

Integrating Domain Knowledge in Multi-Source Classification Tasks


Alexandre Thurow Bender   [Federal University of Pelotas | atbender@inf.ufpel.edu.br]

Emillyn Mellyne Gobetti Souza  [Federal University of Pelotas | emgsouza@inf.ufpel.edu.br]

Ihan Belmonte Bender  [Federal University of Pelotas | ibbender@inf.ufpel.edu.br]

Ulisses Brisolará Corrêa  [Federal University of Pelotas | ulisses@inf.ufpel.edu.br]

Ricardo Matsumura Araujo  [Federal University of Pelotas | ricardo@inf.ufpel.edu.br]

 Postgraduate Program in Computing, Federal University of Pelotas, Rua Gomes Carneiro, 1, Pelotas, RS, 96010-610, Brazil.

Received: 03 February 2024 • **Accepted:** 20 June 2024 • **Published:** 29 June 2024

Abstract: This work presents an extended investigation into multi-domain learning techniques within the context of image and audio classification, with a focus on the latter. In machine learning, collections of data obtained or generated under similar conditions are referred to as domains or data sources. However, the distinct acquisition or generation conditions of these data sources are often overlooked, despite their potential to significantly impact model generalization. Multi-domain learning addresses this challenge by seeking effective methods to train models to perform adequately across all domains seen during the training process. Our study explores a range of model-agnostic multi-domain learning techniques that leverage explicit domain information alongside class labels. Specifically, we delve into three distinct methodologies: a general approach termed Stew, which involves mixing all available data indiscriminately; and two batch domain-regularization methods: Balanced Domains and Loss Sum. These methods are evaluated through several experiments conducted on datasets featuring multiple data sources for audio and image classification tasks. Our findings underscore the importance of considering domain-specific information during the training process. We demonstrate that the application of the Loss Sum method yields notable improvements in model performance (0.79 F1-Score) compared to conventional approaches that blend data from all available domains (0.62 F1-Score). By examining the impact of different multi-domain learning techniques on classification tasks, this study contributes to a deeper understanding of effective strategies for leveraging domain knowledge in machine learning model training.

Keywords: Multi-Domain Learning, Batch Regularization, Classification Task, Image, Audio

1 Introduction

Training machine learning models with limited data poses a well-recognized challenge, as the scarcity of examples may impede the ability of the model to generalize effectively to unseen data [LeCun *et al.*, 2015; Domingos, 2012]. However, recent years have seen a notable shift in the focus of research toward the critical role of data quality in achieving high-performance models [Jain *et al.*, 2020; Westermann *et al.*, 2022; Sambasivan *et al.*, 2021]. Constructing datasets that accurately represent real-world scenarios while ensuring an ample supply of labeled examples is inherently challenging. This challenge stems from the substantial costs associated with collecting and annotating samples in natural settings compared to the relative ease of automatically capturing or generating examples and their labels in controlled environments. Moreover, even when efforts are made to mimic real-world conditions, data collection often occurs under specific circumstances, such as using consistent devices (e.g. employing the same camera for image capture) or environmental conditions (e.g. recording audio clips indoors). Such collections of data, obtained or generated under similar conditions, are commonly referred to as domains or data sources.

The distinct conditions of data acquisition or generation are often neglected, yet understanding them is crucial to address any phenomena arising from these differences that

might impede model generalization. One such phenomenon is domain shift, which occurs whenever the distribution of the data used during training differs from that encountered during deployment [Quinero-Candela *et al.*, 2008]. Several factors can contribute to domain shift, including changes in the data generation process, variations in data capturing devices, or the presence of different types of noise or structure in the data. This inconsistency can lead to a significant decrease in performance for models trained on one domain but deployed on another, further evidencing the importance of considering the potential effects of domain shift when designing and deploying machine learning models in real-world applications.

There are several types of domain shift, and despite being discreet, covariate shift is arguably the most common [Quinero-Candela *et al.*, 2008]. It occurs when the input distribution of training and test is different, but the underlying task remains the same. For example, a model trained on images taken during daylight hours may perform poorly when tested on images taken at night, irrelevant to the task.

One of the main challenges in dealing with domain shift is the lack of labeled data from the target domain [Ganin and Lempitsky, 2015], which makes it difficult to adapt the model to the new distribution. Several methods have been proposed to tackle this problem, including domain adaptation techniques such as transfer learning [Weiss *et al.*, 2016],

adversarial training [Ganin *et al.*, 2016], and meta-learning [Vanschoren, 2018]. These techniques aim to align the distributions of the source and target domains or to learn domain-invariant representations.

Other notorious approaches in addressing the challenge of limited data are pre-training and fine-tuning. Machine learning models are commonly trained and evaluated using examples from the same domain. However, whenever there is limited data available for a specific task, a popular solution is to pre-train a model using out-of-domain information (usually in the form of a different, more extensive dataset) and then fine-tune it to the target domain. This technique has become favored over the past years [Niu *et al.*, 2020], as it is accessible to use while also allowing a faster training process. Another advantage of pre-training is the reduced risk of overfitting, notably when working with smaller datasets.

Fine-tuning a pre-trained model on a different dataset is a potential solution whenever there is a single target domain and performance in the pre-trained domain is not necessarily a concern. But whenever performance in the pre-trained domain becomes desirable, this approach might encounter difficulties. In fact, this is a major problem when training models on multiple domains [Ribeiro *et al.*, 2019]. Maintaining performance in an already trained domain while adding new knowledge to the model is a challenge of its own, as the model is prone to forgetting its previous knowledge [Goodfellow *et al.*, 2014]. This particular issue is called catastrophic forgetting [French, 1999], and to overcome it when learning multiple domains at the same time, other techniques must be used.

Traditionally, the standard approach is to mix all training data without any particular concern for their pertaining domains. While doing this might be enough given sufficient data, significantly large datasets and the computational power to train models using them are not easily attainable. One of the reasons for this approach to be acceptable in these conditions is the high difference across examples and domains: if the data does not have a prominent domain, the model is pushed towards domain-agnostic representations. In other words, the domain-specific characteristics in data samples are diluted for not holding a common structure, and as such, they are discarded as noise.

In the vast majority of cases, data does not display this richness of domains. Datasets often have no more than two or three sources of data acquisition. In light of these limiting factors, the potential benefits of using domain information from samples explicitly remain largely uninvestigated. This study explores techniques that incorporate domain knowledge during the training process of machine learning models when using datasets with multiple sources of data. As such, this work proposes injecting domain information by guaranteeing balanced representations of each domain in a batch, building upon the work of Bender [2022]. The latter investigated the concept of multi-domain regularization and proposed balancing information between domains in image data solely. The current study proposes novel approaches to domain regularization, further incorporates audio data in this analysis, and adds numerous experiments with different tasks and datasets comparing the techniques.

We investigate the effects of previously proposed ap-

proaches and expand them for further comparison. Differently from previous works, mostly which used only image classification tasks as a basis for evaluation, we assess the training methods using image and audio classification tasks, with a focus on the latter. To the best of our knowledge, there is a lack of batch-level domain regularization evaluations or proposals using audio data at the present date.

This research aims to understand the best way to learn from multi-domain datasets at once, dismissing the need to train multiple models for different situations. For this, we explore batch domain regularizations, which are usually overlooked in multi-domain learning. Additionally, we expect any potential gains in this regard will directly benefit smaller organizations and individuals with limited access to extremely large and varied datasets that overcome multi-domain issues.

This work represents an extended and revised iteration of our previous study Bender *et al.* [2023], with a heightened focus on exploring the implications of our methods within the context of audio classification tasks. Building upon the hypothesis that the training process of machine learning models for multi-domain tasks benefits from the explicit consideration of domain information, we present an array of new experiments designed for audio data. In addition to evaluating the general approach of mixing data from disparate domains during training, we include our two proposed methodologies that were previously confined to image data. The first, Balanced Domains, refines the general approach by ensuring a balanced representation of samples from each domain within every training batch. The second, Loss Sum, involves computing the loss of each domain separately using the cross-entropy loss function applied to individual batches, followed by the aggregation of these losses before initiating backpropagation. Through a series of experiments, we aim to elucidate the efficacy and adaptability of these methods in the context of audio data, thus contributing to a deeper understanding of domain-aware training strategies in machine learning.

As more diverse data sources are incorporated in training datasets (including in Large Language Models), it becomes necessary to better understand how to make the best use of this diversity. Our work proposes methods and shows benefits in incorporating domain knowledge into the training process. Its main contributions include (1) evaluating novel multi-domain learning approaches that use model-agnostic techniques; (2) identifying effective solutions for multi-domain problems; and (3) highlighting the importance of considering domain-specific information during the training process in machine learning problems.

The structure of the work is outlined as follows: Section 1 sets the context and outlines the motivation and objectives of the study; Section 2 offers a concise overview of the historical background of the field and reviews key concepts essential for comprehending the study; Section 3 examines notable prior research in the field; Section 4 details the methodology, encompassing the conducted experiments and their configurations; Section 5 discusses and analyzes the outcomes of the experiments; Finally, Section 6 presents final remarks and proposes potential avenues for future research.

2 Theoretical Background

This section provides key concepts relevant to understanding this work. It goes over a brief history of the area, loss functions, audio processing in neural networks, and multi-domain learning.

2.1 Loss Functions

Loss functions are mathematical cost functions commonly used to quantify discrepancies between predicted values and expected values in supervised machine learning algorithms. They serve as error measures to evaluate model performance. They quantify the disparities between predicted and expected values, guiding the adjustment of weight parameters during training.

In regression tasks, common loss functions like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) compare predicted and expected values using arithmetic mean and algebraic manipulations to address signal variations. Conversely, classification tasks necessitate different approaches due to discrete class values. Instead of directly comparing classes, loss functions like Cross-Entropy operate on model logits, which represent class probabilities. By applying a softmax function to logits, they ensure a bounded interval for comparison. This approach improves training granularity by considering the certainty of predictions.

The concept of loss landscapes refers to model performance in weight space, crucial for optimization algorithms like gradient descent. While typically depicted in 2D for visualization, high-dimensional parameters may require dimensionality reduction techniques. Despite the problems of high-dimensionality visualization, the concept of loss landscape remains an important aspect to take into account when designing or analyzing objective functions and their optimizers.

2.2 Audio Processing

Audio is often defined as a form of sound that is limited within the acoustic range humans are biologically capable of hearing. Beyond that, audio is a signal. A signal can be understood as a quantity that changes over time. For audio, the quantity in question is air pressure. To store this information digitally, it is necessary to sample it. Sampling is the process of measuring a signal at discrete points in time. The most common sampling rate for audio is 44.1kHz, which means that 44,100 samples are taken per second. By digitizing audio in this way, it becomes possible to manipulate it in a variety of ways, including editing, processing, and transmitting it to other devices. It is an essential aspect to many technologies.

Once an audio signal is sampled and digitized it becomes possible to visualize its audio wave (Figure 1). The audio for the example plots is the first 12 seconds of the song "Playing God", by Tim Henson¹. While raw audio waves contain valuable information about the sound signal, most of this knowledge remains concealed in the frequency domain.

¹<https://youtu.be/DSBBEDAGOTc>

One of the reasons for this is that audio waves are rich, complex signals that contain multiple frequency components that typically overlap and interfere with each other. As a result, extracting and analyzing the complex behavior of the audio signal often requires additional techniques.

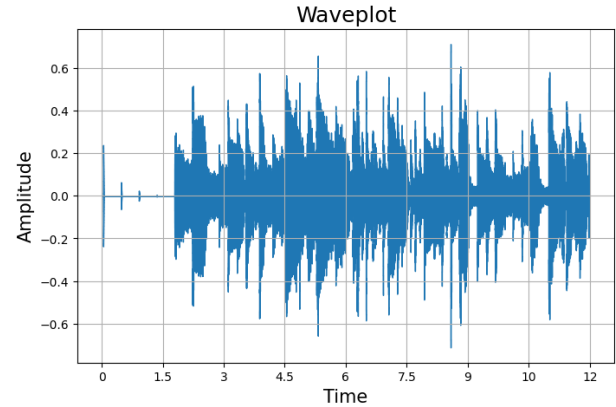


Figure 1. Raw audio waveplot depicting the first 12 seconds of the song *Playing God*, by Tim Henson. This same excerpt is used for forthcoming example plots. Source: Author.

Unlike the time domain of a signal, which shows variations in the signal over time, the frequency domain decomposes a signal across its constituent frequencies and shows how much of each frequency component is present. This is desirable because most types of audio signals, such as speech, music, or environmental sounds, have distinct characteristics in terms of frequency patterns and structures.

