate a is true, and 0 otherwise.

It is difficult to estimate the supremum over the hypothesis class $\mathcal{H}$. Ganin et al. [6] suggested that even if it is generally hard to compute $d_{\mathcal{H}}(\boldsymbol{\mathcal{D}}_s, \boldsymbol{\mathcal{D}}_t)$ exactly (e.g., when $\mathcal{H}$ is the space of linear classifiers on $\boldsymbol{\mathcal{X}}$), we can easily approximate it by running a learning algorithm on the problem of discriminating between source and target domains. Therefore, it is proposed to train a domain classifier h on samples $\boldsymbol{x}_s \in \boldsymbol{\mathcal{S}}$ with label 0 and $\boldsymbol{x}_t \in \boldsymbol{\mathcal{T}}$ with label 1, i.e., we construct a new (domain labeled) dataset

$$\boldsymbol{\mathcal{U}} = \{(\boldsymbol{x}_s, 0)\}_{s=1}^{n_s} \cup \{(\boldsymbol{x}_t, 1)\}_{t=1}^{n_t}, \tag{3}$$

and the empirical $\mathcal{H}$-divergence can be computed by finding a domain classifier that separates the source domain from the target domain, i.e., by minimizing the

classification error,

$$\varepsilon(h) \leftarrow \min_{h \in \mathcal{H}} \Big[\frac{1}{n_s} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{S}}} \mathbb{1}[h(\boldsymbol{x}) = 1] \; + \; \frac{1}{n_t} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{T}}} \mathbb{1}[h(\boldsymbol{x}) = 0] \Big]. \tag{4}$$

Equation (4) shows that when the classification error is small, it means that the (domain) classifier clearly separates the two domains, and hence the $\mathcal{H}$-divergence is large, and vice-versa. It can be shown that an approximation to the empirical $\mathcal{H}$-divergence between the source and target domains, called the Proxy $\mathcal{A}$-distance (see [66, 67]), is given by

$$\hat{d}_{\mathcal{A}}(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}) \; = \; 2\,(1 \; - \; 2\varepsilon(h)), \tag{5}$$

i.e., in order to get an approximation to the $\mathcal{H}$-divergence between the source and target domains, all that is needed is to compute the classification error and the Proxy $\mathcal{A}$-distance, $\hat{d}_{\mathcal{A}}(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}})$.

3.2 Domain Adaptation Adversarial Learning

The theory suggests to have a good representation for cross-domain transferability, i.e., the algorithm must not learn how to identify the input's original domain. To this end, we want the model to learn discriminative and domain-invariant features at the same time. We can achieve this by optimizing the features and two classifiers that will operate on these features: (i) a label classifier that will predict the classification labels during training and testing, and (ii) a domain discriminator that will discriminate between source and target data during training. We optimize the parameters of the classifier and the discriminator to minimize the training error for both. At the same time, we optimize the deep feature mapping parameters to minimize the loss of the label classifier and to maximize the loss of the domain discriminator. The loss of the domain discriminator thus works adversarially, encouraging domain-invariant features to appear during the optimization. In practice, this is realized by coupling the feature extractor G_f with a label classifier G_y and a domain discriminator G_d in a three-branch architecture optimized under a joint-objective that combines the label-prediction loss $\mathcal{L}_y$ and the domain loss $\mathcal{L}_d$. During training, the gradient of $\mathcal{L}_d$ with respect to the parameters of G_f is multiplied by a negative constant via a gradient reversal layer, so that G_f is updated to increase the domain classification error while still decreasing $\mathcal{L}_y$, thereby driving the empirical discrepancy between $\mathcal{S}(G_f)$ and $\mathcal{T}(G_f)$ toward zero. We next formalize this construction by introducing the feature-space representations $\mathcal{S}(G_f)$ and $\mathcal{T}(G_f)$ and the corresponding adversarial loss. To this end, our approach is to design a feature extractor neural network G_f. Let us denote the source and target feature representations as

$$\begin{cases} \mathcal{S}(G_f) = \{G_f(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{\mathcal{S}}\}, & (6) \\ \mathcal{T}(G_f) = \{G_f(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{\mathcal{T}}\}, & (7) \end{cases}$$

for which the $\mathcal{H}$-divergence of a symmetric hypothesis class $\mathcal{H}$ between samples $\mathcal{S}(G_f)$ and $\mathcal{T}(G_f)$ is given by

$$\hat{d}_{\mathcal{H}}(\mathcal{S}(G_f), \mathcal{T}(G_f)) = 2\left(1 - \min_{h \in \mathcal{H}} \left| \frac{1}{n_s} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{S}}} [h(G_f(\boldsymbol{x})) = 0] + \frac{1}{n_t} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{T}}} [h(G_f(\boldsymbol{x})) = 1] \right| \right) \quad (8)$$

Hence, in domain adversarial feature adaptation, the neural network aims to build an internal representation that contains no discriminative information about the origin of the input (source or target), while preserving a low risk on the source (labeled) examples. Therefore, the goal of the domain adversarial neural network model is to train a (composed) deep neural network $\varphi \colon \boldsymbol{\mathcal{X}} \to \mathcal{Y}$ that can predict labels for $\boldsymbol{\mathcal{D}}_t$ by employing the information obtained from $\boldsymbol{\mathcal{D}}_s$. In general, the approximating function $\varphi(\cdot)$ can be decomposed into two functions, i.e., $\varphi = G_y \circ G_f$, that includes a feature extractor $G_f \colon \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{F}}$ that will map samples from input space $\boldsymbol{\mathcal{X}}$ to feature space $\boldsymbol{\mathcal{F}}$, and a small classifier on top that predicts the target labels, i.e., $G_y \colon \boldsymbol{\mathcal{F}} = G_f(\boldsymbol{\mathcal{X}}) \to \mathcal{Y}$ to predict from feature space into label space. So learning is performed via empirical risk minimization, and the architecture is trained with a standard classification objective to *minimize*:

$$\mathcal{L}_y(\theta_f, \theta_y) = \frac{1}{n_s} \sum_{(\boldsymbol{x}, y) \in \boldsymbol{\mathcal{S}}} \ell_y(G_y(G_f(\boldsymbol{x})), y), \quad (9)$$

with respect to any proper loss function $\ell_y(\cdot, \cdot)$. We expect the function G_y to generalize well to the target data when the distributions of source and target features generated by G_f are well aligned, i.e., there is less mismatch between the two distributions.

The cascade of the feature extractor G_f and the label classifier G_y acts as a normal fully-connected feed-forward neural network that is trained and used to classify an input sample $\boldsymbol{x}$ into one of the possible labels of the label space $\mathcal{Y}$. A `softmax` activation is used as the last layer of the label classifier G_y , which models the probability $P(y \mid \boldsymbol{x})$, $\forall\, y \in \mathcal{Y}$ of a given input $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$. Additionally, domain adversarial training introduces a domain prediction branch, which is another (binary) classifier G_d on top of the feature extractor G_f and whose goal is to approximate the domain discrepancy to distinguish whether the features come from the source or the target domain. The domain adaptation is achieved by training the feature extractor G_f in adversary with respect to the domain discriminator G_d, which tries to distinguish the two domains. This leads to the following training objective to *maximize*:

$$\mathcal{L}_d(\theta_f, \theta_d) = \frac{1}{n_s} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{S}}} \ell_d(G_d(G_f(\boldsymbol{x})), s) + \frac{1}{n_t} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{T}}} \ell_d(G_d(G_f(\boldsymbol{x})), t). \quad (10)$$

The domain discriminator G_d works as a logistic regression model that predicts the probability that an input sample $\boldsymbol{x}$ comes from the source ($s = 0$ if $\boldsymbol{x} \in \boldsymbol{\mathcal{S}}$) or the target distribution ($t = 1$ if $\boldsymbol{x} \in \boldsymbol{\mathcal{T}}$), where s (and t) denotes a binary variable that

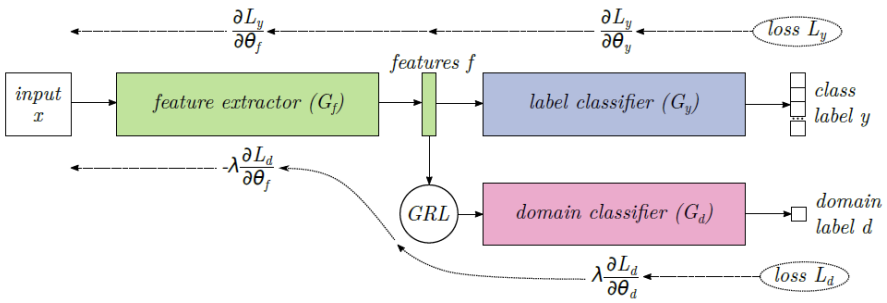

Fig. 2: The proposed architecture for DANN consists of a feature extractor and a label classifier. To enable domain adaptation, a domain discriminator is added and connected to the feature extractor by a gradient reversal layer (GRL) that multiplies the gradient by a negative constant during back-propagation.

indicates the domain of the sample. Therefore,

$$\ell_d(G_d(G_f(\boldsymbol{x})), d) = \begin{cases} -\log(1 - G_d(G_f(\boldsymbol{x}))) & \text{if } \boldsymbol{x} \in \boldsymbol{\mathcal{S}}, \\ -\log(G_d(G_f(\boldsymbol{x}))) & \text{if } \boldsymbol{x} \in \boldsymbol{\mathcal{T}}. \end{cases} \tag{11}$$

The final objective can thus be written as:

$$\begin{cases} \mathcal{L}(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\theta_f, \theta_y) + \lambda_d \mathcal{L}_d(\theta_f, \theta_d), & (12) \\ \qquad\qquad \Downarrow \\ \theta_f^*, \theta_y^* = \arg\min \mathcal{L}(\theta_f, \theta_y, \theta_d), & (13) \\ \theta_d^* = \arg\max \mathcal{L}(\theta_f, \theta_y, \theta_d). & (14) \end{cases}$$

Domain adversarial adaptation aims to learn domain invariant representations for unsupervised domain adaptation by adversarially training the feature extractor G_f and domain classifier (discriminator) G_d. The adversarial domain adaptation can be achieved by optimizing the classification loss $\mathcal{L}_y(\theta_f, \theta_y)$ and domain adversarial loss $\mathcal{L}_d(\theta_f, \theta_d)$. The model utilizes a Gradient Reversal Layer (GRL) to confuse the features generated from the feature extractor G_f, forcing it to produce features that are generic to both datasets, as illustrated in Figure 2. The gradient reversal layer is introduced to evaluate both gradients in one standard back-propagation step. It is added between the feature extractor G_f and the domain discriminator G_d, as depicted in Figure 2. The idea is that the output of G_f is normally propagated to G_d, however during back-propagation, its gradient is multiplied by -1.