Digital signal processing techniques are indispensable tools for analyzing and manipulating signals in many fields, such as communication systems, image processing, and audio processing. One of the most commonly used techniques in digital signal processing is the Fourier transform. It allows us to traverse between the time and frequency domains of a signal, which is essential for analyzing signals with complex frequency components.

The Fourier transform is a mathematical technique that converts a signal from the time domain to the frequency domain (Figure 2). It decomposes a signal into a sum of sine and cosine waves of different frequencies, each with its own amplitude and phase. By analyzing the frequency components of a signal, we can extract valuable information about its behavior and characteristics, such as its dominant frequency, harmonics, and noise. In audio processing, the Fourier transform is used for a wide range of tasks, such as speech recognition, music analysis, and noise reduction.

Despite being a powerful tool for signal analysis, applying the Fourier transform to the signal in its entirety may yield uninformative results, making it difficult to understand the properties of the signal. The main limitation of this approach is that it assumes the signal is stationary. This means it considers the frequency content of the signal to remain constant over time. Expectedly, real-world signals are commonly non-stationary, i.e. their frequency content changes over time. Spectrograms provide an alternative approach to deal with non-periodic signals, addressing the issue by breaking the signal into small segments before computing the Fourier transform for each part. Being so, this approach

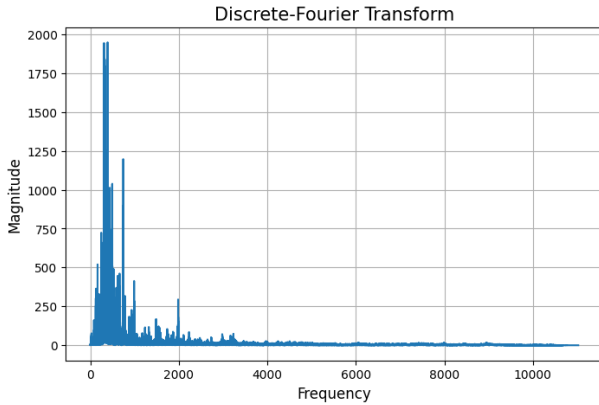


Figure 2. Discrete-Fourier Transform depicting the frequency domain of the example song (time component information is lost). Notice how certain frequencies dominate the signal. Source: Author.

allows us to track how the frequency components in the signal change over time. Therefore spectrogram representations allow for the capture of frequency domain properties that would otherwise be missed should we apply the transform on the entire signal all at once.

Most frequencies contribute very little to the overall amplitude of the sound. For this reason, spectrograms commonly use the logarithmic scale to represent the frequency content of signals. Their values are usually expressed using decibels (dB), the same logarithmic scale used to measure the power of sound waves. In spectrograms, the intensity of the sound at each frequency and time point is commonly represented with different colors representing the intensity levels. Because their values are in the decibel scale, it becomes simple to compare the relative loudness of different parts in the signal.

In addition to using the logarithmic scale to present the amplitude of the frequency components (color axis), spectrograms typically also use a logarithmic scale on the frequency axis (y-axis) as well. They do so to represent signal information in a manner that is consistent with human auditory perception, for the relationship between the frequency of sounds and how we perceive their pitch is logarithmic as well. Consider a sound with a frequency of 100Hz being doubled to 200Hz: we perceive this difference in pitch as being the same as if we had doubled the frequency again to 400Hz. In fact, humans generally perceive an octave (a doubling in frequency) as being the same change in pitch, regardless of the starting frequency. Ultimately, this also implies we are better at detecting differences in lower frequencies than higher ones. It is trivial to tell the difference between 500Hz and 1,000Hz, but it can be hard to distinguish between 10,000Hz and 10,500Hz, despite their distance being the same. For this reason, spectrograms commonly use a specific logarithmic scale called the Mel scale. It is in fact a perceptual scale of pitches judged by its listeners to be equal in distance from one another. Historically, there have been several proposals to define a psychophysical pitch scale dating back to 1937. Since then, the curves depicting the conversion of f hertz to m mels have evolved to the now-popular version with the 700Hz corner frequency published in 1976 [Makhoul and Cosell, 1976], which can be seen in Equation 1.

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (1)$$

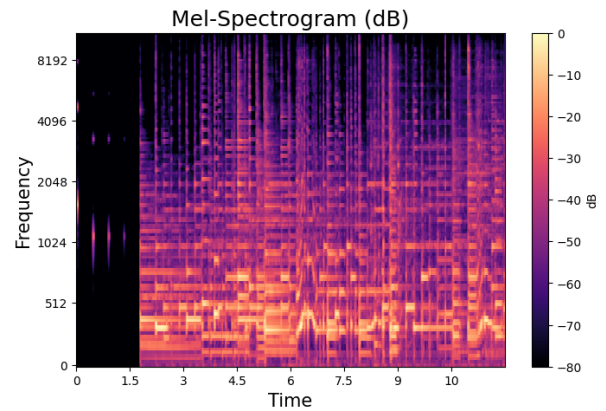


Figure 3. Audio mel-spectrogram plot of the same example song. Notice how structural patterns in the signal are more evident in the Mel scale. Source: Author.

The overall result of using the Mel scale is a better visualization of the low-frequency components of the signal (Figure 3), which can be difficult to see on an otherwise linear scale. Another relevant reason to use such a scale on the frequency axis is to compress the dynamic range of the signal. This type of scale compresses large values, making it easier to see small frequency changes throughout the signal. Notably, this is useful when working with signals with a wide dynamic range, including music or environmental sounds, while also offering a scale conformable to human auditory perception.

2.3 Multi-Domain Learning

Despite the significant amount of data available nowadays, current training paradigms are restricted in terms of the variety of data they can handle. Typically, models are trained and work with a single data source, usually from a narrow domain. Inevitably, models learn their structural patterns and become biased toward that particular domain, performing well only when working within it. This is a major limitation in terms of generalization when models are expected to perform well in multiple scenarios. Multi-domain learning is concerned with learning multiple domains simultaneously. This paradigm allows models to learn from a variety of domains without harming their ability to learn more nuanced features structurally inherent in each domain.

To better understand the significance of domains conceptually, it is useful to view them through the lens of a task. Mathematically, whenever models are being trained on a task, they are learning a mapping function from the domain (the data) and the image (model output). Even though we commonly refer to the task in a more abstract manner (e.g. animal classification using images), in reality, the task being learned is much more strict. The learned task could potentially be "differentiating very specific animals using photos taken using a DSLR camera with a particular sensor during an exact time of day with determined weather conditions". In fact, the learned task is very specific to the input data, and much

expectation is placed on the ability of models to generalize *ad infinitum*. This often creates a dissonance between the task machine learning specialists are trying to solve and the task the model is being trained on. Not rarely do models fail to generalize to the data distribution in the actually intended task, a very literal instance of solving the wrong problem.

Domain differences lead to errors in a number of ways [Ben-David et al., 2006, 2010]. Domain-specific distributions often differ in favoring different features. As such, some features may only appear in one domain. Additionally, features may behave distinctly regarding the label distribution in each domain.

Examples using image data are useful to understand how domain differences manifest. One of the widely used datasets in multi-domain research for images is the Office-31 [Xu et al., 2021; Na et al., 2021; Kang et al., 2019; Xu et al., 2019]. The Office-31 dataset is suitable for multi-domain learning studies since it has different domains but maintains the same classes across domains (this is an important characteristic and will be further elaborated in Section 4). Its three domains are: Amazon, Webcam, and DSLR [Saenko et al., 2010].



Figure 4. Examples from the Office-31 dataset. Each line presents examples of the classes Bike, Headphone, and Scissors for a domain. The domains are Amazon, DSLR, and Webcam, from top to bottom. Source: Author.

Domain differences in audio are more subtle and are very difficult for humans to perceive in spectrograms. Figure 5 depicts a comparison between the same song from Figure 3 and an artificially mixed version of it, where city ambient noises were introduced to simulate a noisy domain. Despite being the exact same song, the spectrograms are only vaguely similar on first inspection. In fact, without the information about what exactly is contained in each audio clip, it would not be trivial to point out any potential domain differences just by looking at the spectrograms (contrasting to Figure 4).

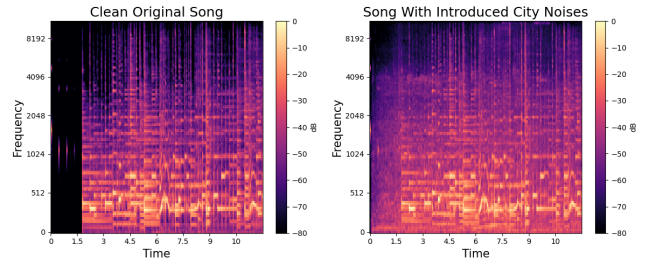


Figure 5. Audio mel-spectrograms of the original song example and an alternative version of it, mixed with city background noises. Manually investigating domain differences and their properties via mel-spectrogram visualization is not a reasonable task for humans. Source: Author.

3 Related Works

Within machine learning research areas there are several sub-fields dedicated to finding and analyzing the best way to train a model in multi-domains at the same time. Among these, domain generalization aims to make a model perform well when using as a test a domain that was not used during its training [Gulrajani and Lopez-Paz, 2020]. Another task, called multi-domain learning seeks to find the best way to train a model so that it performs adequately in all domains used during training [Liu et al., 2019].

During the last few years there has been a growing research interest that addresses the training of models with multi-domains, with research focusing on domain generalization [Gulrajani and Lopez-Paz, 2020; Arpit et al., 2021; Laparra et al., 2020; Xie et al., 2018; Li et al., 2018, 2017] but few studies have been developed in multi-domain learning. Although many of the single-model multi-domain learning contributions end up proposing domain-specific architectural changes [Sicilia et al., 2021] and although these solutions theoretically consist of the use of a single model, it is still necessary to create a different architecture for different datasets or if new domains are added to the original, making it difficult to scale and adding complexity. Typically, multi-domains are manipulated by creating multiple branches within the neural network, one for each domain to be learned, which shares the initial part of the network as the feature extractor and domain decider.

Nam and Han [2016] proposed MDNet, referred to as Multi-Domain Network. Their approach separates domain-independent information from domain-specific one and learns generic feature representations for video-tracking. To enable this, each domain in MDNet is trained individually while the shared layers of the network are updated in every iteration. In their study, each video sequence in their task of visual tracking is referred to as a domain. Therefore, the proposed MDCNN requires retraining.

In Chen et al. [2018], Chen advocated for BAMDCNN, a Branch-Activated Multi-Domain Convolutional Neural Network for the task of visual tracking. In addition to the main convolutional layers of the CNN, the network has additional branch layers, each specializing in handling a particular group. They extract key frames from the sequence dataset and group them using a clustering algorithm. During inference, they compare the similarity of the initial frame of test sequences across known groups to identify which inputs should be processed by which branches. As such, they achieve substantial effectiveness when compared to various

other state-of-the-art methods for visual tracking.

Liu *et al.* [2019] argued redundant common features often exist in the intersection of multiple domains, and models frequently learn those features as it is a simple and efficient way to address tasks. However, the existence of this redundancy in the network implies the model does not make full use of features and their learning spaces. In practice, this reduces the discriminability of the features and therefore increases the difficulty of the task. As such, the work addressed the undesirable mixture of features from different classes across domains by proposing an end-to-end network and orthogonality regularizations to separate domain-specific and domain-invariant features. The learning of what they refer to as compact-features (domain-specific features) significantly improves general classification performance.

One of the main contributions over the last recent years of learning multi-domains with the same model comes from speech recognition. SpeechStew [Chan *et al.*, 2021] performs state-of-the-art speech recognition tasks just by mixing all the data and training the same model using different domains, such as the Stew method we evaluate in this study. Other previous studies trained multi-domain models by mixing all available data [Chojnacka *et al.*, 2021; Narayanan *et al.*, 2018; Likhomanenko *et al.*, 2020], the main difference being that SpeechStew scales to larger models.

Batch-level domain regularizations were previously evaluated in the context of image classification by Bender [2022]. They obtain competitive performance using the Loss Sum approach, where the loss is calculated individually for domains and summed before backpropagation.

Notably, Tetteh *et al.* [2021] uses multi-domain balanced batch sampling techniques to address X-ray pathology classification tasks in the biomedical domain. They denote performance gains using a balanced batch sampling technique which is analogous to Loss Sum, previously proposed by Bender [2022].

More recently, Guo *et al.* [2023] propose DAMF (Domain-Adapting Moral Foundation), a data fusion framework designed for training moral foundation classifiers on heterogeneous datasets. They demonstrated its superiority over three distinct baseline methods. The framework involves performing transformations on text embeddings to result in domain-invariant representations. These representations are then discriminated using a domain classifier trained in an adversarial manner, compelling the encoder to learn embeddings that remain invariant across domains. Despite the modality of the work being focused on text data, this study underscores the ongoing concerns regarding different domains in recent research.

To explore the capabilities of vision transformers, Wang *et al.* [2023] introduce a framework for multi-domain vision tasks. It integrates tasks into a single supernet, optimizing them collectively. Key features include storage efficiency via parameter sharing, a coarse-to-fine search space, and two sharing policies for fine-grained parameter control. Their joint-subnet search algorithm challenges traditional practices by finding optimal architectures for each task. Their approach shows competitive performance and resilience to forgetting domains.

In their work, Wang *et al.* [2024] focus on the task of de-

tecting attempts to deceive biometric authentication methods, with a particular emphasis on the challenge posed by the unavailability of original training data due to privacy or other constraints. They observe that conventional approaches to training models on new data often result in forgetting the knowledge learned from the original data. To address this issue, they propose a novel method called multi-domain incremental learning, which aims to learn from new domains while also preserving performance across previous domains. Their approach achieves state-of-the-art performance compared to prior methods of incremental learning. Notably, they introduce adaptive domain-specific expert blocks to learn domain-specific knowledge separately, thereby mitigating interference between different domains.

While most studies in multi-domain learning propose architectural changes in models, we propose batch-level regularizations to guarantee appropriate domain representation in examples during the training of models. In fact, this is an architecture-agnostic approach and can be utilized without major alterations in the classical training loop of machine learning models.

4 Methodology

This section provides a detailed description of the procedures and techniques employed to conduct the study. It describes the systematic approach and methods used to address the research questions and objectives of the study. Additionally, we describe the experimental setup configuration.

4.1 Datasets

In order to evaluate the proposed multi-domain learning training methods, we need datasets that contain explicit domain characteristics. Additionally, the examples must have annotations depicting the domain they are a part of. Furthermore, we are interested in datasets containing an additional, distinct feature to use as the target of classification tasks. It is important to avoid direct relationships between the class target and the domain, as such interactions would hinder the evaluation of domain regularizations by confusing them with class regularization. In fact, when the domain has a direct relationship with the class label, figuring out the domain of an example is often reducible to discovering its class; being at least as complicated as correctly classifying samples (i.e. solving domain classification would imply solving target classification). Ultimately this means developers in this scenario do not have the domain information annotated or easily attainable. For this study, we select three datasets with these characteristics to perform the audio experiments: DAPS (Device and Produced Speech) containing book excerpt readings, and two bird call recording datasets, FF1010BIRD and WARBLRB10K. For the image experiments, we use the previously mentioned Office-31 dataset.

4.2 Image Experiments

The dataset containing images is called Office-31 and has three domains: Amazon, Webcam, and DSLR with 2817,

795, and 498 images respectively Saenko *et al.* [2010]. All domains have the same 31 classes that are composed of images of objects commonly found in offices. The Office-31 dataset was chosen in this study because it is widely used in research with multi-domain training, especially in domain generalization tasks Xu *et al.* [2021]; Na *et al.* [2021]; Kang *et al.* [2019]; Xu *et al.* [2019]. Although domain generalization studies have a different purpose, the Office-31 dataset is still suitable for multi-domain learning studies since the image set has different domains but maintains the same classes across domains.

The images in the Amazon domain are provided by an online sales store, therefore all images have a white background where their objects are in a unified color scale. The Webcam domain has low-resolution images (640x480) with significant noise. Finally, the DSLR domain is composed of high-resolution images with low noise (4288x2848). Figure 4 demonstrates examples of images from the Bike, Headphone, and Scissor classes, where the first line presents images from the Amazon domain classes, the second line presents images from the DSLR domain classes and the last line presents images from the Webcam domain classes.

Perceptibly, the amount of samples between domains is highly unbalanced. This can be a problem when training multi-domain learning models since they can present a bias towards the largest domain, especially when using the Stew approach, where the majority domain would present a larger amount of samples than the other two domains, which could cause the model prioritizes samples from the Amazon domain to obtain a smaller Loss. In addition, within the same domain, the images between the classes are also unbalanced, which may also be responsible for a bias in the training of the model, where it may prioritize classes with a greater number of samples over those with a smaller number. Another inappropriate factor in correct training and evaluating multi-domain learning models using the Office-31 dataset is the fact that there are identical or very similar images within the same class in the same domain, which can generate a kind of data leak.

To deal with these challenges, we remove all duplicate or very similar images within the same class across all domains. Then, after separating the training and testing sets, the training set is balanced so that all classes, within their respective domains, have approximate amounts through the use of the `WeightedRandomSampler` function (from the library of machine learning `PyTorch`). Finally, to avoid possible training bias for the larger domain, samples of equal sizes between domains are used so that all domains have the same total number of images and approximately the same number of images between classes.

Due to the imbalance in the number of images between the domains of the Office-31 dataset and seeking to avoid overfitting for the domain with the highest number of images, around 300 images of each domain are used for training, and 155 images of each domain are used for testing. This image amount is defined according to the number of images available in the smallest domain. In addition, all images used in training and testing were resized to 224x224 because it is the default size of the ResNet-50 input layer. No other pre-processing was applied to the images.

4.2.1 Speaker Identification — DAPS

One of the audio datasets is called Device and Produced Speech (DAPS) Mysore [2014] and contains speech segments of 20 different readers (10 male and 10 female readers) in various recording device types and environmental conditions (15 different domains). Each speaker read 5 public domain book excerpts under different conditions (about 14 minutes of duration per speaker). In its entirety, the dataset consists of about 4 1/2 hours of audio recordings. DAPS was initially used as a speech recognition dataset, but we decide to use it for the task of speaker recognition, classifying the 20 different speakers. We focus on classification problems in this study because, typically, they are more straightforward, thus reliable alternatives to testing new multi-domain train paradigms.

It is expected to encounter domain shift regarding the difference in data recording conditions, i.e. audio clips recorded using an iPhone in a conference room will likely differ in characteristics from those recorded by an iPad in a balcony prone to street noise. Despite noise being a more intuitive cause of domain shift, the differences in recording devices and room acoustic conditions likely also play an important role.

Each domain is split into train and test folds. We do not use a validation fold as we are not optimizing hyperparameters or performing optimization tasks in the model configuration. How each domain is split is important and requires attention to a few details. Note this is a classification task with 20 different classes (the speakers), thus it is important to guarantee a balanced representation of these classes in training and test sets. We use class stratification to address this issue, while also guaranteeing a somewhat even distribution of text scripts and speaker gender.

Each audio clip is processed to handle trailing silence at the beginning and the end, as some speakers take significant time before they start talking. The former pre-processing is relevant as the clip is then split into 5-second segments, further avoiding examples without speech. Audio clips are then converted from waveform to mel-spectrograms. This representation visually represents the signal amplitude across different frequencies over time. Ultimately, spectrograms can be understood as the application of Fourier transforms on overlapping windowed segments of the signal. The mel scale is a unit of pitch to approximate the human perceived frequencies. The use of mel-spectrograms is common in audio processing because humans do not perceive frequencies on a linear scale.

However, we divide training and test sets before splitting the audio into 5-second recordings and then use that same fold scheme for the other domains. This is relevant because clips have a different duration from their counterparts in other sources, and would otherwise be unaligned across domains. Performing the train-test separation after splitting the audio tracks into 5-second segments would enable unaligned segments containing the same reader and script content but in distinct domains to be included in train and test sets. Some domain instances in audio are incredibly similar. Consider the signal differences of the same sentence being uttered in a conference room and living room. Detecting said differ-

ences could be hard even for a human listener. Therefore having the same (or very similar) audio content included in both train and test sets can cause data leakage leading to an over-optimistic result, even if said data originates from different domains.

4.2.2 Bird Detection — FF1010BIRD and WARBLRB10K

Freefield (FF1010BIRD) Stowell and Plumbley [2013] and Warblr (WARBLRB10K) are both bird detection datasets, but they do not have any domain semantics attributed to example classes. For this reason, we use them together, each behaving as a domain. Despite both being bird presence detection datasets, they are very different. In fact, Freefield is a dataset of professional recordings of on-site observations of birds (collected from the FreeSound online database²). It is very diverse in terms of location and environment. Expectedly, they use better recording equipment and usually there is not much background noise. Additionally, it has some label imbalance towards the negative class, supposedly because once the equipment is set on-site, it remains recording audio most of the time.

In contrast to FF1010BIRD, WARBLRB10K contains crowdsourced recordings of birds using the bird-watching smartphone app Warblr³. Its label imbalance is towards the positive class, as most users use their devices to record bird calls when in the presence of said birds. This dataset, however, has heavy background noise, including city sounds and even users imitating bird calls, allegedly to coax birds to answer. The recordings vary heavily in terms of audio quality, depending on the smartphone used.

The significant difference between the elected bird datasets is by design and desirable for this study, as domains too similar in nature would entail a difficult multi-domain analysis. The FF1010BIRD dataset contains only 25% bird presence, while the Warblr dataset contains 75% bird presence.

Table 1. Audio Experiment Results Across Domains

Dataset	Not Bird	Bird	Total
FF1010BIRD	5755	1935	7690
WARBLRB10K	1951	6045	7996

4.3 Evaluated Methods

Reiterating, our hypothesis is that, during the training of machine learning models on multi-domain tasks, the training process takes advantage of explicitly using this data and its domain. In order to evaluate it, we evaluate three methods for training models in multi-domain tasks, including the traditional method that does not explicitly considers the different domains. This section describes Stew, Balanced Domains, and Loss Sum.

4.3.1 Stew

The more intuitive approach to using data from multiple sources at the same time is the Stew method (named due to the SpeechStew method Chan *et al.* [2021]). The method consists of simply mixing data from multi-domains together homogeneously, without any special processing or distinction. This method is already in use for various multi-domain tasks in areas such as speech recognition.

To compose large datasets, it is common to use different sources containing the same data classes, so that the data come from different domains. Commonly these domains are not explicit, which makes the Stew method the only possible option without the need to perform complex analyses to infer domains. In this way, the Stew method is inherently present in most models trained using such datasets. Regardless of its simplicity, the Stew approach yields competitive results in multi-domain learning tasks, in particular whenever there are large amounts of data available.

Whenever a dataset has numerous well-represented domains, it is speculated to encourage the model toward domain-agnostic representations. Even if generic representations are desirable, the datasets containing the information necessary for a model to be capable of achieving such knowledge are rare. Not only is the creation of datasets a challenging endeavor, but the verification of it is often neglected. ImageNet has been the standard pre-train dataset for image-related tasks for the past years, and researchers frequently stumble on annotation errors and report them even at present.

4.3.2 Balanced Domains

Historically, there have been significant improvements in image classification using simple data processing methods and regularizations, such as dealing with label imbalance. During training, the Stew approach is understood to have no balance of domains whatsoever. Essentially, the batches are expected to have more samples from the majority domains, since batches are randomly sampled from the mixed dataset. This can be visualized in Figure 6. For this reason, there is room to explore regularization to address domain-level selection in training batches.

In classification tasks, the presence of a class imbalance in batches might hinder the model generalization, biasing it towards classes with more examples during the training process. Label balancing addresses this problem and can be especially useful in cases where there are few, highly unbalanced classes.

Similarly, we hypothesize a disproportionate domain presentation on a batch level might also interfere with the model learning. In this case, the model would potentially specialize in the majority domain. Intending to perform well in all presented domains, we propose a variation of the Stew method called Balanced Domains.

To compose the batches seen in model training, instead of sampling from a unique dataset, Balanced Domains samples from each of the available domains separately (as seen in Figure 6). The domain sample size is set to accommodate an equal (or close to equal) composition of each domain while maintaining the original batch size unaffected.

²<https://freesound.org/>

³<https://www.warblr.co.uk/>

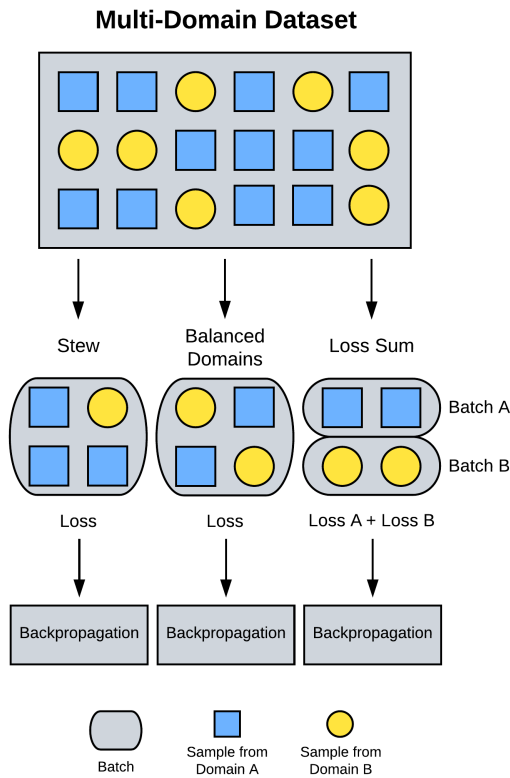


Figure 6. Evaluated methods illustrated. The Stew method is shown on the left, sampling randomly from the mixed datasets without any kind of balancing, usually having batches where the major domain has more examples. The middle flow describes Balanced Domains, where we also sample from the mixed domains dataset, but strive for similar amounts of each domain. On the right, Loss Sum flow is presented, where we sample from the domains individually and calculate a different loss for each domain batch, then sum them up to achieve the final loss. Notice that Loss Sum has smaller batches for each domain, but the number of samples in all batches amounts to the same batch size in previous methods. Source: Author.