We use this operation to force G_f to learn generic features that do not allow discriminating the domains.

Augmented with the gradient reversal layer, the final model is trained by minimizing Equation (12), i.e., the sum of losses $\mathcal{L}_y + \lambda_d \mathcal{L}_d$, which corresponds to the

optimization problem in Equations (13)–(14). Therefore,

$$\frac{\partial \ell_d(G_d(G_f(\boldsymbol{x};\theta_f);\theta_d),d)}{\partial \theta_f} = \frac{\partial \ell_d(G_d(\mathcal{R}(G_f(\boldsymbol{x};\theta_f));\theta_d),d)}{\partial \theta_f}, \tag{15}$$

$$= \frac{\partial \ell_d(\theta_f,\theta_d)}{\partial \mathcal{R}(G_f(\boldsymbol{x};\theta_f))} \frac{\partial \mathcal{R}(G_f(\boldsymbol{x};\theta_f))}{\partial G_f(\boldsymbol{x};\theta_f)} \frac{\partial G_f(\boldsymbol{x};\theta_f)}{\partial \theta_f}, \tag{16}$$

$$= \frac{\partial \ell_d(\theta_f,\theta_d)}{\partial G_f(\boldsymbol{x};\theta_f)} (-\boldsymbol{I}) \frac{\partial G_f(\boldsymbol{x};\theta_f)}{\partial \theta_f}, \tag{17}$$

$$= -\frac{\partial \ell_d(\theta_f,\theta_d)}{\partial G_f(\boldsymbol{x};\theta_f)} \frac{\partial G_f(\boldsymbol{x};\theta_f)}{\partial \theta_f}. \tag{18}$$

In other words, for the update of θ_d, the gradients of $\mathcal{L}_d$ with respect to activations are computed normally (minimization), but they are then propagated with a minus sign in the feature extraction part of the network (maximization). This is implemented by a function called `detach()`.

3.3 Wasserstein Domain Adaptation

Recent advances in adversarial training, e.g. [68], explain that by minimizing the error under adversarial perturbations using a Wasserstein distance to the source can likely bound the error on the target domain [69]. Wasserstein distance defines a notion of closeness between distributions. And in recent years optimal transport has become popular in machine learning and image processing problems. For instance, it is used to transform an image into another by finding the optimal way to change a histogram distribution to another [70]. Wasserstein distance is a measure to find the optimal transport of one point to match the target point in a different distribution, and has been used for an extensive variety of applications, including Bayesian inversion, image retrieval, and clustering. One way of aligning domain distributions in feature space is to add a loss function that measures the mismatch between source and target distributions into a deep learning process. In the context of this work, the Wasserstein distance provides a principled way to quantify the cost of transporting feature representations of samples from the source domain to those of the target domain under the mapping induced by G_f. Minimizing such a transport cost as an auxiliary regularization term encourages the learned feature distributions to become closer in a geometric sense, complementing the adversarial objective in promoting domain invariance. However, directly computing the Wasserstein distance in high-dimensional feature space is computationally demanding and statistically fragile, which motivates the sliced Wasserstein formulation introduced in the following subsection.

3.4 Sliced Wasserstein

Take $\text{Prob}(\mathbb{R}^L)$ to be the space of probability measures over Euclidean space $\mathbb{R}^L$. Then, the *2-Wasserstein* distance between distributions μ, $\nu \in \text{Prob}(\mathbb{R}^L)$ is defined as

$$\mathcal{W}_2^2(\mu,\nu) := \inf_{\gamma \in \Gamma(\mu,\nu)} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma} \left[\|\boldsymbol{x}-\boldsymbol{y}\|^2 \right] \tag{19}$$

where $\Gamma(\cdot, \cdot)$ denotes the space of *measure couplings*.

The Wasserstein distance is hard to compute in a high dimensional space, and does not generalize well, in the sense that the population distance, i.e., the expectation, cannot be approximated by an empirical distance when there are only a small set of samples. Therefore, projecting onto a low-dimensional subspace, mitigates the artificial effect in high dimensions, and the distance of the projected samples reflects the true distance.

Let $\mathbb{S}^{L-1}$ denote the unit sphere in $\mathbb{R}^L$. Then, the *sliced Wasserstein* distance between μ, ν equals

$$\mathrm{SW}^2(\mu, \nu) := \mathop{\mathbb{E}}_{v \sim \mathbb{S}^{L-1}} [\mathcal{W}_2^2(\mathrm{proj}_v\, \mu, \mathrm{proj}_v\, \nu)], \tag{20}$$

where $\mathrm{proj}_v\, \mu$ denotes the projection of the distribution μ onto the line through the origin, in the Radon sense. In words, $\mathrm{SW}^2(\mu, \nu)$ is the expected transport distance between the projections of μ and ν onto arbitrary directions through the origin on the unit sphere $\mathbb{S}^{L-1}$. Koulouri et al. have shown that the sliced Wasserstein distance satisfies the properties of non-negativity, symmetry, and sub-additivity, and hence it is a true metric [71].

In our setting, the measures μ and ν correspond to the empirical label-prediction distributions induced by the source and target batches, i.e., the pushforwards of $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ through the composition $G_y \circ G_f$. By minimizing the sliced Wasserstein distance between these distributions, the model is encouraged to align not only feature representations but also class-discriminative structure in the label space, thereby promoting class-consistent transfer under cross-event shifts. In practice, we approximate the expectation over directions $\boldsymbol{v} \sim \mathbb{S}^{L-1}$ with a finite number m of random projections, yielding the Monte Carlo estimator in Equation (21), which defines a differentiable loss that can be optimized jointly with $\mathcal{L}_y$ and $\mathcal{L}_d$ in Equation (24).

One-dimensional transport distances between empirical distributions can be computed in closed-form by sorting, see [72, Theorem 2.18]. We, therefore, approximate the sliced Wasserstein distance by summing over the projections along random directions suggesting an extremely straightforward and unbiased estimator for SW^2:

$$\mathrm{SW}^2(\mu, \nu) \approx \frac{1}{m} \sum_{k=1}^{m} \mathcal{W}_2^2(\mathrm{proj}_{v_k}\, \hat{\mu}, \mathrm{proj}_{v_k}\, \hat{\nu}), \tag{21}$$

where $v_1, \ldots, v_m \sim \mathbb{S}^{L-1}$ are drawn i.i.d., i.e., random unit vectors of size L, m is the number of (random) projections, and $\hat{\mu}, \hat{\nu}$ are the corresponding empirical distributions. Let $\pi_{\hat{\mu}}$ and $\pi_{\hat{\nu}}$ be permutations that sort the projected sample sets $\mathrm{proj}_{v_k}\, \hat{\mu}$ and $\mathrm{proj}_{v_k}\, \hat{\nu}$ respectively, i.e., $(\mathrm{proj}_{v_k}\, \hat{\mu})_{\pi_{\hat{\mu}}^{(1)}} \leq (\mathrm{proj}_{v_k}\, \hat{\mu})_{\pi_{\hat{\mu}}^{(2)}} \leq \cdots \leq (\mathrm{proj}_{v_k}\, \hat{\mu})_{\pi_{\hat{\mu}}^{(|\hat{\mu}|)}}$. Therefore,

$$\mathcal{W}_2^2(\mathrm{proj}_{v_k}\, \hat{\mu}, \mathrm{proj}_{v_k}\, \hat{\nu}) = \frac{1}{|\hat{\mu}|} \sum_{i=1}^{|\hat{\nu}|} \left\| (\mathrm{proj}_{v_k} \hat{\mu})_{\pi_{\hat{\mu}}^{(i)}} - (\mathrm{proj}_{v_k} \hat{\nu})_{\pi_{\hat{\nu}}^{(i)}} \right\|^2. \tag{22}$$

In the case of the proposed architecture, we have

$$\mathcal{L}_{\mathrm{SW}}(\theta_f, \theta_y) = \frac{1}{m n_s} \sum_{k=1}^{m} \sum_{i=1}^{n_t} \left\| \left[\mathrm{proj}_{v_k} G_y(G_f(\boldsymbol{x}_s)) \right]_{\pi_{\hat{\mu}}^{(i)}} - \left[\mathrm{proj}_{v_k} G_y(G_f(\boldsymbol{x}_t)) \right]_{\pi_{\hat{\nu}}^{(i)}} \right\|^2 \quad (23)$$

where $G_y(G_f(\boldsymbol{x}_s; \theta_f); \theta_y)$ and $G_y(G_f(\boldsymbol{x}_t; \theta_f); \theta_y)$ are the label predictor's `softmax` output of source and target domains; and $\mathrm{proj}_{v_k} G_y(G_f(\boldsymbol{x}_s))$ is the one-dimensional projection of $G_y(G_f(\boldsymbol{x}_s))$ onto the random unit direction $v_k \sim \mathbb{S}^{L-1}$, see Algorithm 1.

The final objective can thus be written as:

$$\mathcal{L}(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\theta_f, \theta_y) + \lambda_d \mathcal{L}_d(\theta_f, \theta_d) + \lambda_s \mathcal{L}_{\mathrm{SW}}(\theta_f, \theta_y). \quad (24)$$

Hence, by applying standard (or stochastic) gradient descent, the domain adversarial neural networks objective leads to the following gradient update rules (25)–(27)

$$\begin{cases} \theta_f \leftarrow \theta_f - \mu \left(\dfrac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda_d \dfrac{\partial \mathcal{L}_d}{\partial \theta_f} + \lambda_s \dfrac{\partial \mathcal{L}_{\mathrm{SW}}}{\partial \theta_f} \right), & (25) \\ \theta_y \leftarrow \theta_y - \mu \left(\dfrac{\partial \mathcal{L}_y}{\partial \theta_y} + \lambda_s \dfrac{\partial \mathcal{L}_{\mathrm{SW}}}{\partial \theta_y} \right), & (26) \\ \theta_d \leftarrow \theta_d + \mu \dfrac{-\lambda_d \, \partial \mathcal{L}_d}{\partial \theta_d}. & (27) \end{cases}$$

For neural networks, the gradients of the loss with respect to the parameters are obtained with the back-propagation algorithm. The current system of equations is very similar to the standard back-propagation scheme, except for the opposite sign in the derivative of $\mathcal{L}_d$ with respect to θ_d and θ_f. Therefore, running stochastic gradient descent in the resulting model implements the updates of Equations (25)–(27) and converges to a saddle point of Equation (24).

Implementation details. In Algorithm 1, $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ represent mini-batches of source and target samples, each of size n_s and n_t. The algorithm assumes $n_s = n_t$; when the batch sizes differ, the larger batch is randomly subsampled to ensure one-to-one correspondence of samples during projection. Each random direction $\boldsymbol{v}_k$ is drawn uniformly from the unit sphere $\mathbb{S}^{L-1} \subset \mathbb{R}^L$, and the projected values $\tilde{\boldsymbol{a}}_k = \boldsymbol{A}\boldsymbol{v}_k$ and $\tilde{\boldsymbol{b}}_k = \boldsymbol{B}\boldsymbol{v}_k$ (of size n_s and n_t) are sorted to approximate the one-dimensional optimal transport distance. For clarity, the number of projections m is treated as a fixed hyperparameter (typically 64) rather than a stochastic variable. The sliced Wasserstein loss is normalized first by the number of samples and then by the number of projections, in agreement with Equation (23). These explicit specifications remove ambiguity about dimensions, sampling, and normalization, ensuring that Algorithm 1 can be implemented reproducibly.

In summary, the proposed methodology integrates adversarial and optimal-transport-based domain adaptation within a unified optimization framework. All loss components, parameter update rules, and projection procedures are now presented explicitly to ensure reproducibility. The accompanying Algorithm 1 provides a step-by-step computational overview, while the following section (Section 4.2) complements

Algorithm 1 Computation of the sliced Wasserstein loss $\mathcal{L}_{\text{SW}}$. The algorithm estimates the discrepancy between source and target feature distributions by projecting both into multiple random one-dimensional subspaces, sorting the projected samples, and averaging their pairwise distances. All dimensions, sampling steps, and normalization factors are specified explicitly to ensure reproducibility and correspondence with Equation (23).

$\boldsymbol{a}_i = G_y(G_f(\boldsymbol{x}_s^{(i)}; \theta_f); \theta_y) \in \mathbb{R}^L$ and $\boldsymbol{b}_i = G_y(G_f(\boldsymbol{x}_t^{(i)}; \theta_f); \theta_y) \in \mathbb{R}^L$
$\boldsymbol{A} = G_y(G_f(\boldsymbol{X}_s)) \in \mathbb{R}^{n_s \times L}$ ▷ $\boldsymbol{a}_i^T = \boldsymbol{A}(i\text{th-row})$, $\boldsymbol{X}_s$: batch of n_s source samples
$\boldsymbol{B} = G_y(G_f(\boldsymbol{X}_t)) \in \mathbb{R}^{n_t \times L}$ ▷ $\boldsymbol{b}_i^T = \boldsymbol{B}(i\text{th-row})$, $\boldsymbol{X}_t$: batch of n_t target samples
procedure SLICED-WASSERSTEIN-LOSS($\boldsymbol{A}, \boldsymbol{B}; p$) ▷ $n_s = n_t$
    $m \leftarrow$ GEOMETRICRANDOM(p) ▷ m is the number of projections; $p \in [0, 1]$
    Initial Loss: $\mathcal{L}_{\text{SW}}(\theta_f, \theta_y) \leftarrow 0$
    for $k \in \{1, \ldots, m\}$ **do**
        $\boldsymbol{v}_k \leftarrow$ UNIFORMRANDOM($\mathbb{S}^{L-1}$) ▷ $\mathbb{S}^{L-1}$ denotes the unit sphere in $\mathbb{R}^L$
        $\tilde{\boldsymbol{a}}_k \leftarrow \boldsymbol{A}\boldsymbol{v}_k$ ▷ projection of $\boldsymbol{A}$ onto $\boldsymbol{v}_k$
        $\tilde{\boldsymbol{b}}_k \leftarrow \boldsymbol{B}\boldsymbol{v}_k$ ▷ projection of $\boldsymbol{B}$ onto $\boldsymbol{v}_k$
        $\tilde{\boldsymbol{a}}_k^{\pi} \leftarrow \text{sorted}\, \tilde{\boldsymbol{a}}_k$ and $\tilde{\boldsymbol{b}}_k^{\pi} \leftarrow \text{sorted}\, \tilde{\boldsymbol{b}}_k$
        $\mathcal{L}_{\text{SW}} \leftarrow \mathcal{L}_{\text{SW}} + \frac{1}{n_s} \|\tilde{\boldsymbol{a}}_k^{\pi} - \tilde{\boldsymbol{b}}_k^{\pi}\|^2$
    end for
return $\mathcal{L}_{\text{SW}} \leftarrow \frac{\mathcal{L}_{\text{SW}}}{m}$
end procedure

this by specifying all implementation details, hyperparameters, and hardware settings. Together, these additions make the framework fully transparent and readily replicable for future research.

4 Experiments

This section introduces the dataset and presents our experimental setup and results. In addition, it provides insights based on further investigation of the feature spaces and domain distribution discrepancy.

4.1 Dataset

The experiments were performed on a specialized dataset called Damage Assessment Dataset (DAD) [36], which contains images collected from social media during four different disaster events (i.e., domains): **N**epal Earthquake, **E**cuador Earthquake, Hurricane **M**atthew, and Typhoon **R**uby. The dataset was originally labeled with three damage severity classes: no damage, mild damage, and severe damage. Later, Li et al. [13] leveraged this dataset to perform domain adaptation experiments in the disaster domain. To simplify the task, they suggested merging severe and mild damage categories and focused only on differentiating between damage and no damage images during domain adaptation. This approach addresses the challenge of information overload by filtering the most relevant content, enabling emergency response experts to

Table 1: Data distribution

Class	Ecuador	Matthew	Nepal	Ruby
No Damage	844	127	7919	400
Damage	933	206	11,183	433
Total	1,724	333	19,102	833

Fig. 3: Sample images from the Damage Assessment Dataset [36]

prioritize high-impact disaster images for detailed human analysis. Furthermore, by focusing on binary classification, we ensure high accuracy and robustness in this study, which are prerequisites for scaling the approach to include finer-grained severity levels in the future. Table 1 shows the data distribution after the simplification. The resulting dataset remains to be challenging as it is still significantly imbalanced across disaster events (domains) and class distributions. For example, Nepal Earthquake has 19,102 images whereas Matthew Earthquake has only 333 images. Another factor is the high variance in class distributions across events. For instance, damage class in Nepal Earthquake is proportionally larger than it is in the other events. In order to capture the full story, we perform experiments for all possible source-target domain transfer tasks.

Figure 3 shows sample images from each disaster event for both classes. Since the images were collected in real-time from social media, no damage images typically include scenes with affected individuals, public prayers, aid and relief efforts whereas damage images usually show impacted objects and structures such as collapsed buildings, damaged roads, and submerged cars. This diverse nature of the classes makes the classification task challenging, too.

Table 2: Learning rate and weight decay settings.

Model	Learning Rate (LR)	LR Decay	LR Gamma	Weight Decay (WD)
Source-only	0.001	0.75	0.001	0.001
CDAN	0.001	0.75	0.001	0.001
DANN	0.001	0.75	0.001	0.001
AWDANN	0.001	0.75	0.001	0.001
CORAL	0.0005	—	—	0.0005
MMD	0.0005	—	—	0.0005

4.2 Experimental Setup

Experiments were conducted using six different models. The first model served as the baseline and was trained solely on the source domain data, referred to as the source-only model. A ResNet50 architecture [18] pre-trained on ImageNet was employed and fine-tuned on the source domain data. The ResNet50 network was selected instead of the VGG19 architecture used in prior work [13] due to its residual learning framework, which enables deeper architectures and alleviates the vanishing gradient problem. This design ensures better gradient flow during training, leading to higher accuracy on a variety of benchmarks. Moreover, ResNet50 is more parameter-efficient, requiring less memory and computational power while facilitating the learning of more complex features through residual connections.

The source-only model was assumed to define the lower bound of performance, as no target domain data were utilized during training. Once trained, it was directly evaluated on the target domain data. The second model employed was the Domain Adversarial Neural Network (DANN) [6], followed by additional state-of-the-art UDA methods, namely CORAL[20], MMD [21], and CDAN [22]. The final model evaluated was the Adversarial Sliced Wasserstein Domain Adaptation Network (AWDAN) [17]. The ResNet50 architecture was consistently used as the backbone across all models to ensure comparability. For AWDAN, optimal λ_d and λ_s values in Equation (24) were chosen from the range $\{0.001, 0.01, 0.1, 1\}$ for each transfer task separately.

All models were trained using the stochastic gradient descent (SGD) optimizer with hyperparameters summarized in Table 2. The optimizer hyperparameters (i.e., learning rate, learning rate decay, learning rate update, and weight decay) for each method follow the configurations recommended in the Transfer Learning Library[1] (tllib) and are kept fixed across all source-target pairs to ensure a fair and method-consistent comparison. A batch size of 32 images was used in all experiments, and models were trained for 50 epochs with 1000 iterations per epoch.

Following standard UDA protocols [5, 6, 22, 46, 51], all labeled source examples and unlabeled target examples were utilized during training. A half of each batch was populated by the samples from the source domain (with known labels) while the other half was composed of the samples from the target domain (with labels not revealed to the models during training). This strategy helped address the data imbalance by oversampling with replacement from the smaller domain (dataset). Each experiment

[1] https://github.com/thuml/Transfer-Learning-Library

was repeated with three random seeds, and the mean and standard deviation (std) of classification metrics (i.e., accuracy, precision, recall, and F1-score) were reported to characterize performance variability and robustness across runs.

For data pre-processing, all images were resized to 224×224 pixels and normalized using mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225]. No additional data augmentations (e.g., random cropping, horizontal flipping, or rotation) were applied. Class imbalance was not explicitly addressed, consistent with prior UDA studies [5, 6, 22, 46, 51]. All experiments were executed on a SLURM-managed high-performance computing (HPC) cluster equipped with NVIDIA P100 (16 GB) and V100 (16 GB) GPUs. Each training job was allocated a single GPU, 2 CPU cores, and 8 GB of RAM. To support reproducibility and independent replication, the code is publicly available at https://github.com/abalhomaid/disaster-assesment, and the datasets and trained model weights are available at https://huggingface.co/datasets/abalhomaid/disaster-damage-assessment.