This method is envisioned as a batch-sampling abstraction to enforce each domain to have approximately an equal number of examples in a batch. Thus the training step consists of sampling all obtainable domains, grouping the data into a single batch before presenting it to a model, and backpropagating the calculated loss from the batch.

The number of batches in an epoch is bounded by the largest domain. Whenever the domains differ in their number of samples, the smaller domain entries will be shown multiple times throughout a single epoch. Admittedly, this entails the same implications associated with oversampling and therefore requires the same cautionary practices.

4.3.3 Loss Sum

One way of penalizing the model whenever it underperforms on a training domain is achieved by balancing the domain representation. Another way to implement this is, instead of mixing domain samples into a single balanced batch, to calculate the Loss from each domain separately and sum the Loss across all domains before backpropagating it (as seen in Figure 6). Notably, this sort of approach was previously proposed by Bender [2022] and Tetteh *et al.* [2021] for image classification.

The Balanced Domains method regulates domains at a batch level, but this technique does not guarantee that the model will properly learn all domains since it might still prioritize domains with similar features. For instance, in cases where there are several domains, the difference in Loss of a small portion of them might not be enough to pose a difference. The intuition behind the Loss Sum approach is punishing the model with greater overall Loss values whenever it yields bad results in any domain. In this situation, penalizing the model by calculating the Losses individually and then adding them will lead to a higher Loss value.

It is relevant to note that although each domain is presented separately during model training, the total batch size between the three methods (Stew, Balanced Domains, and Loss Sum) remains unchanged. Loss Sum requires an additional call to the loss function for every domain, therefore it requires more training steps. Smaller batch sizes will also reduce the parallelization achieved by the GPU during training, effectively making it slower.

4.4 Counterfactual and Comparison Methods

This section describes additional methods, mainly counterfactual methods, useful to derive properties when comparing their results to the main methods described in the previous section.

4.4.1 Random Sum

Admittedly, the Loss Sum method operates on a different scale than other regular methods, and this is due to the fact it sums up the loss of multiple domains. Previous studies in image classification have suggested an increase in F1-Score when training neural networks using the Loss Sum approach. However, it is unclear whether this is due to the separate loss calculation and sum operation or due to simply having a higher loss value. For this reason, we devise a counterfactual method that shares the same scale (higher loss) as the Loss Sum method but does not apply the loss function to domains properly separated, but does so in mini-batches containing examples sampled randomly from the entire dataset. Thus, we refer to this method as Random Sum.

The idea behind this counterfactual experimental method is to understand whether the improved performance previously seen while using Loss Sum method is attributed to its higher loss values or some other mechanism. Should a method with a similar loss scale but trained without domain separation (e.g. Random Sum) performs similarly to the Loss Sum method, we may attribute its overall better performance to higher loss values. The reasoning for this potential outcome is related to how a higher loss entails more abrupt weight updates, which may or may not be appropriate for a given task. Alternatively, Random Sum failing to achieve competing performance with the Loss Sum (despite sharing its loss scale) would be evidence supporting the individual domain loss calculation mechanism.

4.4.2 Loss Mean

The Loss Mean method is another counterfactual method, complementing the Random Sum. It is also missing from previous studies with similar batch regularization proposals. In this approach, we perform the same procedure described for the Loss Sum process, but we divide it by the total of domains present in the task before backpropagating the loss. By doing this, we force the loss scale back to being comparable to other regular methods.

Depending on whether this experiment shows results similar to the Loss Sum method, we may collect further evidence for the separated loss calculation improving multi-domain learning tasks. Analogously, the hypothetical scenario where Loss Mean underperforms in comparison to Loss Sum would imply evidence for the higher loss scale being useful for the task (instead of the individual domain loss system).

4.4.3 Sequential And Inverse Sequential

Additionally, we configure two other training methods for comparison purposes. The Sequential training method is defined as training on individual domains in sequence. Conversely, the Inverse Sequential approach does the same, but in the reverse order of domains. Both of these approaches are prone to catastrophic forgetting (where the model forgets previous knowledge, replacing it with new information). Neither of these is expected to show competitive results with the other methods. However, we argue their results offer an interesting perspective on the learning tasks. As such, they enrich the comparison by showing the pitfalls of naive methods. Another argument to be made is that these methods behaving as expected conveys evidence of the correct implementation of the experiments.

4.5 Comparison Metrics

The baseline for our comparison is the Stew training method, as it is commonly used and *de facto* standard in the literature. The performance of models in an experiment for each method (Stew, Balanced Domains, Loss Sum, Random Sum, and Loss Mean) is calculated using the average F1-Score across domains. The F1-Score was chosen because it summarizes the learning objective: learning all domains at the same time while generalizing the classes.

When calculating F1-Score, we are presented with a choice of whether we use the macro F1-Score, the weighted F1-Score, or the micro F1-Score. The macro version calculates the F1-Score for each individual class, then returns their average value. It is an interesting alternative when the classes themselves are the object of investigation. When the classes are not balanced, that is, some classes have more representation in a given task, the weighted F1-Score provides an alternative by calculating the F1-Score for each class, and returning the weighted average of the result. The weights are typically calculated using the inverse of the example frequency from a given class. Finally, the micro F1-Score sums up the metrics for all classes and calculates a single, unified F1-Score for the model. This approach abstracts the concept of classes and focuses solely on how the model performs.

Because the object of focus for this study is the model itself and not the particular classes, we chose the micro F1-Score calculation. We argue choosing either macro or weighted F1-Scores would, in a way, hide aspects of the model performance behind averages and class weights. This is an unnecessary layer of abstraction that would, in fact, difficult model evaluation. This is particularly noticeable in cases where each domain has a different class distribution. Thus, considering the objective of evaluation in this study, projecting raw scores is the preferable alternative.

4.6 Evaluating Multi-Domain Learning Models

According to Gulrajani and Lopez-Paz [2020], there are two major approaches to multi-domain model evaluation that are rarely discerned and seldom discussed.

In the Leave-One-Domain-Out-Cross-Validation approach, one model is trained for every domain: each holding one of the training domains out. The withheld domain is then used to evaluate its corresponding model. The average of the score metrics across folds is then reported. Expectedly, domain characteristics can greatly impact this evaluation. When domains are similar enough, implicit data leakage can occur. Alternatively, in the scenario where the domains show significant distinct characteristics, covariate domain shift could be an obstacle. For this reason, we refrain from using it to evaluate the datasets.

In the Training-Domain Validation Set approach, each domain is split into training and testing subsets. The resulting partitions are pooled to create multi-domain train and test folds. The model is evaluated using the resulting overall test fold. This strategy assumes a certain similarity between training and testing example distributions. Overall, it is a conservative approach whenever prior knowledge of domain characteristics is limited. Thus, this approach is more appropriate for the current study.

Moreover, we purposely avoid using k-fold cross-validation for similar reasons. When working with multi-domain datasets, there are several relevant distribution characteristics we are interested in controlling cautiously. Take the DAPS dataset, for example: besides guaranteeing a similar distribution between train and test sets for the speaker id (which is the target feature), the speaker gender is also an important feature to control. Furthermore, the book excerpt is another dimension we are interested in controlling. And because we are interested in a more detailed task setup distribution-wise, the randomness from k-fold cross-validation could actually prove harmful for our evaluation. It is useful to remember the scope of this study is to evaluate the training methods and not to solve the classification tasks used for the evaluation.

4.7 Experimental Setup

We perform several experiments using the methods and datasets described previously in a Titan X Pascal graphics processing unit. We detail their configuration and the reasoning behind them in the subsections below. Their results are reserved for a dedicated section and can be inspected in

section 5. The code implementation for the experiments is available on GitHub⁴, including a Docker image to reproduce the experimental setup.

4.7.1 Differentiating Audio Domains Experiment

We are interested in analyzing the behaviors of the proposed methods in different situations regarding domain and class distributions. Before we do so, one important question to address is: how hard is it to distinguish audio domains? We have reviewed this from the human point of view in section 2.4. Whereas for images it is trivial for humans to distinguish different domains, for audio data it is unexpectedly hard to do so, either by hearing the audio clips (e.g. differentiating a recording in a conference room from the same recording in an office can be challenging), or by looking at the spectrogram. But more importantly, how hard is this task for computers to accomplish? If, similarly to humans, they prove ineffective for differentiating domains, then there would be no real benefit in using the training methods described in section 4.3. In fact, in this scenario the domain similarity would characterize a duplication of examples which could cause data leakage or just harm the learning process overall if not treated properly. Conversely, computers being good at distinguishing audio domains would be further evidence of the potential usefulness of these multi-domain training approaches.

To achieve an estimate of how effective computers are at addressing this task, we perform an experiment in which the task is domain classification in each experiment dataset configuration. For example, DAPS dataset samples will be classified as Ipad_Balcony, iPhone_Confroom, or any other of the domains. The merged bird dataset samples would be classified as either FF1010BIRD or WARBLRB10K. Note the actual class these samples pertain to (presence or absence of bird calls) is irrelevant to this task. In this experiment, we train a machine-learning classification model using a ResNet-34⁵ and the configuration described below. Its result can be inspected in Section 5.

4.7.2 Dataset Distribution Manipulation Experiments

We artificially partition the datasets to perform 5 major experiments (Table 2). The idea behind these experiments is similar in nature to the idea of ablation studies, a type of experimental analysis performed to understand the importance of specific factors, where researchers systematically modify or remove components to infer their impact on the model performance. The goal is to gain insight into the behavior of each method when used in different situations.

Experiments 1 to 4 use the bird detection datasets (WARBLRB10K and FF1010BIRD), while experiment 5 uses the speech dataset (DAPS). We choose the bird detection dataset to perform dataset manipulations, as it is rich in terms of class distribution differences between the domains. Additionally, it is easier and cheaper to train timewise.

The experiments use the ResNet-18 architecture⁶, pre-

trained with imageNet and fine-tuned using the DAPS or bird datasets (depending on the experiment). The choice of this particular network architecture is because it is relatively simple (the simplicity here is useful when comparing the different training methods and allows the execution of the numerous experiments necessary in this study), it is well-established in the literature, and the fact it is a convolution neural network suitable for the use with audio spectrograms. We use 10 epochs with a batch size of 256 and an 80/20 split for train/test. We maintain configurations across experiments whenever possible. This includes the deep learning neural network and hyperparameters. We do so because our focus is to evaluate multi-domain training methodological approaches and not hyperparameter tuning.

Each experiment is repeatedly performed with 30 repetitions with different seeds. The seed values influence the model weight parameter initialization values. In other words, they are responsible for the model configuration starting position in the loss landscape during the optimization process. It is important to guarantee this propriety when comparing the different training methods. Otherwise, some model instances could potentially start at more beneficial locations in the loss landscape, thus complicating the method comparison.

Another relevant propriety to consider is the order examples are shown to the model. The loss landscape traversal is greatly influenced by the order in which a model processes examples, and having it vary across learning methods will hinder a desirable fair evaluation.