4.3 Results

Table 3 reports the mean and standard deviation of classification results, computed over three runs, in terms of accuracy, precision, recall, and F1-score for all tasks, with the best results highlighted in bold. Across all metrics and transfer tasks, AWDAN attains the highest scores in nearly half of the cases (21 out of 48), with notable F1-score improvements exceeding 10 points on several tasks, including E→R (+16.8), N→R (+14.7), M→N (+12.7), M→R (+16.8), and R→N (+11.9). As shown in Table 4, domain adaptation methods yield larger relative performance gains over source-only models when transferring between different disaster types (e.g., earthquake → hurricane, hurricane → earthquake), where domain shifts are more pronounced and adaptation proves more beneficial. In contrast, the gains are smaller—though still substantial—for transfers within the same disaster type (e.g., earthquake → earthquake, hurricane → hurricane), reflecting greater similarity and feature overlap between source and target domains. Interestingly, all models exhibit an overall drop in precision with respect to source-only model and accomplish a bigger improvement in recall when transferring between the same disaster type events (see Section 5.1 for a discussion on this). Finally, Table 5 summarizes the overall averages across all tasks, showing that AWDAN consistently outperforms other models on all metrics except precision. While three repetitions limit formal significance testing, the consistent ranking across runs and tasks reinforces the robustness of our comparative findings.

There are challenging transfer tasks where adaptation does not improve model accuracy, i.e., there is no noted improvement over source-only results, e.g., see M→R in Table 3. This is potentially because Matthew and Ruby are very small datasets providing relatively limited visual context (i.e., information) about the disaster event. Therefore, even though the sampling-with-replacement training scheme matches the source and target domain dataset sizes, it falls short of providing genuine data samples with enough diversity to represent various types of impacts that can be caused by the disaster, which ultimately hampers model training and domain adaptation performance.

Table 3: Classification results (in %) in terms of accuracy (A), precision (P), recall (R) and F1-score (F1) for all tasks (mean_{std}). The best results are highlighted in bold while the second best results are underlined.

	E → M				E → N				E → R			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
Source-only	$69.7_{0.6}$	$82.6_{5.5}$	$65.2_{6.0}$	$72.6_{2.0}$	$81.2_{1.0}$	$\mathbf{90.9_{1.0}}$	$77.3_{1.1}$	$83.5_{0.9}$	$71.5_{1.9}$	$78.7_{4.7}$	$55.1_{4.4}$	$64.7_{2.8}$
CDAN	$\underline{76.9_{3.0}}$	$\mathbf{88.1_{1.6}}$	$72.3_{4.0}$	$79.4_{3.0}$	$\mathbf{85.9_{0.3}}$	$89.0_{0.3}$	$\mathbf{86.6_{0.3}}$	$\mathbf{87.8_{0.3}}$	$78.9_{1.5}$	$\mathbf{80.9_{1.7}}$	$77.6_{1.2}$	$79.2_{1.5}$
CORAL	$76.1_{0.9}$	$83.0_{0.8}$	$\mathbf{77.2_{0.9}}$	$80.0_{0.8}$	$81.4_{0.3}$	$85.3_{1.0}$	$82.5_{0.9}$	$83.9_{0.1}$	$78.7_{0.3}$	$78.9_{1.2}$	$80.6_{1.5}$	$79.7_{0.2}$
MMD	$76.6_{0.3}$	$84.3_{0.2}$	$76.4_{0.8}$	$\underline{80.1_{0.4}}$	$84.0_{0.6}$	$88.0_{0.9}$	$84.3_{0.6}$	$86.1_{0.5}$	$\mathbf{80.6_{1.0}}$	$\underline{79.4_{0.8}}$	$\underline{84.7_{1.2}}$	$\mathbf{82.0_{0.9}}$
DANN	$\underline{76.9_{0.8}}$	$\underline{87.6_{0.6}}$	$73.0_{1.3}$	$79.6_{0.8}$	$85.6_{1.0}$	$90.1_{0.8}$	$\underline{84.7_{1.2}}$	$87.3_{0.9}$	$78.4_{0.7}$	$78.9_{0.6}$	$79.7_{0.9}$	$79.3_{0.6}$
AWDAN	$\mathbf{78.3_{1.7}}$	$86.7_{2.0}$	$\underline{76.7_{0.9}}$	$\mathbf{81.4_{1.4}}$	$\underline{85.7_{0.3}}$	$\underline{90.5_{0.5}}$	$84.4_{0.1}$	$\underline{87.4_{0.3}}$	$\underline{79.2_{1.5}}$	$75.9_{2.6}$	$\mathbf{88.0_{2.2}}$	$\underline{81.5_{0.8}}$
Li et al. [13]	68.7	79.1	68.3	72.6	82.0	84.2	84.0	72.6	74.1	77.4	72.8	74.4

	N → E				N → M				N → R			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
Source-only	$79.7_{1.0}$	$81.1_{1.0}$	$84.9_{2.9}$	$82.9_{1.1}$	$77.5_{0.2}$	$84.4_{1.2}$	$78.0_{1.7}$	$81.1_{0.4}$	$68.1_{1.4}$	$64.0_{2.2}$	$75.6_{2.7}$	$69.3_{0.4}$
CDAN	$\mathbf{90.2_{0.9}}$	$88.8_{0.6}$	$\mathbf{93.7_{1.1}}$	$\mathbf{91.2_{0.8}}$	$\underline{82.7_{0.2}}$	$88.3_{0.2}$	$\underline{83.0_{0.7}}$	$\underline{85.6_{0.2}}$	$\mathbf{83.3_{0.6}}$	$81.1_{0.8}$	$\underline{88.5_{1.2}}$	$\mathbf{84.6_{0.4}}$
CORAL	$88.9_{0.3}$	$\underline{90.3_{0.6}}$	$89.1_{1.2}$	$89.7_{0.4}$	$78.7_{0.4}$	$87.0_{0.7}$	$77.2_{1.5}$	$81.8_{0.5}$	$77.2_{0.3}$	$80.4_{0.4}$	$74.4_{0.7}$	$77.3_{0.3}$
MMD	$89.4_{0.6}$	$90.2_{0.1}$	$90.2_{1.6}$	$90.2_{0.7}$	$81.9_{0.2}$	$88.5_{1.1}$	$81.4_{2.1}$	$84.7_{0.5}$	$79.7_{0.3}$	$\mathbf{82.3_{0.4}}$	$77.6_{1.0}$	$79.8_{0.4}$
DANN	$89.8_{0.2}$	$\mathbf{90.5_{1.4}}$	$\underline{90.7_{1.6}}$	$90.6_{0.2}$	$82.4_{0.5}$	$\underline{89.0_{0.3}}$	$81.6_{0.8}$	$85.1_{0.4}$	$\underline{82.4_{0.4}}$	$\underline{81.2_{0.6}}$	$86.2_{2.9}$	$83.6_{0.8}$
AWDAN	$\underline{90.1_{0.4}}$	$88.6_{0.9}$	$\mathbf{93.7_{1.1}}$	$\underline{91.1_{0.3}}$	$\mathbf{84.2_{1.1}}$	$\mathbf{89.4_{1.4}}$	$\mathbf{84.5_{0.6}}$	$\mathbf{86.8_{0.8}}$	$82.3_{2.2}$	$79.2_{1.5}$	$\mathbf{89.4_{2.9}}$	$\underline{84.0_{2.1}}$
Li et al. [13]	87.1	86.0	90.9	88.4	70.6	86.0	63.4	72.4	80.0	80.8	81.2	81.9

	M → E				M → N				M → R			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
Source-only	$\mathbf{88.4_{0.5}}$	$\mathbf{93.6_{0.3}}$	$86.0_{0.9}$	$\mathbf{89.6_{0.5}}$	$71.2_{2.1}$	$\mathbf{91.3_{1.4}}$	$59.1_{3.3}$	$71.7_{2.6}$	$70.9_{2.4}$	$\mathbf{87.0_{4.3}}$	$45.6_{2.9}$	$59.8_{3.6}$
CDAN	$79.7_{0.4}$	$78.2_{0.4}$	$86.7_{1.2}$	$82.2_{0.5}$	$\underline{79.8_{1.7}}$	$80.3_{1.5}$	$\mathbf{86.8_{1.8}}$	$\underline{83.4_{1.4}}$	$70.7_{0.4}$	$68.3_{1.0}$	$81.5_{0.7}$	$74.3_{0.2}$
CORAL	$80.7_{0.4}$	$79.7_{1.3}$	$86.2_{0.8}$	$82.9_{0.3}$	$77.6_{0.3}$	$79.5_{2.2}$	$83.4_{2.9}$	$81.3_{0.7}$	$\underline{74.4_{2.0}}$	$\underline{72.0_{3.6}}$	$83.1_{0.6}$	$\underline{77.1_{1.6}}$
MMD	$\underline{82.4_{0.3}}$	$\underline{80.7_{1.3}}$	$\underline{88.7_{1.3}}$	$\underline{84.5_{0.2}}$	$79.4_{0.9}$	$80.8_{2.2}$	$85.2_{4.4}$	$82.9_{1.3}$	$\mathbf{74.9_{1.2}}$	$71.9_{2.2}$	$\mathbf{84.6_{1.7}}$	$\mathbf{77.8_{1.1}}$
DANN	$78.4_{0.7}$	$77.8_{0.5}$	$84.1_{0.8}$	$80.8_{0.6}$	$\underline{79.8_{0.5}}$	$81.1_{0.6}$	$85.5_{0.8}$	$83.3_{0.6}$	$71.5_{0.4}$	$69.3_{0.6}$	$81.1_{0.2}$	$74.7_{0.4}$
AWDAN	$81.3_{0.4}$	$77.9_{0.6}$	$\mathbf{91.4_{1.1}}$	$84.1_{0.4}$	$\mathbf{81.3_{0.5}}$	$\underline{82.5_{1.3}}$	$\underline{86.4_{1.8}}$	$\mathbf{84.4_{0.5}}$	$73.3_{0.5}$	$70.4_{2.3}$	$\underline{84.1_{3.0}}$	$76.6_{0.1}$
Li et al. [13]	76.1	74.7	87.0	80.2	75.5	78.4	80.5	79.3	68.1	66.9	77.4	71.3