- **Experiment 1 — Bird Detection, Original Dataset Size.** In this experiment, we use the proposed methods to train a model on the task of bird detection using the bird dataset (WARBLRB10K and FF1010BIRD) in their entirety. In this scenario, the datasets are similar in terms of example amount. We expect selecting examples stochastically domain-wise would not greatly affect the model, as each dataset would have a similar representation in the training dataset, in terms of example amount. Thus, this experiment evaluates the methods when domains are similar in terms of quantity, although being different in composition.
- **Experiment 2 — Bird Detection, Reduced FF1010BIRD.** We alter the configuration of the domains by artificially reducing one of them, using stratification to maintain the class distribution in each domain. For experiment 2, we randomly remove examples from the FF1010BIRD domain. In practice, this means domain WARBLRB10K is more influential during the training of the model when using a vanilla training approach. Expectedly, potential benefits from balancing domain presence in batches would appear in this scenario.
- **Experiment 3 — Bird Detection, Reduced WARBLRB10K.** This experiment is the natural counterpart of experiment 2. The reduced domain is now WARBLRB10K, while FF1010BIRD remains in its original characteristics. This is useful to review how the model behaves when most of the data comes from the ff distribution. Again, it is expected the regularization of the domains would be notable in this scenario should it im-

⁴<https://github.com/papercoderepo/integrating-domain-knowledge-jis2024>

⁵<https://pytorch.org/vision/torchvision.models.resnet34.html>

⁶<https://pytorch.org/vision/torchvision.models.resnet18.html>

Table 2. Experiment Description Summary

Experiment Number	Task	Datasets	Domains
Experiment 1	Bird Detection	WARBLRB10K, FF1010BIRD	2
Experiment 2	Bird Detection	WARBLRB10K, FF1010BIRD	2
Experiment 3	Bird Detection	WARBLRB10K, FF1010BIRD	2
Experiment 4	Bird Detection	WARBLRB10K, FF1010BIRD	2
Experiment 5	Speaker Identification	DAPS	14

prove model learning and generalization. In this experiment, we discard 1/3 of the WARBLRB10K domain, amounting to 5,598 removed audio clips. Only 2,400 examples remain.

- **Experiment 4 — Bird Detection, Reduced Symmetric.** Experiment 4 greatly reduces both domains to 300 samples, also using stratification to maintain the class distribution in each domain. The objective of this experiment is to evaluate the training methods when few examples are available. Despite the small number of examples, similarly to experiment 1, both domains are in equal amounts.
- **Experiment 5 — Speaker Identification, Original Dataset Size.** In this experiment, we use the DAPS dataset, which already has several domains defined. There are 15 domains in the original dataset. However, we decide to remove the domain `clean_raw` from this experiment, as it is notably distant from other domains (it is the only domain with breath sounds, lip smacks, and other speech noises). More importantly, it would differ very little from the `clean_speech` domain and would cause complications by having duplicated examples in the dataset (this is easy to happen when we split the original audio clips into 5-second segments). Despite the domain differences regarding acoustic conditions and recording devices, they are balanced in terms of class distribution.

5 Results

This section presents and discusses the experimental results of the audio classification tasks. We start by estimating how difficult it is for machine learning models to distinguish audio domains, then proceed by presenting the experimental results of dataset manipulations.

5.1 Image Experiments

In this section, the results obtained from training the Stew, Balanced Domains, and Loss Sum methods with the Office-31 dataset are presented. The multi-domain learning results are obtained by training models using combinations of the three domains (Amazon, Webcam, and DSLR) for each method. In this way, one experiment presents the results of training the model using only the Webcam and DSLR domains, the other experiment uses only the Amazon and Webcam domains, another experiment uses only the Amazon and DSLR domains, and finally, the last experiment is performed with the three domains Amazon, Webcam, and DSLR together.

In addition to the results of multi-domain learning, it is also possible to view the results of domain generalization, since the models that use only 2 of the 3 domains available for training can use the domain not seen in training to be tested in domain generalization.

All Stew, Balanced Domains, and Loss Sum methods show better multi-domain learning results in the Webcam and DSLR domains compared to the Amazon domain (Tables 3 and 4). This may have occurred due to the significant heterogeneity between the images within the same class in the Amazon domain, while the images within the same class of the Webcam and DSLR domains have greater similarity and may be more easily learned by the models.

In addition to this similarity between images within the same class of the same domain, the Webcam and DSLR domains have images similar to each other, sometimes presenting images of the same object but obtained differently, thus constituting different domains. These similarities between the DSLR and Webcam domains may explain the good domain generalization results in experiments that use one of these two domains in training and another (not seen in training) for testing. It is notable that, concerning the domain generalization task, the DSLR domain presented better results than the Webcam domain in all experiments (Tables 3 and 4). The same occurs with the multi-domain learning task, where 11 out of 12 experiments that used the Webcam and DSLR domains in training (with or without using the Amazon domain) show better results in the DSLR domain than in the Webcam domain. This may occur due to the Webcam domain presenting a high degree of noise, which may have negatively influenced the learning of this particular domain.

It is also possible to analyze that the Balanced Domains methodology performed, in general, with higher quality when compared to the Stew methodology (Tables 3 and 4). In the context of images, this can mean multi-domain learning has an advantage when the domains are balanced at a batch level. Likewise, domain generalization also shows better results in Balanced Domains compared to Stew, where it is possible that domain generalization also takes advantage of batch-level domain balancing).

Finally, when analyzing the results of the experiments with different combinations of domains between the Stew, Balanced Domains, and Loss Sum methods, it is possible to observe that the Loss Sum method presented the best results in both experiment groups, with a greater and lesser amount of training epochs (Tables 3 and 4). Thus, the results indicate presenting the domains individually potentially improves multi-domain learning and domain generalization.

We also test an additional counterfactual method called Random Sum. Random Sum is essentially Loss Sum but

with randomized domain data. The idea behind this experiment is to assert whether Loss Sum works because of a higher Loss value or because of its individual domain presentation and Loss calculation scheme. It is possible to verify that in the experiments with a smaller number of epochs, all the results of the Loss Sum method present better results in comparison with the Random Sum method. In experiments with a greater number of epochs, the Loss Sum method generally presents better multi-domain learning results, but the domain generalization results of the Random Sum method stood out. This demonstrates that multi-domain learning may benefit from Loss Sum more than domain generalization does.

Table 3. Image Experiment Results - F1-Score - 17 Epochs

Training Domains		Domain Evaluated		
		AMZ	DSLR	Webcam
Stew	DSLR, Webcam	0.508 ± 0.028	0.916 ± 0.022	0.892 ± 0.026
	AMZ, Webcam	0.710 ± 0.026	0.901 ± 0.025	0.906 ± 0.030
	AMZ, DSLR	0.704 ± 0.032	0.922 ± 0.031	0.884 ± 0.022
	AMZ, DSLR, Webcam	0.707 ± 0.019	0.962 ± 0.015	0.966 ± 0.009
Balanced	DSLR, Webcam	0.593 ± 0.035	0.980 ± 0.012	0.969 ± 0.012
	AMZ, Webcam	0.727 ± 0.033	0.927 ± 0.018	0.930 ± 0.019
	AMZ, DSLR	0.738 ± 0.040	0.948 ± 0.020	0.906 ± 0.021
	AMZ, DSLR, Webcam	0.719 ± 0.018	0.970 ± 0.012	0.971 ± 0.007
Loss Sum	DSLR, Webcam	0.610 ± 0.032	0.990 ± 0.007	0.988 ± 0.007
	AMZ, Webcam	0.767 ± 0.038	0.952 ± 0.017	0.971 ± 0.012
	AMZ, DSLR	0.772 ± 0.034	0.980 ± 0.012	0.940 ± 0.021
	AMZ, DSLR, Webcam	0.765 ± 0.026	0.989 ± 0.007	0.988 ± 0.006

Table 4. Image Experiment Results - F1-Score - 52 Epochs

Training Domains		Domain Evaluated		
		AMZ	DSLR	Webcam
Stew	DSLR, Webcam	0.568 ± 0.029	0.971 ± 0.012	0.976 ± 0.007
	AMZ, Webcam	0.748 ± 0.016	0.955 ± 0.014	0.937 ± 0.017
	AMZ, DSLR	0.740 ± 0.033	0.949 ± 0.014	0.943 ± 0.018
	AMZ, DSLR, Webcam	0.744 ± 0.028	0.978 ± 0.010	0.976 ± 0.006
Balanced	DSLR, Webcam	0.604 ± 0.037	0.983 ± 0.008	0.975 ± 0.009
	AMZ, Webcam	0.747 ± 0.023	0.961 ± 0.011	0.945 ± 0.023
	AMZ, DSLR	0.743 ± 0.031	0.958 ± 0.015	0.935 ± 0.023
	AMZ, DSLR, Webcam	0.756 ± 0.019	0.983 ± 0.013	0.982 ± 0.005
Loss Sum	DSLR, Webcam	0.611 ± 0.024	0.986 ± 0.006	0.981 ± 0.009
	AMZ, Webcam	0.767 ± 0.013	0.969 ± 0.014	0.954 ± 0.017
	AMZ, DSLR	0.757 ± 0.022	0.969 ± 0.010	0.949 ± 0.014
	AMZ, DSLR, Webcam	0.769 ± 0.024	0.991 ± 0.005	0.990 ± 0.007

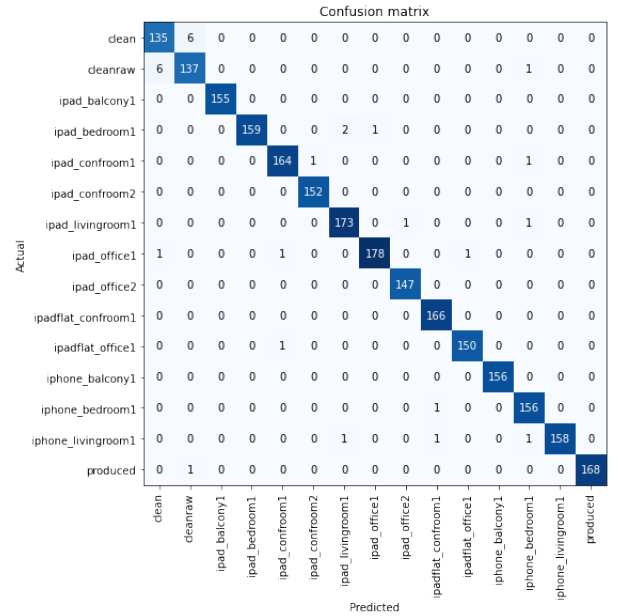
5.2 Differentiating Audio Domains Experiment

When using the DAPS dataset for domain classification, we achieve a 0.99 accuracy score in the test set using a ResNet34. A brief report of the training can be viewed in Table 5. Figure 7 denotes the confusion matrix for the test set classifications. We can further inspect a batch of model predictions in Figure 8.

When using the bird detection datasets WARBLRB10K and FF1010BIRD for domain classification, we achieve similar results of up to 0.959 validation accuracy score (the report can be seen in Table 6). It is interesting to note the first accuracy score values from the bird domain classification (binary classification problem using 2 domains) are dramatically worse than the DAPS domain classification (multiclass classification problem with 15 domains). At first sight, this

Table 5. DAPS Domain Classification Report

Epoch	Train Loss	Validation Loss	Accuracy
0	0.910	0.534	0.801
1	0.394	0.291	0.895
2	0.219	0.146	0.949
3	0.131	0.099	0.969
4	0.106	0.094	0.971
5	0.071	0.060	0.980
6	0.040	0.054	0.981
7	0.042	0.053	0.983
8	0.026	0.049	0.983
9	0.026	0.049	0.983

**Figure 7.** Confusion matrix of domain classification using the DAPS dataset. Source: Author.

is not an intuitive result, but it can be attributed to the significantly different amount of domain examples forming the bird dataset. WARBLRB10K forms approximately 48% of examples, while FF1010BIRD comprises 52% of them. Hence we can explain this lower initial score by the model initially guessing all examples as WARBLRB10K or as FF1010BIRD during the initial epochs. This behavior is further intensified by the bird dataset having only two classes. The test accuracy results also maintained the values seen before in validation, with the FF1010BIRD domain achieving 94.3%, and the WARBLRB10K domain achieving up to 97.2% accuracy. An example of predictions can be seen in Figure 9. A SHAP value analysis for image data can be reviewed in Appendix A.

Considering the use of machine learning to perform domain classification, based on the evidence presented in this section, we theorize this task pertains to the subset of tasks that are quite difficult for humans but unexpectedly trivial for computers. Possibly due to characteristics of individual frequency components either from recording sensors or domain acoustic features (not mutually exclusive). Notably, these results also reflect the potential of multi-domain regularization methods.

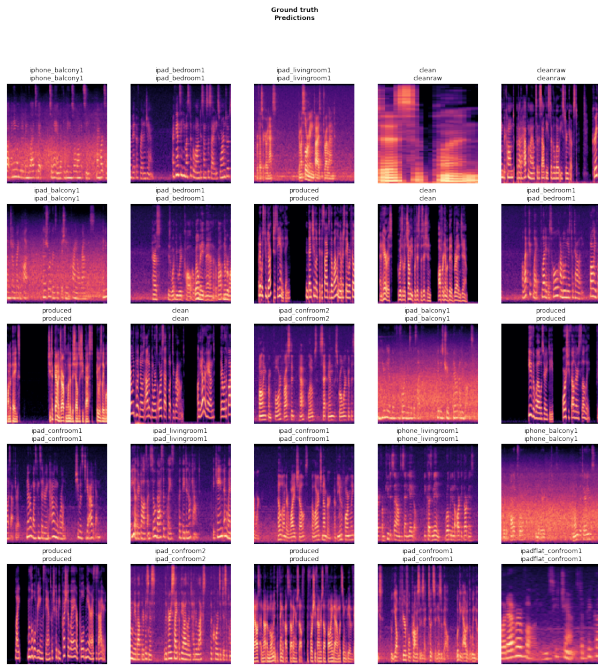


Figure 8. Illustration depicting model prediction of several examples in the DAPS dataset. Ground truth is the top label and model predictions are shown below it. Source: Author.