	R → E				R → M				R → N			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
Source-only	$80.6_{1.5}$	$86.6_{2.6}$	$79.1_{6.4}$	$82.5_{2.3}$	$74.1_{0.7}$	$83.5_{1.5}$	$72.5_{2.4}$	$77.6_{1.0}$	$69.9_{0.7}$	$85.2_{0.7}$	$62.1_{2.0}$	$71.9_{1.1}$
CDAN	$87.5_{0.3}$	$\underline{90.0_{1.1}}$	$86.6_{1.8}$	$88.2_{0.5}$	$\underline{76.2_{0.8}}$	$83.3_{1.0}$	$\underline{77.0_{2.9}}$	$\underline{80.0_{1.1}}$	$\underline{80.1_{0.8}}$	$\underline{87.3_{0.3}}$	$77.3_{1.7}$	$82.0_{0.9}$
CORAL	$85.5_{0.2}$	$85.7_{0.5}$	$88.0_{1.2}$	$86.8_{0.3}$	$76.1_{1.1}$	$82.4_{0.5}$	$\mathbf{78.0_{2.0}}$	$\mathbf{80.1_{1.1}}$	$78.4_{0.3}$	$82.4_{0.2}$	$\underline{80.3_{0.7}}$	$81.3_{0.4}$
MMD	$\underline{88.2_{0.4}}$	$89.7_{1.0}$	$\mathbf{88.4_{0.6}}$	$\underline{89.0_{0.3}}$	$75.5_{1.5}$	$\underline{83.8_{0.7}}$	$74.4_{1.4}$	$78.8_{0.7}$	$80.5_{0.6}$	$86.7_{0.6}$	$\underline{78.9_{1.0}}$	$\underline{82.6_{0.6}}$
DANN	$87.0_{0.7}$	$89.0_{0.5}$	$86.2_{0.8}$	$87.5_{0.6}$	$74.9_{0.5}$	$81.7_{0.6}$	$74.5_{0.8}$	$78.0_{0.6}$	$\underline{80.1_{0.4}}$	$86.5_{0.6}$	$78.7_{0.5}$	$82.4_{0.4}$
AWDAN	$\mathbf{89.1_{0.2}}$	$\mathbf{90.2_{0.9}}$	$\underline{88.2_{0.8}}$	$\mathbf{89.2_{0.4}}$	$\mathbf{76.4_{0.9}}$	$\mathbf{84.1_{1.5}}$	$75.4_{2.1}$	$79.5_{1.2}$	$\mathbf{80.8_{1.0}}$	$\mathbf{87.4_{0.9}}$	$\mathbf{80.5_{1.5}}$	$\mathbf{83.8_{1.0}}$
Li et al. [13]	83.5	84.7	85.1	84.9	75.7	81.9	76.5	79.1	77.2	80.4	77.7	79.0

More importantly, we note that domain adaptation is not necessarily a symmetric operation. For instance, a model adapted from domain M to domain E will be—in general—different from a model adapted from domain E to domain M due to the differences in training dataset sizes, class distributions, and more importantly, supervision provided by the source domain labels that guide the training process. Therefore, it is expected to observe discrepancies between each experiment (e.g., M→E) and its counterpart (e.g., E→M). These discrepancies seem to decrease as the training dataset sizes increase (e.g., Nepal and Ecuador) because having larger and visually more diverse training data helps to reduce the domain shift gap better.

In the same vein, transfer tasks that have either Nepal or Ecuador as the target dataset yield the highest performance thanks to their sizes, since having many visually

Table 4: Comparison of overall improvements achieved by each model against the baseline (source-only) model between transfer tasks of the same disaster type versus different disaster types.

	Different Disaster Types				Same Disaster Type			
	A	P	R	F1	A	P	R	F1
CDAN	5.53	1.99	10.69	8.29	4.28	-3.28	14.63	7.38
CORAL	3.75	0.19	8.26	7.06	3.73	-3.13	13.10	6.75
MMD	5.38	1.89	10.73	8.54	4.48	-2.15	13.30	7.28
DANN	4.84	1.70	9.94	8.04	3.98	-2.73	12.68	6.70
AWDAN	6.33	1.44	13.59	9.79	4.90	-2.23	14.33	7.70
Average	5.16	1.44	10.64	8.34	4.27	-2.70	13.61	7.16

Table 5: Average classification statistics across all tasks. The best results are highlighted in bold while the second best results are underlined. Values are shown as mean_{std}.

	Accuracy	Precision	Recall	F1-score
Source-only	$75.2_{6.1}$	$\mathbf{84.1}_{7.8}$	$70.0_{12.5}$	$75.6_{8.6}$
CDAN	$81.0_{5.3}$	$83.6_{6.2}$	$\underline{83.1}_{6.1}$	$\underline{83.2}_{4.6}$
CORAL	$79.5_{4.1}$	$82.2_{4.7}$	$81.7_{4.6}$	$81.8_{3.6}$
MMD	$\underline{81.1}_{4.5}$	$83.8_{5.2}$	$83.0_{5.2}$	$\underline{83.2}_{3.8}$
DANN	$80.6_{5.1}$	$\underline{83.9}_{6.3}$	$81.9_{5.4}$	$82.7_{4.4}$
AWDAN	$\mathbf{81.9}_{4.8}$	$\underline{83.9}_{6.6}$	$\mathbf{85.0}_{6.0}$	$\mathbf{84.2}_{4.1}$
Li et al. [13]	76.1	79.8	78.4	77.7

diverse data points from target domain during training helps the feature extractor G_f and domain discriminator G_d to learn better features to reduce the domain shift gap. We also compare our results with [13] which implemented a DANN model with a pre-trained VGG-19 as a feature extractor whereas for our DANN, we used a ResNet50 with layer freezing. We see that both our DANN and AWDAN results are better than the results reported in [13] across all tasks. In particular, according to Table 5, our DANN results are 9.4% better in F1-score (i.e., 75.6 vs. 82.7) and 7.2% in accuracy (i.e., 75.2 vs. 80.6) against [13]. Furthermore, our AWDAN results show an additional improvement of 1.8% in F1-score (i.e., 82.7 vs. 84.2) and 1.6% in accuracy (i.e., 80.6 vs. 81.9) over our DANN results.

5 Discussion

Our experimental results show that the proposed method yields the best performance with an average improvement of 11.4% in F1-score (i.e., 75.6 vs. 84.2) and 8.9% in

Table 6: Confusion matrices for E→M

		Predicted label					
		Source-only		DANN		AWDAN	
		No damage	Damage	No damage	Damage	No damage	Damage
True label	No damage	119	8	108	19	107	20
	Damage	105	101	52	154	46	160

accuracy (i.e., 75.2 vs. 81.9) over the source-only (lower bound) model. It also outperforms the prior art [13] by 8.4% in F1-score and 7.6% in accuracy as well as other strong baselines such as CORAL, MMD, and CDAN by 1.2-2.9% in F1-score and 1.0-3.0% in accuracy. In this section, we aim to shed light on the reasons behind the superior performance of the proposed method by analyzing confusion matrices, latent feature spaces, distribution discrepancies, and qualitative results. Since the main highlight of our study is DANN and AWDAN, we present a more focused interpretation of their comparative performances with respect to the source-only model as well as the prior art [13].

5.1 Confusion Matrices

To further explain the general trends in performance, we analyze confusion matrices of source-only, DANN, and AWDAN for one of the E→M runs, as an example, in Table 6. On the left, we can see that the source-only model classifies most of the no-damage images correctly (i.e., 119 out of 127) but struggles with damage images where it classifies a big portion of the damage images as no damage (i.e., 105 out of 206). In the middle, DANN classifies damage images better than the source-only model (154 out of 206) but also misses more of the no-damage images (i.e., 19 out of 127). On the right, AWDAN's classification performance is more balanced, i.e., it gets more damage images (i.e., 160 out of 206) and no-damage images (i.e., 107 out of 127) correctly classified as compared to DANN. Consequently, source-only model has the highest precision value (i.e., 92.7) thanks to its ability to make fewer incorrect damage predictions (i.e., FP=8). However, its recall is the lowest (i.e., 49.0) due to mistaking too many damage images as no damage (i.e., FN=105). Therefore, the source-only model achieved the lowest F1-score (i.e., 64.1). We note that this trend is prominent in source-only model performances across all tasks. On the other hand, DANN has a relatively lower precision (i.e., 89.0) due to a large increase in incorrect damage predictions (i.e., FP=19) despite achieving a significant improvement in recall (i.e., 74.8), which also leads to an overall improvement in F1-score (i.e., 81.3). Finally, AWDAN achieves the highest recall (i.e., 77.7) while also attaining a reasonable precision (i.e., 88.9), which eventually leads to the highest F1-score (82.9) thanks to its ability to find a better balance between precision and recall. We observe that this scenario holds for many other transfer tasks, as well. Thus, by inductive reasoning, we conclude that such model behavior explains the trends summarized in Table 5.

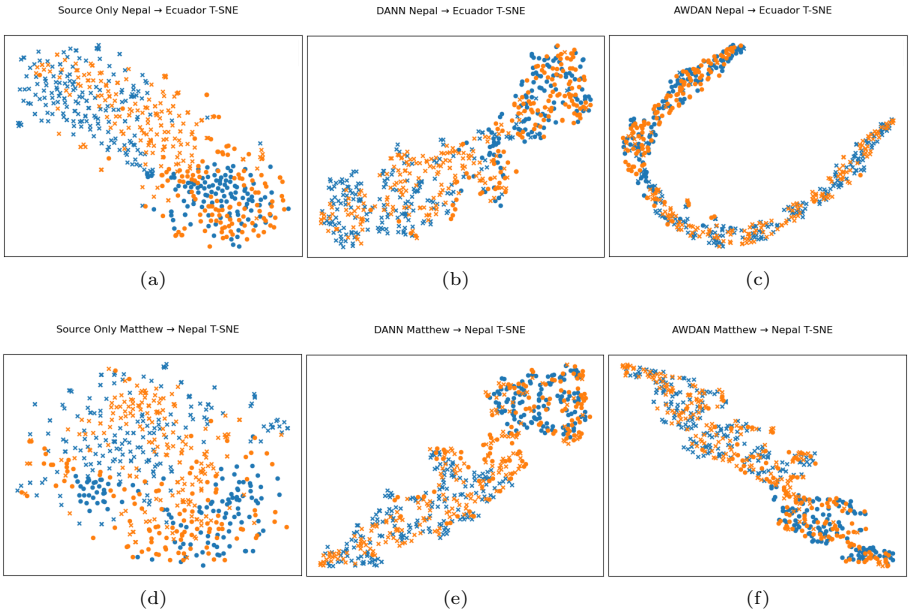

Fig. 4: Feature visualizations for (top) N→E and (bottom) M→N using t-SNE for all the models: (left) source-only, (middle) DANN, and (right) AWDAN. Blue color represents the source domain whereas orange color represents the target domain. Circles and crosses represent no-damage and damage classes, respectively.