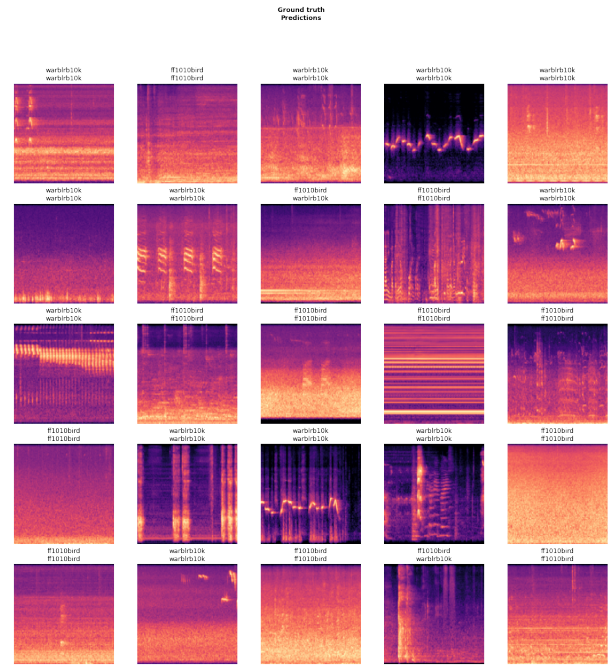


Figure 9. Illustration depicting model prediction of several examples in the bird dataset. Ground truth is the top label and model predictions are shown below it. Source: Author

Table 6. Bird Domain Classification Report

Epoch	Train Loss	Validation Loss	Accuracy
0	0.435	3.841	0.484
1	0.315	6.093	0.522
2	0.289	2.140	0.490
3	0.222	0.178	0.940
4	0.175	0.696	0.668
5	0.149	0.175	0.942
6	0.141	0.133	0.953
7	0.124	0.168	0.944
8	0.110	0.116	0.960
9	0.104	0.114	0.960

5.3 Dataset Distribution Manipulation Experiments

This section presents the results of the multi-domain learning experiments in bird classification (Experiments 1 – 4), and speaker identification (Experiment 5).

5.3.1 Experiment 1 — Bird Detection, Original Dataset Size

This experiment uses all of the bird detection datasets (which includes FF1010BIRD and WARBLRB10K as domains). The summarized results can be viewed in Table 7. The summarized results omit the standard deviation and the results from the sequential and inverse sequential methods (which are less relevant to the discussion here). For the expanded table, refer to Appendix B.

When looking at the average score of each method in Table 7, we notice the best result is from using the Loss Sum approach. Additionally, the Random Sum results are far worse than Loss Sum, despite being in the same loss scale. This is evidence against the argument stating that Loss Sum is

better because of its higher loss scale. In fact, even the baseline Stew approach performed better than Random Sum. Furthermore, the Loss Mean method performs similarly to Loss Sum, despite not operating in the higher loss scale. This is yet another argument against the higher loss scale being responsible for the Loss Sum improved performance. Notably, the Balanced method also improved the average performance when compared to the Stew baseline. However, there is a tradeoff where the war domain increased in performance at the cost of lower performance in the FF1010BIRD domain.

We are also interested in how each method evolves regarding its performance during training (some methods may cause a faster generalization during training than others). Figure 10 denotes the test F1-Score evolution during training. We see a significant increase in the F1-Score of the Loss Sum and Loss Mean methods. The Balanced, Stew, and Random Sum methods perform similarly, with the Balanced method being slightly better than the Stew baseline, and the Random Sum performing slightly worse than the baseline. Below we see the Sequential and Inverse Sequential methods, where the moment the training domains change is evidenced by the sudden decline in the 5th epoch.

Detailed visualizations of loss curve convergence can be seen in Appendices C.

5.3.2 Experiment 2 — Bird Detection, Reduced FF1010BIRD

After reducing the FF1010BIRD dataset, its baseline performance using the Stew method dropped (Table 8 and Figure 11). This was expected, as there are fewer examples to learn from in this domain. Conversely, the performance in the WARBLRB10K domain improved for the same reason: it has more examples, and as a result the model focuses on learning its characteristics and achieves better results on it.

The Balanced approach achieved worse results in compar-

Table 7. Experiment 1 — Bird Detection, Original Dataset Size, Micro F1-Score (Summarized)

Domain	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.623	0.694	0.803	0.618	0.799
FF1010BIRD	0.631	0.574	0.779	0.627	0.779
Average	0.627	0.634	0.791	0.622	0.789

Table 8. Experiment 2 — Bird Detection, Reduced FF1010BIRD, Micro F1-Score (Summarized)

Domain	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.745	0.641	0.797	0.752	0.793
FF1010BIRD	0.475	0.605	0.793	0.512	0.796
Average	0.610	0.623	0.795	0.632	0.795

Table 9. Experiment 3 — Bird Detection, Reduced WARBLRB10K, Micro F1-Score (Summarized)

Domain	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.520	0.649	0.757	0.529	0.768
FF1010BIRD	0.745	0.647	0.771	0.737	0.773
Average	0.632	0.648	0.764	0.633	0.771

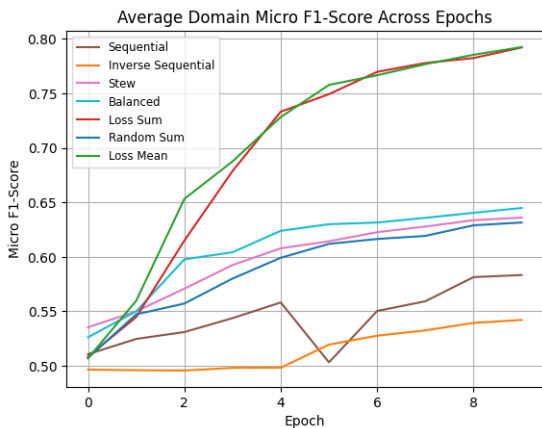


Figure 10. Average test set domain micro F1-score across epochs for Experiment 1. Source: Author.

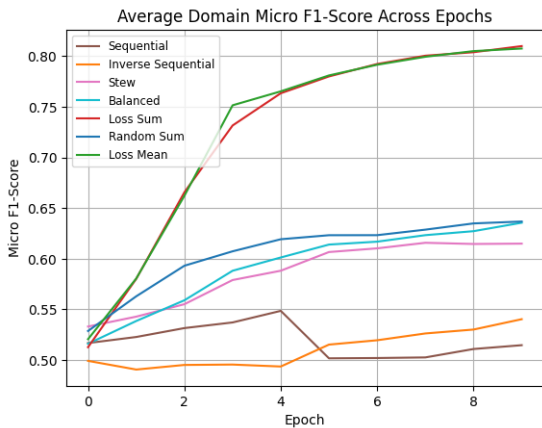


Figure 11. Average test set domain micro F1-score across epochs for Experiment 2. Source: Author.

ison to using the entire dataset. However, compared to the Stew method, it has maintained performance in the minority dataset (FF1010BIRD), at the cost of some of the performance from the majority dataset. Again, this is aligned with

our previous expectations, as the intent behind Balanced Domains is to act as regularization to force the neural network to perform well across domains.

Remarkably, the Loss Sum method achieves slightly better performance when compared to Experiment 1. Despite losing performance in the WARBLRB10K domain, it achieves better performance in FF1010BIRD. This improvement is not entirely expected and might be attributed to its smaller, less reliable test set. Nevertheless, maintaining similar performance to Experiment 1 is interesting evidence of how well Loss Sum performs when one of the training domains is notably smaller.

The Random Sum method in Experiment 2 performed better than the one in Experiment 1, but when we investigate the domain scores, we see it neglected the reduced domain, similar to the Stew method. The minor improvement is possibly attributed to Random Sum operating on a different loss scale. This is problem-dependent; while this characteristic may help in some problems, in others it will not.

Finally, Loss Mean performs very similarly to Loss Sum. It also potentially suffers from the same optimistic evaluation in the reduced domain.

5.3.3 Experiment 3 — Bird Detection, Reduced WARBLRB10K

Reducing the WARBLRB10K domain yields similar effects to Experiment 2. The Stew method performs marginally better, although it still prioritizes the larger domain (FF1010BIRD).

In this experiment, we have further evidence depicting how the Balanced method stops the model from focusing solely on the larger domain (Table 8 and Figure 12). Notably, Loss Sum and Loss Mean achieve the best results.

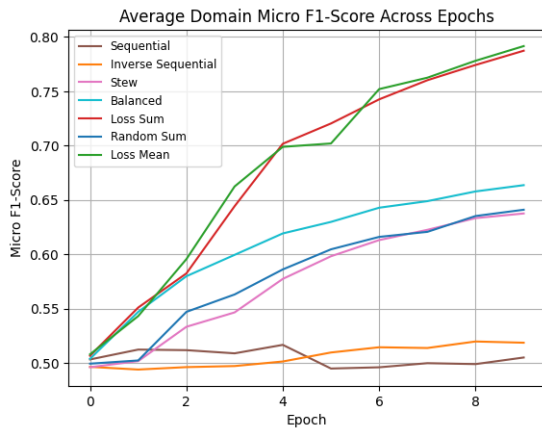
Furthermore, the Random Sum approach does not seem to yield competitive results despite operating on a higher loss scale. This provides evidence against the argument that

Table 10. Experiment 4 — Bird Detection, Reduced Symmetric, Micro F1-Score (Summarized)

Domain	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.603	0.538	0.588	0.569	0.656
FF1010BIRD	0.390	0.429	0.424	0.442	0.411
Average	0.497	0.483	0.506	0.505	0.533

Table 11. Experiment 5 — Speaker Identification, Original Dataset Size, Micro F1-Score (Summarized)

Domain	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
CLEAN	0.132	0.130	0.136	0.129	0.123
IPAD_BALCONY1	0.139	0.139	0.142	0.133	0.136
IPAD_BEDROOM1	0.134	0.130	0.130	0.125	0.127
IPAD_CONFROOM1	0.129	0.129	0.131	0.125	0.125
IPAD_CONFROOM2	0.135	0.132	0.135	0.128	0.121
IPADFLAT_CONFROOM1	0.140	0.137	0.140	0.128	0.133
IPADFLAT_OFFICE1	0.135	0.132	0.132	0.126	0.125
IPAD_LIVINGROOM1	0.135	0.131	0.131	0.128	0.126
IPAD_OFFICE1	0.129	0.127	0.133	0.123	0.121
IPAD_OFFICE2	0.137	0.134	0.134	0.133	0.130
IPHONE_BALCONY1	0.141	0.139	0.146	0.132	0.139
IPHONE_BEDROOM1	0.133	0.130	0.130	0.126	0.126
IPHONE_LIVINGROOM1	0.126	0.125	0.125	0.126	0.123
PRODUCED	0.134	0.130	0.139	0.129	0.128
Average	0.134	0.132	0.134	0.128	0.127

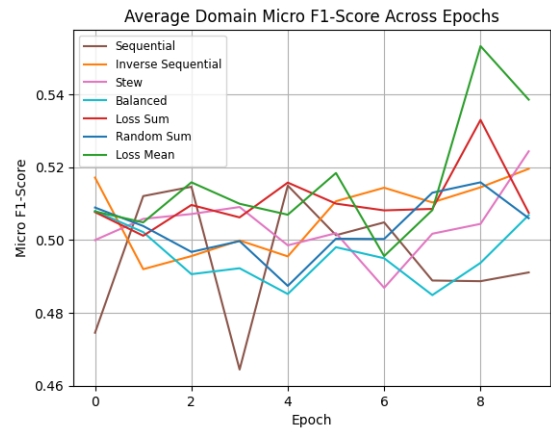
**Figure 12.** Average test set domain micro F1-score across epochs for Experiment 3. Source: Author.

higher loss values help in this particular task.

5.3.4 Experiment 4 — Bird Detection, Reduced Symmetric, Micro F1-Score

Reducing both domains to a few hundred examples causes model performance to drop dramatically (Table 10). It is relevant to remember this configuration is balanced, as both domains have the same amount of examples.