5.2 Feature Visualization

We use the feature extractor G_f to visualize the learned feature spaces and analyze the similarities between the source and target domains for each model using t-SNE [73] in Figure 4. In these plots, we use blue and orange colors to distinguish the source and target domain data, respectively. We also discriminate between the classes using crosses ($\times$) and circles ($\bullet$) to indicate the damage and no-damage images, respectively. We compare the domain adaptation results between disasters of the same type (e.g., Nepal and Ecuador earthquakes) for N→E in Figures 4a-4c and different types (e.g., Hurricane Matthew and Nepal Earthquake) for M→N in Figures 4d-4f. When we compare the source-only plots, we notice a larger overlap between the source-only features for the same disaster type (Figure 4a) whereas we observe a limited overlap (i.e., larger domain shift) between the source-only features for different disaster types (Figure 4d). Furthermore, we see a better separation of damage and no-damage classes for the same disaster type (Figure 4a) whereas we witness more confusion of classes for the different disaster types (Figure 4d). As we go from source-only to DANN and AWDAN for the same disaster type (Figures 4b-4c), we see the source and target features are aligned more closely and the classes are clustered more tightly with clearer separation boundaries. Similarly, for different disaster types in Figures 4e-4f, we observe that both DANN and AWDAN are able to reduce the domain shift reasonably well with

Table 7: $\mathcal{A}$-distance results on all tasks

	E→M	E→N	E→R	M→E	M→N	M→R	N→E	N→M	N→R	R→E	R→M	R→N	AVG
Source-only	1.097	0.919	1.278	1.032	1.006	0.869	0.844	1.078	1.344	1.375	0.898	1.352	1.091
DANN	0.070	0.083	0.368	0.041	0.056	0.431	0.354	0.111	0.542	0.208	0.118	0.333	0.226
AWDAN	0.126	0.153	0.625	0.007	0.076	0.056	0.216	0.236	0.243	0.306	0.354	0.083	0.207

Table 8: Average $\mathcal{A}$-distances of tasks with the same target domain

	→E	→M	→N	→R
Source-only	1.084	1.024	1.092	1.164
DANN	0.201	0.100	0.157	0.447
AWDAN	0.176	0.239	0.104	0.308

respect to the source-only scenario (Figure 4d). However, they struggle with proper clustering of the classes, DANN more so than AWDAN, resulting in higher inter-class confusions than their same-disaster-type counterparts (Figures 4b-4c). Overall, we can conclude that AWDAN achieves better clustering of the classes thanks to its sliced Wasserstein loss that accounts for the alignment of the label spaces. This conclusion is in line with previous research [17].

5.3 Distribution Discrepancy

To quantitatively evaluate each model's ability to deal with distribution discrepancy (i.e., domain shift gap), we used the proxy $\mathcal{A}$-distance [66] as introduced by Equations (4) and (5) in Section 3.1. Following [6, 74, 75], we trained an SVM model to classify source and target domain features learned by each domain adaptation model. We then used the resulting SVM model's classification error (ϵ) to compute the $\mathcal{A}$-distance. The results are summarized in Table 7. Before discussing the results, it is important to reiterate that $\mathcal{A}$-distances for transfers between two domains (e.g., E→N and N→E) may not be the same because domain adaptation is not necessarily symmetric and each transfer task may result in a different model, and hence, different feature representations. With this in mind, we observe that DANN and AWDAN show great improvement as compared to the source-only model by achieving more than 80% reduction in average $\mathcal{A}$-distance. Both DANN and AWDAN achieved the lowest $\mathcal{A}$-distances for M→E at 0.041 and 0.007, respectively. DANN and AWDAN performed similarly (i.e., less than 0.1 $\mathcal{A}$-distance difference) in many of the tasks, such as E→M, E→N, M→N, and R→E. On average, AWDAN (0.207) performed slightly better than DANN (0.226) although the difference is negligible. This can be attributed to the fact that AWDAN seeks a compromise between feature alignment and label alignment across the source and target domains. Furthermore, when we break down the $\mathcal{A}$-distance results and compute average scores for the tasks with the same target

domain (e.g., M→E, N→E, and R→E) separately as shown in Table 8, we observe that AWDAN achieves lower scores (hence, better performances) when Ecuador and Nepal are the target domains. This observation supports our previous finding in Section 4.3 that domains with larger and more diverse training datasets yield better reduction in domain shift gap.

5.4 Qualitative Analysis

Here, we qualitatively analyze representative success and failure cases using class activation maps based on Smooth-GradCam++ [76], which highlight the discriminative image regions the models rely on to predict whether an image depicts damage or no damage. Figure 5 presents results for E→M: the top two rows show examples correctly classified by both DANN and AWDAN, while the bottom two rows display examples misclassified by both models across both classes. For true positives (correctly classified damage images), the activation maps reveal that both models primarily attend to flooded streets and, to a lesser extent, cloudy skies—key indicators of disaster impact. For true negatives (correctly classified no-damage images), both models focus on intact areas with no visible signs of destruction, successfully ignoring misleading cues such as cloudy skies or natural surface irregularities (e.g., rough beach textures). In contrast, for false positives (no-damage scenes misclassified as damage), the examples include an old building—where both models focus on worn-out structures—and a desert scene—where attention concentrates on rocky terrain. In both cases, the models appear to confuse naturally degraded or rugged regions with disaster ruins. For false negatives (damage scenes misclassified as no damage), the images depict floods without clear evidence of structural damage. Here, both models either fail to attend to discriminative regions (e.g., focusing on the sky or people instead of waterlogged areas) or correctly localize the affected regions (e.g., floating objects on a lake-like surface) but misinterpret their significance, leading to incorrect predictions.

Figure 6 further highlights the disagreement cases between DANN and AWDAN. The top two rows show images misclassified by DANN but correctly classified by AWDAN, while the bottom two rows show the opposite. For the top two no-damage images, AWDAN tends to focus on a single salient region, whereas DANN attends to two distinct areas in each image, which may introduce confusion and lead to incorrect predictions. In the bottom two no-damage images, DANN focuses effectively on the vehicles—key cues for identifying no-damage scenes—while disregarding cluttered regions, such as cloudy skies and humans, that AWDAN attends to more broadly, potentially reducing its confidence. Regarding the misclassified damage images (right-hand side), the top two examples show flood scenes visually similar to those in Figure 5. DANN's attention is scattered over irrelevant background objects, leading to misclassification, whereas AWDAN focuses more precisely on flooded regions and relevant foreground structures, resulting in correct predictions. Conversely, in the bottom two damage images, DANN attends to meaningful cues such as dark skies and vegetation, supporting accurate predictions, while AWDAN's attention is more diffuse and less targeted—despite partially focusing on the overturned car and wind-blown tree—ultimately producing erroneous predictions. These cases require further investigation.

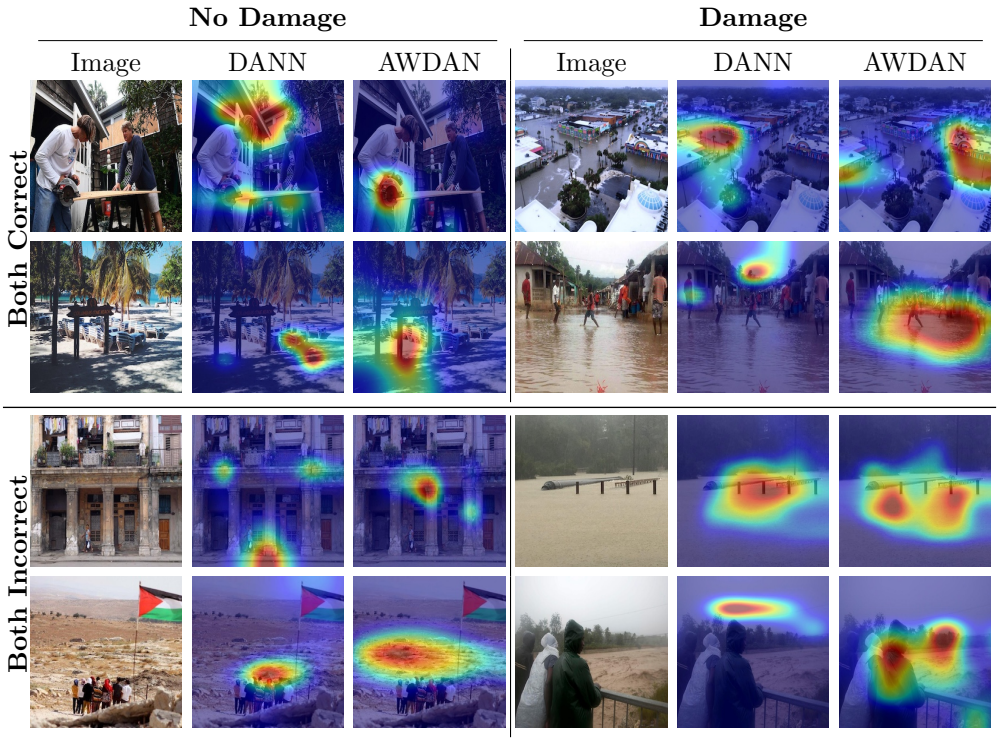

Fig. 5: Example cases for E→M using DANN and AWDAN. The top two rows show examples correctly classified by both models, while the bottom two rows present examples misclassified by both models. Each example is shown alongside the corresponding class activation maps from DANN and AWDAN.

5.5 Computational Cost Analysis

To assess the practical feasibility of different UDA methods, we conducted a comparative analysis of their computational efficiency and resource requirements. All models share the same ResNet-50 backbone, ensuring that any observed computational differences stem solely from their domain adaptation components. We report relative training time and memory overheads with respect to the source-only baseline model.

CORAL is the most lightweight method, introducing only a covariance alignment loss with negligible computational overhead (∼1.05× training time, +5% memory). MMD incurs moderate additional cost (∼1.2×, +10–20%) due to kernel-based computations but achieves stronger feature alignment. DANN adds a small domain classifier trained adversarially, modestly increasing cost (∼1.3×, +20%) while providing stable and robust adaptation performance.