In this extreme scenario, the Stew approach still prioritizes one of the domains. Even though the Balanced method performs worse than the Stew one, it mitigates domain favoritism. Note there is a significant tradeoff where the favorite domain drops in performance. The Loss Sum and Random Sum perform similarly in this scenario, with the Loss Mean approach yielding the best results.

**Figure 13.** Average test set domain micro F1-score across epochs for Experiment 4. Source: Author.

The scarce amount of examples is insufficient for methods to learn domain-specific distributions. Bird domains contain different class distributions, and the F1-Score across epochs is erratic for all training methods (Figure 13).

5.3.5 Experiment 5 — Speaker Identification, Original Dataset Size

This experiment uses DAPS, a larger dataset with several domains. Thus, we evaluate it on a classification task with 20 different classes (Table 11). Expectedly, the results are worse than the previous experiments which used a binary classification task. Moreover, the results also suggest the number of epochs to approach this task could be increased for better results in the target task.

The Stew method achieved competitive results in this ex-

periment. Additionally, the Balanced approach marginally diminished the performance, when compared to the Stew baseline. This is not entirely unexpected, as the domains in this experiment are already balanced. Domain balancing in batches for an already-balanced multi-domain dataset mostly only introduces overhead.

Random Sum and Loss Mean performed below the comparison baseline. But the Loss Sum method achieved similar results to the Stew approach. However, there are some differences in the performance of individual domains. Overall, the training methods behave very differently when used on a high quantity of domains.

Overall, findings across experiments indicate that Balanced Domains and Loss Sum are particularly effective at reducing domain favoritism, especially when domain data availability is inconsistent. Reducing domain favoritism is relevant because it ensures generalization across diverse datasets and domains. When a model exhibits domain favoritism, it tends to prioritize certain domains over others during training, leading to biased predictions and reduced performance on unseen data from underrepresented domains. This is particularly important in tasks involving multi-domain learning, where the model needs to adapt to different data distributions and characteristics across domains. Additionally, reducing domain favoritism promotes fairness and equity in machine learning applications by ensuring that all domains are treated equally, regardless of their representation in the training data.

It is important to recognize that many tasks traditionally perceived as single-domain, actually involve data from multiple sources, albeit without explicit labeling of their origins. In these cases, mitigating domain favoritism becomes crucial. Failure to address domain variability can result in biased predictions and reduced performance on data from certain sources, undermining the reliability of the model in real-world scenarios. Consider, for example, a sentiment analysis task that aggregates user reviews from various websites. Although the reviews come from different domains (e.g., Amazon, Yelp, Twitter), they are typically treated as a single dataset for analysis. However, each domain may exhibit unique linguistic patterns, cultural nuances, or review tendencies that can influence the learning process of the model.

Our results provide evidence that understanding and leveraging domain differences can significantly improve predictions across diverse domains. By acknowledging and leveraging these domain-specific characteristics, batch-level domain regularizations enable models to learn more nuanced representations that generalize better to unseen data from different sources. This suggests that instead of treating all domains equally, models can benefit from allocating more resources to underrepresented domains.

It is noteworthy that the approaches we evaluated in our study are model agnostic, meaning they can be applied across various machine learning architectures without dependency on specific models. This characteristic shows the versatility and broad applicability of these methods in addressing domain variability differences in multi-domain tasks.

6 Conclusion

In this extended study, we evaluated multi-domain learning training approaches aimed at effectively regulating domain presence within batches for image and audio classification tasks, focusing on the latter. Our investigation centered on assessing the efficacy of the following methods: Stew, Balanced Domains, and Loss Sum. Additionally, we explored several counterfactual and comparison methods, including Random Sum, Loss Mean, and sequential approaches. Through experimentation and analysis, we aimed to provide deeper insights into the performance and behavior of these techniques in handling multi-domain learning challenges across datasets and task domains.

When handling domains with different class distributions, Balanced domains, and Loss Sum seemed to mitigate model domain favoritism. Particularly when some domains are limited in terms of data quantity. Loss Sum consistently presented competitive results in most experiments, improving baseline results in most scenarios.

Despite improving results in several scenarios, Experiment 4 evidenced the necessity of data quantity in domains to better leverage the regulation capacity of the evaluated methods. The experiments also provide evidence supporting the argument the loss aggregation methods benefit model training because of the individual domain calculation, and not because of the higher loss scale they operate in.

The results suggest that using explicit domain information by presenting them separately in individual batches for each domain potentially benefits the learning when training models in multi-domain tasks. This becomes more evident in experiments with fewer domains with unique class distributions.

It is important to address the limitations of the study. We refrain from hyperparameter tuning to be able to cover multiple dataset partitions, as it is a computationally intensive task. We also lack benchmarks or similar studies using this sort of approach for audio classification problems. As a result, validating or comparing these experiments is a difficult task.

Overall, multi-domain learning techniques using individual domain loss calculation, such as Loss Sum, provide an interesting strategy when dealing with multiple domains. Loss Mean performs similarly to Loss Sum likely due to the presence of a similar mechanism. However, according to our experiments, it is not, in fact, due to the higher loss values — as Loss Mean does not operate on a higher loss scale and often achieves competitive results as well.

Future studies could include evaluating prior domain knowledge, as certain approaches may demonstrate greater robustness to domain characteristics compared to others; Exploring alternative techniques for aggregating losses in domain datasets; Adding other multi-domain datasets; Replicating findings across tasks beyond classification, such as speech recognition.

Our study reinforces the importance of the often overlooked but critical understanding of the distinct conditions of data acquisition or generation. By exploring multi-domain learning techniques, we have highlighted the significance of considering how examples are presented to the model. We

hope that our research inspires further exploration of domain-aware training strategies to improve machine-learning models across various real-world applications.

Declarations

Funding

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. Furthermore, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001.

Authors' Contributions

Ricardo and Ulisses were involved in conceptualizing the work and overseeing its progress. Emillyn managed the data curation and performed the image experiments. Ihan was responsible for resource management and contributed to the initial investigation of the subject. Alexandre led the design of the methodology and conducted the audio experiments, being the main contributor to the conception of the study and the primary author of the manuscript. All authors have reviewed and endorsed the final manuscript.

References

- Arpit, D., Wang, H., Zhou, Y., and Xiong, C. (2021). Ensemble of averages: Improving model selection and boosting performance in domain generalization. *arXiv preprint arXiv:2110.10832*. DOI: <https://doi.org/10.48550/arXiv.2110.10832>.
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. (2010). A theory of learning from different domains. *Machine Learning*, 79:151–175. DOI: <https://doi.org/10.1007/s10994-009-5152-4>.
- Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2006). Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19.
- Bender, A. T., Souza, E. M. G., Bender, I. B., Corrêa, U. B., and Araujo, R. M. (2023). Improving multi-domain learning by balancing batches with domain information. In *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web*, pages 96–103. DOI: <https://doi.org/10.1145/3617023.3617037>.
- Bender, I. B. (2022). Evaluating machine learning methodologies for multi-domain learning in image classification. Master's thesis (computer science), Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas.
- Chan, W., Park, D., Lee, C., Zhang, Y., Le, Q., and Norouzi, M. (2021). Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*. DOI: <https://doi.org/10.48550/arXiv.2104.02133>.
- Chen, Y., Lu, R., Zou, Y., and Zhang, Y. (2018). Branch-activated multi-domain convolutional neural network for visual tracking. *Journal of Shanghai Jiaotong University (Science)*, 23:360–367. DOI: <https://doi.org/10.1007/s12204-018-1951-8>.
- Chojnacka, R., Pelecanos, J., Wang, Q., and Moreno, I. L. (2021). Speakerstew: Scaling to many languages with a triaged multilingual text-dependent and text-independent speaker verification system. *arXiv preprint arXiv:2104.02125*. DOI: <https://doi.org/10.48550/arXiv.2104.02125>.
- Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87. DOI: <https://doi.org/10.1145/2347736.234775>.
- French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135. DOI: [https://doi.org/10.1016/S1364-6613\(99\)01294-2](https://doi.org/10.1016/S1364-6613(99)01294-2).
- Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, Lille, France. PMLR, JMLR.org. DOI: <https://doi.org/10.48550/arXiv.1409.7495>.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030. DOI: https://doi.org/10.1007/978-3-319-58347-1_10.
- Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2014). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *2nd International Conference on Learning Representations, ICLR 2014*. DOI: <https://doi.org/10.48550/arXiv.1312.6211>.
- Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*. DOI: <https://doi.org/10.48550/arXiv.2007.01434>.
- Guo, S., Mokhberian, N., and Lerman, K. (2023). A data fusion framework for multi-domain morality learning. *Proceedings of the International AAAI Conference on Web and Social Media*, 17(1):281–291. DOI: <https://doi.org/10.1609/icwsm.v17i1.22145>.
- Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S., Mujumdar, S., Afzal, S., Sharma Mittal, R., and Munigala, V. (2020). Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*, pages 3561–3562. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3394486.3406477>.
- Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, CA, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR.2019.00503>.
- Laparra, E., Bethard, S., and Miller, T. A. (2020). Rethinking domain adaptation for machine learning over clinical language. *JAMIA open*, 3(2):146–150. DOI: <https://doi.org/10.1093/jamiaopen/ooaa010>.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. DOI: <https://doi.org/10.1038/nature14539>.
- Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017).

- Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, Venice, Italy. IEEE. DOI: <https://doi.org/10.1109/ICCV.2017.591>.
- Li, H., Pan, S. J., Wang, S., and Kot, A. C. (2018). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, Salt Lake City, UT, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR.2018.00566>.
- Likhomanenko, T., Xu, Q., Pratap, V., Tomasello, P., Kahn, J., Avidov, G., Collobert, R., and Synnaeve, G. (2020). Rethinking evaluation in asr: Are our models robust enough? *arXiv preprint arXiv:2010.11745*. DOI: <https://doi.org/10.48550/arXiv.2010.11745>.
- Liu, Y., Tian, X., Li, Y., Xiong, Z., and Wu, F. (2019). Compact feature learning for multi-domain image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7193–7201, Long Beach, CA, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR.2019.00736>.
- Makhoul, J. and Cosell, L. (1976). Lpcw: An lpc vocoder with linear predictive spectral warping. In *ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 466–469, Philadelphia, Pennsylvania, USA. IEEE, IEEE. DOI: <https://doi.org/10.1109/ICASSP.1976.1170013>.
- Mysore, G. J. (2014). Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges. *IEEE Signal Processing Letters*, 22(8):1006–1010. DOI: <https://doi.org/10.1109/LSP.2014.2379648>.
- Na, J., Jung, H., Chang, H. J., and Hwang, W. (2021). Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1094–1103, Nashville, TN, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR46437.2021.00115>.
- Nam, H. and Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, Las Vegas, Nevada, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR.2016.465>.
- Narayanan, A., Misra, A., Sim, K. C., Pundak, G., Tripathi, A., Elfeky, M., Haghani, P., Strohmaier, T., and Bacchiani, M. (2018). Toward domain-invariant speech recognition via large scale training. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 441–447, Athens, Greece. IEEE, IEEE. DOI: <https://doi.org/10.1109/SLT.2018.8639610>.
- Niu, S., Liu, Y., Wang, J., and Song, H. (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166. DOI: <https://doi.org/10.1109/TAI.2021.3054609>.
- Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2008). *Dataset shift in machine learning*. Mit Press.
- Ribeiro, J., Melo, F. S., and Dias, J. (2019). Multi-task learning and catastrophic forgetting in continual reinforcement learning. *arXiv preprint arXiv:1909.10008*. DOI: <https://doi.org/10.48550/arXiv.1909.10008>.
- Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226, Heraklion, Crete. Springer, Springer. DOI: https://doi.org/10.1007/978-3-642-15561-1_16.
- Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. (2021). “everyone wants to do the model work, not the data work”: Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, Okohama, Japan. ACM. DOI: <https://doi.org/10.1145/3411764.3445518>.
- Sicilia, A., Zhao, X., Minhas, D. S., O’Connor, E. E., Aizenstein, H. J., Klunk, W. E., Tudorasu, D. L., and Hwang, S. J. (2021). Multi-domain learning by meta-learning: Taking optimal steps in multi-domain loss landscapes by inner-loop learning. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 650–654, Nice, France. IEEE, IEEE. DOI: <https://doi.org/10.1109/ISBI48211.2021.9433977>.
- Stowell, D. and Plumbley, M. D. (2013). An open dataset for research on audio field recording archives: freefield1010. *arXiv preprint arXiv:1309.5275*. DOI: <https://doi.org/10.48550/arXiv.1309.5275>.
- Tetteh, E., Viviano, J. D., Kruege, D., Bengio, Y., and Cohen, J. P. (2021). Multi-domain balanced sampling improves out-of-distribution generalization of chest x-ray pathology prediction models. *Medical Imaging meets NeurIPS*. DOI: <https://doi.org/10.48550/arXiv.2112.13734>.
- Vanschoren, J. (2018). Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*. DOI: <https://doi.org/10.48550/arXiv.1810.03548>.
- Wang, K., Zhang, G., Yue, H., Liu, A., Zhang, G., Feng, H., Han, J., Ding, E., and Wang, J. (2024). Multi-domain incremental learning for face presentation attack detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5499–5507. DOI: [10.1609/aaai.v38i6.28359](https://doi.org/10.1609/aaai.v38i6.28359).
- Wang, S., Xie, T., Cheng, J., Zhang, X., and Liu, H. (2023). Mdl-nas: A joint multi-domain learning framework for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20094–20104.
- Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.
- Westermann, H., Savelka, J., Walker, V., Ashley, K., and Benyekhlef, K. (2022). Data-centric machine learning: Improving model performance and understanding through dataset analysis. In *Legal Knowledge and Information Systems: JURIX 2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8-10 December 2021*, volume 346, page 54. IOS Press, IOS Press. DOI: <https://doi.org/10.3233/FAIA210316>.
- Xie, S., Zheng, Z., Chen, L., and Chen, C. (2018). Learning

- semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, pages 5423–5432. PMLR, JMLR.org.
- Xu, T., Chen, W., Wang, P., Wang, F., Li, H., and Jin, R. (2021). Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2109.06165*. DOI: <https://doi.org/10.48550/arXiv.2109.06165>.
- Xu, X., Zhou, X., Venkatesan, R., Swaminathan, G., and Majumder, O. (2019). d-sne: Domain adaptation using stochastic neighborhood embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2497–2506, Long Beach, CA, USA. IEEE. DOI: <https://doi.org/10.1109/CVPR.2019.00260>.