AWDAN, which employs adversarial sliced-Wasserstein alignment, is more computationally demanding (∼1.6×, +30%) because of multiple random projections and sorting operations. However, it yields smoother and more stable alignment,

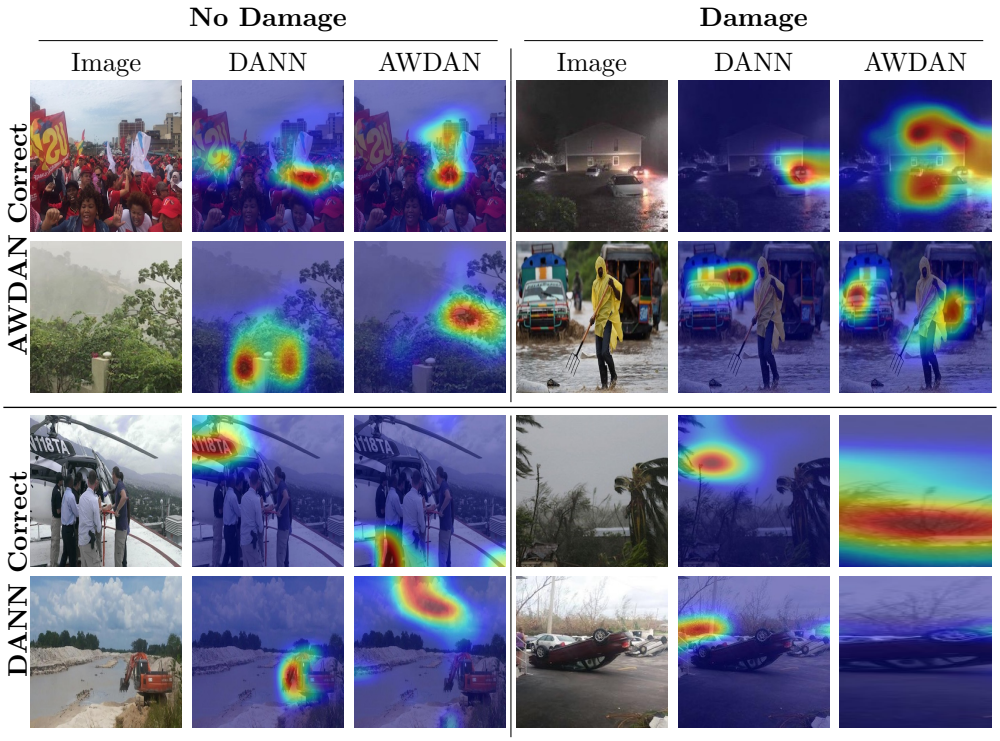

Fig. 6: Disagreement cases between DANN and AWDAN for E→M. The top two rows present examples misclassified by DANN but correctly classified by AWDAN, whereas the bottom two rows show the opposite. The left two columns illustrate *no-damage* images misclassified as *damage*, and the right two columns illustrate *damage* images misclassified as *no damage*. Each example is accompanied by the corresponding class activation maps from both DANN and AWDAN.

particularly under large domain shifts. CDAN represents the most computationally intensive approach (∼2×, +40–50%), as conditioning the discriminator on joint feature–label representations increases dimensionality and occasionally introduces training instability, albeit often leading to the highest accuracy.

Overall, MMD and DANN offer an effective balance between computational efficiency and adaptation performance, making them suitable for most experimental settings. CORAL remains an efficient and fast baseline, ideal for rapid prototyping and ablation studies. While CDAN delivers strong results, its computational burden is substantial. AWDAN, in contrast, achieves the best empirical F1-scores in our experiments with only a moderate increase in training cost, making it a strong choice when computational resources permit and domain gaps are large.

5.6 Future Directions

Even though our dataset is relatively small with respect to many other standard domain adaptation datasets [77–80], it exhibits large disparities in terms of the sizes of individual domains and in terms of the underlying distributions of classes within each domain, as illustrated in Table 1. In this study, we followed the standard practice [5, 6, 22, 46, 51] and accounted for the domain size differences by oversampling with replacement, but did not explicitly consider the class (i.e., label) imbalance issue even though AWDAN aims to achieve better alignment not only in the feature space but also in the label space, which is potentially impacted by the class imbalance. Therefore, an improved algorithm that also addresses the class imbalance problem can be designed by adding more appropriate loss functions [81, 82].

Assessing the severity of disaster damage is undoubtedly an important goal, and our current focus on binary classification (distinguishing between damage and no damage) serves as a critical foundational step toward more advanced assessments. This approach deals with the information overload problem by filtering the most relevant content, allowing emergency responders to prioritize high-impact disaster images for detailed expert analysis. Additionally, it functions effectively as a first-alert system. By prioritizing high accuracy and robustness in this binary task, we establish a reliable baseline upon which finer-grained severity classifications can be built. To this end, future extensions of this work could adapt the proposed framework to handle multi-class labels, enabling severity-level assessments. This progression can be supported by leveraging more recent and larger datasets, such as MEDIC [83] and Incidents1M [35], which offer rich resources for studying unsupervised domain adaptation (UDA) in the context of rapid disaster damage assessment. Furthermore, advanced algorithms, such as incremental training techniques with AWDAN [84], present promising avenues for future research to refine and expand the capabilities of the current approach.

5.7 Potential Biases and Weaknesses

Social media data, including Twitter, can exhibit a variety of biases—geographic, demographic, temporal, and device-related—that may affect the generalizability of models trained on such data. For example, posts are often concentrated in urban areas or regions with higher smartphone penetration, while rural or low-connectivity areas are underrepresented. Demographics may be skewed toward younger populations or certain socioeconomic groups, and particular devices or camera types can dominate the dataset, influencing both the visual and textual content available for analysis. Beyond these structural biases, social media recommender systems tend to amplify extreme or attention-grabbing content, favoring highly positive or highly negative posts. This creates a dataset skew toward sensationalized imagery, which may overrepresent dramatic events while underrepresenting subtle but operationally significant disaster impacts. Models trained on such data may therefore misclassify minor damage, exaggerate trends, or fail to generalize across disaster contexts.

Despite these limitations, social media remains one of the fastest and most accessible sources of situational information during time-critical events. However, much of this data is noisy, redundant, or irrelevant, making it impractical to rely on a single model

for operational assessment. More practical approaches, as shown in prior work [37], use pre-filtering models or auxiliary classifiers to remove duplicates and irrelevant posts before conducting detailed damage assessment. Additionally, our study does not evaluate the authenticity or veracity of the images, which is a separate and critical challenge requiring dedicated verification pipelines, either automatic or manual.

Automated disaster assessment systems also raise important privacy concerns. Social media images may contain personally identifiable information (e.g., faces, license plates, residential areas) or reveal sensitive locations such as critical infrastructure or shelters. Without appropriate safeguards, automated systems could inadvertently expose individuals or communities to privacy risks or misuse. Ethical deployment therefore requires anonymization, aggregation, or other privacy-preserving strategies to minimize potential harm.

Beyond privacy, automated disaster assessment has significant operational and societal implications. While such systems can accelerate situational awareness and improve resource allocation, erroneous predictions—false negatives (missed damage) or false positives (overstated damage)—can misguide emergency response, delay aid, or create public panic. Biases in the underlying data may amplify inequities, disproportionately affecting certain regions or populations. Consequently, these tools must be deployed with human oversight, integrated with validated complementary data sources (satellite, UAV, and ground reports), and continuously monitored to ensure fairness, reliability, and accountability.

In summary, while social media–based automated disaster assessment holds great promise for rapid response, careful attention to bias, privacy, ethical use, and operational impact is essential to avoid unintended harms. Integrating multiple data sources, robust pre-filtering, and human oversight are critical to ensuring that these systems are both reliable and actionable, and ultimately serve the needs of affected communities.

6 Conclusion

Disaster imagery varies widely across events, causing distribution shifts that hinder generalization from historical data to emergent crises. We evaluated modern unsupervised domain adaptation (UDA) methods -particularly DANN and AWDAN- on social-media images from four major disasters (2015 Nepal Earthquake, 2016 Ecuador Earthquake, 2014 Typhoon Ruby, 2016 Hurricane Matthew) and found that cross-event transfer is challenging without adaptation. Integrating optimal-transport–based alignment (AWDAN) with adversarial training and a deeper backbone consistently mitigated these shifts, yielding new state-of-the-art performance for rapid damage assessment and surpassing prior benchmarks (i.e., [13]) by 8.4% in F1 and 7.6%. Remaining limitations include disparities in dataset sizes and class distributions across events. Future work should explore stronger class/data balancing and incremental training, as well as extensions to fine-grained severity estimation via multi-class or regression formulations. Overall, the results indicate that carefully instantiated UDA offers a scalable and adaptable path for robust cross-event disaster assessment.

Acknowledgements. Acknowledgements will be added upon acceptance.

Declarations

Funding Will be added upon acceptance.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Data availability The experimental data that support the findings of this study are available online at: https://crisisnlp.qcri.org/ (Resource #9).

Materials availability Not applicable.

Code availability The source code used in the experiments will be shared publicly upon acceptance.

Author contribution Will be included upon acceptance.

References

[1] Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., He, Q.: Deep subdomain adaptation network for image classification. IEEE transactions on neural networks and learning systems **32**(4), 1713–1722 (2020)

[2] Chen, C., Fu, Z., Chen, Z., Jin, S., Cheng, Z., Jin, X., Hua, X.-S.: Homm: Higher-order moment matching for unsupervised domain adaptation. In: AAAI Conference on Artificial Intelligence, pp. 3422–3429 (2020)

[3] Liu, Y., Tian, X., Li, Y., Xiong, Z., Wu, F.: Compact feature learning for multi-domain image classification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7193–7201 (2019)

[4] Deng, W., Su, Z., Qiu, Q., Zhao, L., Kuang, G., Pietikäinen, M., Xiao, H., Liu, L.: Deep ladder reconstruction-classification network for unsupervised domain adaptation. Pattern Recognition Letters **152**, 398–405 (2021)

[5] Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189 (2015). PMLR

[6] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The journal of machine learning research **17**(1), 2096–2030 (2016)

[7] Wang, Y., Zhang, Y., Xu, L., Qi, S., Yao, Y., Qian, W., Greenwald, S.E., Qi, L.: Tsp-udanet: two-stage progressive unsupervised domain adaptation network for automated cross-modality cardiac segmentation. Neural Computing and Applications **35**(30), 22189–22207 (2023)

[8] Zhang, J., Xie, Y., Sun, D., Huang, R., Wang, T., Lei, B., Chen, K.: Multi-national ct image-label pairs synthesis for covid-19 diagnosis via few-shot generative adversarial networks adaptation. Neural Computing and Applications **36**(9), 5007–5019 (2024)

[9] Wu, X., Fan, X., Luo, P., Choudhury, S.D., Tjahjadi, T., Hu, C.: From laboratory to field: Unsupervised domain adaptation for plant disease recognition in the wild. Plant Phenomics **5**, 0038 (2023)

[10] Wang, K., Zhang, L., Xia, Q., Pu, L., Chen, J.: Cross-domain learning using optimized pseudo labels: Toward adaptive car detection in different weather conditions and urban cities. Neural Computing and Applications, 1–11 (2022)

[11] Ghosh, S., Maji, S., Desarkar, M.S.: Unsupervised domain adaptation with global and local graph neural networks under limited supervision and its application to disaster response. IEEE Transactions on Computational Social Systems **10**(2), 551–562 (2022)

[12] Li, Y., Lin, C., Li, H., Hu, W., Dong, H., Liu, Y.: Unsupervised domain adaptation with self-attention for post-disaster building damage detection. Neurocomputing **415**, 27–39 (2020)

[13] Li, X., Caragea, D., Caragea, C., Imran, M., Ofli, F.: Identifying disaster damage images using a domain adaptation approach. In: International Conference on Information Systems for Crisis Response and Management (ISCRAM), pp. 1–13 (2019)

[14] Khattar, A., Quadri, S.: Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter. Multimedia Tools and Applications **81**(21), 30437–30464 (2022)

[15] Khattar, A., Quadri, S.: Multi-source domain adaptation of social media data for disaster management. Multimedia tools and applications **82**(6), 9083–9111 (2023)

[16] Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)

[17] Zhang, Y., Wang, N., Cai, S.: Adversarial sliced wasserstein domain adaptation networks. Image and Vision Computing **102**, 103974 (2020)

[18] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[19] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[20] Sun, B., Feng, J., Saenko, K.: In: Csurka, G. (ed.) Correlation Alignment for Unsupervised Domain Adaptation, pp. 153–171. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58347-1_8 . https://doi.org/10.1007/978-3-319-58347-1_8

[21] Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409 (2018)

[22] Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. Advances in neural information processing systems **31** (2018)

[23] Kolouri, S., Pope, P.E., Martin, C.E., Rohde, G.K.: Sliced wasserstein auto-encoders. In: International Conference on Learning Representations, pp. 1–19 (2018)

[24] Alnaimi, F.: Deep adversarial sliced wasserstein domain adaptation neural networks and applications to disaster datasets. Master's thesis, Hamad Bin Khalifa University, Qatar (2022)

[25] Palen, L., Anderson, K.M.: Crisis informatics—new data for extraordinary times. Science **353**(6296), 224–225 (2016)

[26] Poblete, B., Guzmán, J., Maldonado, J., Tobar, F.: Robust detection of extreme events using twitter: Worldwide earthquake monitoring. IEEE Transactions on Multimedia **20**(10), 2551–2561 (2018)

[27] Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: A survey. ACM Computing Surveys (CSUR) **47**(4), 1–38 (2015)

[28] Zahra, K., Imran, M., Ostermann, F.O.: Automatic identification of eyewitness messages on twitter during disasters. Information processing & management **57**(1), 102107 (2020)

[29] Lin, X., Spence, P.R., Sellnow, T.L., Lachlan, K.A.: Crisis communication, learning and responding: Best practices in social media. Computers in Human Behavior **65**, 601–605 (2016)

[30] Kruspe, A., Kersten, J., Klan, F.: Detection of actionable tweets in crisis events. Natural Hazards and Earth System Sciences **21**(6), 1825–1845 (2021)

[31] Purohit, H., Sheth, A.P.: Twitris v3: From citizen sensing to analysis, coordination and action. In: Proc. of the 7th ICWSM, 2013, pp. 746–747. AAAI press, ??? (2013)

[32] Imran, M., Castillo, C., Lucas, J., Meier, P., Vieweg, S.: Aidr: Artificial intelligence for disaster response. In: 23rd International Conference on World Wide Web, pp. 159–162 (2014)

[33] Burel, G., Alani, H.: Crisis event extraction service (crees)-automatic detection and classification of crisis-related content on social media. In: International Conference on Information Systems for Crisis Response and Management (ISCRAM), pp. 1–12 (2018)

[34] Lagerstrom, R., Arzhaeva, Y., Szul, P., Obst, O., Power, R., Robinson, B., Bednarz, T.: Image classification to support emergency situation awareness. Frontiers in Robotics and AI **3**, 54 (2016)

[35] Weber, E., Papadopoulos, D.P., Lapedriza, A., Ofli, F., Imran, M., Torralba, A.: Incidents1m: a large-scale dataset of images with natural disasters, damage, and incidents. IEEE transactions on pattern analysis and machine intelligence **45**(4), 4768–4781 (2022)

[36] Nguyen, D.T., Ofli, F., Imran, M., Mitra, P.: Damage assessment from social media imagery data during disasters. In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 569–576 (2017)

[37] Alam, F., Imran, M., Ofli, F.: Image4act: Online social media image processing for disaster response. In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 601–604 (2017)

[38] Alam, F., Ofli, F., Imran, M.: Crisismmd: Multimodal twitter datasets from natural disasters. In: International AAAI Conference on Web and Social Media, pp. 465–473 (2018)

[39] Koshy, R., Elango, S.: Multimodal tweet classification in disaster response systems using transformer-based bidirectional attention model. Neural Computing and Applications **35**(2), 1607–1627 (2023)

[40] Li, H., Caragea, D., Caragea, C., Herndon, N.: Disaster response aided by tweet classification with a domain adaptation approach. Journal of Contingencies and Crisis Management **26**(1), 16–27 (2018)

[41] Alam, F., Joty, S., Imran, M.: Domain adaptation with adversarial training and graph embeddings. In: Gurevych, I., Miyao, Y. (eds.) 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1077–1087. Association for Computational Linguistics, Melbourne, Australia (2018).

https://doi.org/10.18653/v1/P18-1099 . https://aclanthology.org/P18-1099

[42] Li, X., Caragea, D.: Domain adaptation with reconstruction for disaster tweet classification. In: 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1561–1564 (2020)

[43] Krishnan, J., Purohit, H., Rangwala, H.: Unsupervised and interpretable domain adaptation to rapidly filter tweets for emergency services. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 409–416 (2020). IEEE

[44] Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI Conference on Artificial Intelligence, pp. 2058–2065 (2016)

[45] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)

[46] Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 (2015). PMLR

[47] Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European Conference on Computer Vision, pp. 443–450 (2016). Springer

[48] Morerio, P., Cavazza, J., Murino, V.: Minimal-entropy correlation alignment for unsupervised deep domain adaptation. arXiv preprint arXiv:1711.10288 (2017)

[49] Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., Saminger-Platz, S.: Central moment discrepancy (CMD) for domain-invariant representation learning. In: International Conference on Learning Representations, pp. 1–13 (2017)

[50] Zellinger, W., Moser, B.A., Grubinger, T., Lughofer, E., Natschläger, T., Saminger-Platz, S.: Robust unsupervised domain adaptation for neural networks via moment alignment. Information Sciences **483**, 174–191 (2019)

[51] Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: International Conference on Machine Learning, pp. 2208–2217 (2017). PMLR

[52] Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 4058–4065 (2018)

[53] Balaji, Y., Chellappa, R., Feizi, S.: Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6500–6508 (2019)

[54] Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: European Conference on Computer Vision, pp. 597–613 (2016). Springer

[55] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: 30th International Conference on Neural Information Processing Systems. NIPS'16, pp. 343–351. Curran Associates Inc., Red Hook, NY, USA (2016)

[56] Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976. IEEE Computer Society, Los Alamitos, CA, USA (2017)

[57] Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244

[58] Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4500–4509 (2018)

[59] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning, pp. 1989–1998 (2018). PMLR

[60] Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., Yang, M.-H.: Diverse image-to-image translation via disentangled representations. In: European Conference on Computer Vision (ECCV), pp. 35–51 (2018)

[61] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176 (2017)

[62] Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)

[63] Na, J., Jung, H., Chang, H.J., Hwang, W.: Fixbi: Bridging domain spaces for unsupervised domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1094–1103 (2021)

[64] Chen, L., Chen, H., Wei, Z., Jin, X., Tan, X., Jin, Y., Chen, E.: Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7181–7190 (2022)

[65] Tian, Q., Yang, H., Cheng, Y.: Dynamic bias alignment and discrimination enhancement for unsupervised domain adaptation. Neural Computing and Applications **36**(14), 7763–7777 (2024)

[66] Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. Advances in neural information processing systems **19** (2006)

[67] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1), 151–175 (2010)

[68] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, pp. 1–11 (2015)

[69] Lee, J., Raginsky, M.: Minimax statistical learning with wasserstein distances. In: Advances in Neural Information Processing Systems, vol. 31, pp. 1–10 (2018)

[70] Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1853–1865 (2016)

[71] Kolouri, S., Zou, Y., Rohde, G.K.: Sliced wasserstein kernels for probability distributions. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5258–5267 (2016). https://doi.org/10.1109/CVPR.2016.568

[72] Villani, C.: Topics in Optimal Transportation. Graduate studies in mathematics. American Mathematical Society, ??? (2003). https://books.google.com.qa/books?id=MyPjjgEACAAJ

[73] Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

[74] Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: International Conference on Machine Learning, pp. 513–520 (2011). PMLR

[75] Chen, M., Xu, Z., Weinberger, K., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: International Conference on Machine Learning, pp. 767–774 (2012). PMLR

[76] Omeiza, D., Speakman, S., Cintas, C., Weldermariam, K.: Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. arXiv preprint arXiv:1908.01224 (2019)

[77] Li, D., Yang, Y., Song, Y.-Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5542–5550 (2017)

[78] Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5018–5027 (2017)

[79] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1406–1415 (2019)

[80] Fang, C., Xu, Y., Rockmore, D.N.: Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1657–1664 (2013)

[81] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

[82] Rota Bulo, S., Neuhold, G., Kontschieder, P.: Loss max-pooling for semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2126–2135 (2017)

[83] Alam, F., Alam, T., Hasan, M.A., Hasnat, A., Imran, M., Ofli, F.: MEDIC: a multi-task learning dataset for disaster image classification. Neural Computing and Applications **35**(3), 2609–2632 (2023)

[84] Gallego, A.-J., Calvo-Zaragoza, J., Fisher, R.B.: Incremental unsupervised domain-adversarial training of neural networks. IEEE Transactions on Neural Networks and Learning Systems (2020)