Appendices

A Explainability Using SHAP

SHAP is an AI explainability technique (an acronym for SHapley Additive exPlanations). The SHAP values evaluate the impact of features in comparison to the prediction should the feature had some other baseline value. In other words, they allow decomposing prediction into feature importances.

We interpret spectrogram intensity values as SHAP values, and inspect their influence on model predictions. Figures 14, 15, 16, 17, and 18 show SHAP value analysis for a few spectrogram predictions. We intend to verify what information the model is using to make its decisions. In all figures, we see consistent SHAP colors across specific spectrogram frequency bands. This is evidence of sound model decisions, as it is using frequency information to perform its predictions, as expected. The domains may present distinct recording sensor frequency characteristics that make it possible for easy domain distinction.

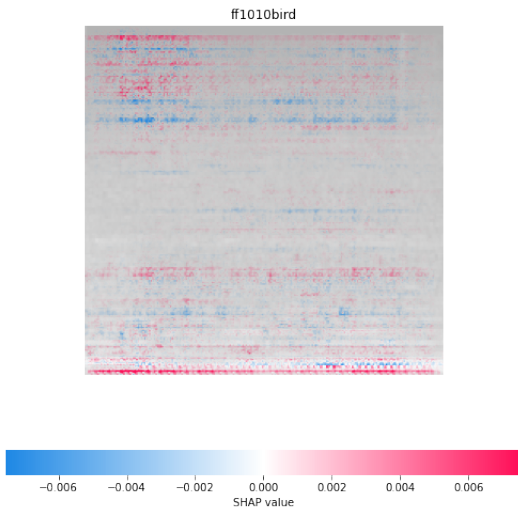


Figure 14. SHAP analysis of bird detection example 1. Source: Author.

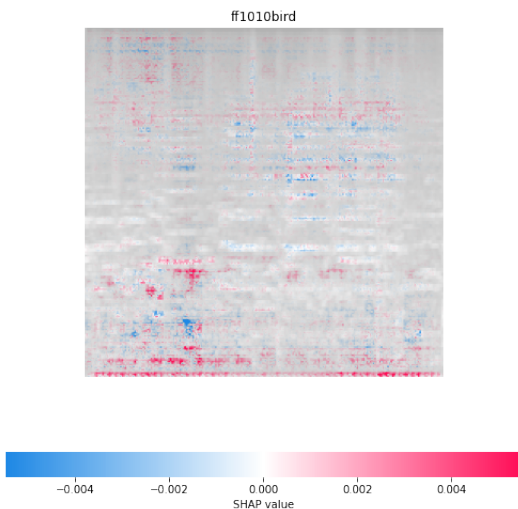


Figure 15. SHAP analysis of bird detection example 2. Source: Author.

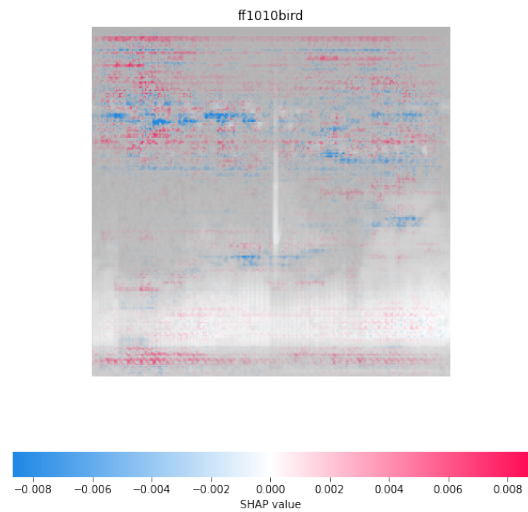


Figure 16. SHAP analysis of bird detection example 3. Source: Author.

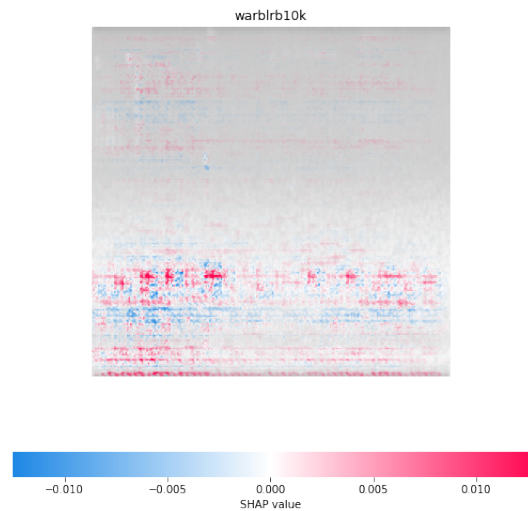


Figure 17. SHAP analysis of bird detection example 4. Source: Author.

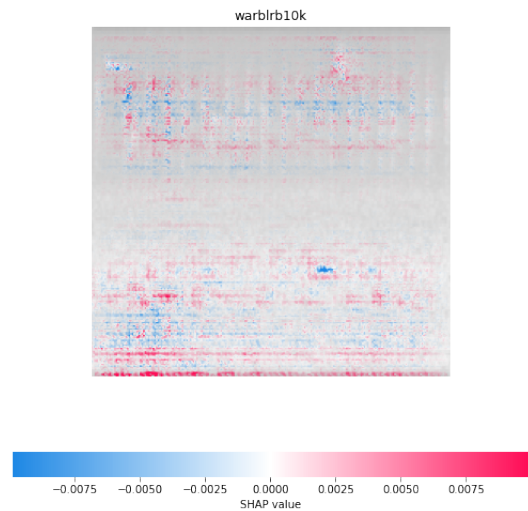


Figure 18. SHAP analysis of bird detection example 5. Source: Author.

B Complete Experiment F1-Score Tables

This appendix contains the expanded tables showing the complete experiment results (Tables 12, 13, 14, 15, 16).

Table 12. Experiment 1 — Bird Detection, Original Dataset Size, Micro F1-Score

Domain	Sequential	Inverse Sequential	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.368 ± 0.055	0.786 ± 0.010	0.623 ± 0.026	0.694 ± 0.038	0.803 ± 0.009	0.618 ± 0.033	0.799 ± 0.013
FF1010BIRD	0.791 ± 0.014	0.290 ± 0.030	0.631 ± 0.026	0.574 ± 0.030	0.779 ± 0.029	0.627 ± 0.036	0.779 ± 0.022
Average	0.580 ± 0.033	0.538 ± 0.016	0.627 ± 0.013	0.634 ± 0.015	0.791 ± 0.015	0.622 ± 0.019	0.789 ± 0.015

Table 13. Experiment 2 — Bird Detection, Reduced FF1010BIRD, Micro F1-Score

Domain	Sequential	Inverse Sequential	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.275 ± 0.047	0.777 ± 0.009	0.745 ± 0.037	0.641 ± 0.026	0.797 ± 0.018	0.752 ± 0.028	0.793 ± 0.031
FF1010BIRD	0.754 ± 0.011	0.285 ± 0.029	0.475 ± 0.051	0.605 ± 0.039	0.793 ± 0.015	0.512 ± 0.032	0.796 ± 0.012
Average	0.514 ± 0.028	0.531 ± 0.017	0.610 ± 0.019	0.623 ± 0.018	0.795 ± 0.013	0.632 ± 0.015	0.795 ± 0.019

Table 14. Experiment 3 — Bird Detection, Reduced WARBLRB10K, Micro F1-Score

Domain	Sequential	Inverse Sequential	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.258 ± 0.044	0.766 ± 0.006	0.520 ± 0.061	0.649 ± 0.031	0.757 ± 0.068	0.529 ± 0.047	0.768 ± 0.049
FF1010BIRD	0.751 ± 0.008	0.265 ± 0.007	0.745 ± 0.032	0.647 ± 0.044	0.771 ± 0.023	0.737 ± 0.042	0.773 ± 0.020
Average	0.504 ± 0.025	0.515 ± 0.006	0.632 ± 0.033	0.648 ± 0.024	0.764 ± 0.038	0.633 ± 0.025	0.771 ± 0.029

Table 15. Experiment 4 — Bird Detection, Reduced Symmetric, Micro F1-Score

Domain	Sequential	Inverse Sequential	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
WARBLRB10K	0.313 ± 0.176	0.761 ± 0.005	0.603 ± 0.221	0.538 ± 0.240	0.588 ± 0.245	0.569 ± 0.234	0.656 ± 0.207
FF1010BIRD	0.698 ± 0.148	0.279 ± 0.090	0.390 ± 0.182	0.429 ± 0.210	0.424 ± 0.231	0.442 ± 0.204	0.411 ± 0.221
Average	0.506 ± 0.080	0.52 ± 0.044	0.497 ± 0.06	0.483 ± 0.073	0.506 ± 0.092	0.505 ± 0.057	0.533 ± 0.095

Table 16. Experiment 5 — Speaker Identification, Original Dataset Size, Micro F1-Score

Domain	Sequential	Inverse Sequential	Stew	Balanced	Loss Sum	Random Sum	Loss Mean
CLEAN	0.049 ± 0.021	0.049 ± 0.021	0.132 ± 0.018	0.130 ± 0.012	0.136 ± 0.017	0.129 ± 0.019	0.123 ± 0.015
IPAD_BALCONY1	0.048 ± 0.019	0.048 ± 0.019	0.139 ± 0.015	0.139 ± 0.016	0.142 ± 0.018	0.133 ± 0.018	0.136 ± 0.016
IPAD_BEDROOM1	0.051 ± 0.022	0.051 ± 0.022	0.134 ± 0.014	0.130 ± 0.013	0.130 ± 0.015	0.125 ± 0.012	0.127 ± 0.012
IPAD_CONFROOM1	0.052 ± 0.021	0.052 ± 0.021	0.129 ± 0.015	0.129 ± 0.014	0.131 ± 0.013	0.125 ± 0.014	0.125 ± 0.015
IPAD_CONFROOM2	0.048 ± 0.019	0.048 ± 0.019	0.135 ± 0.018	0.132 ± 0.015	0.135 ± 0.017	0.128 ± 0.018	0.121 ± 0.020
IPADFLAT_CONFROOM1	0.050 ± 0.018	0.050 ± 0.018	0.140 ± 0.017	0.137 ± 0.010	0.140 ± 0.016	0.128 ± 0.016	0.133 ± 0.013
IPADFLAT_OFFICE1	0.052 ± 0.021	0.052 ± 0.021	0.135 ± 0.017	0.132 ± 0.012	0.132 ± 0.018	0.126 ± 0.017	0.125 ± 0.012
IPAD_LIVINGROOM1	0.048 ± 0.020	0.048 ± 0.020	0.135 ± 0.016	0.131 ± 0.011	0.131 ± 0.017	0.128 ± 0.014	0.126 ± 0.016
IPAD_OFFICE1	0.051 ± 0.019	0.051 ± 0.019	0.129 ± 0.013	0.127 ± 0.011	0.133 ± 0.014	0.123 ± 0.013	0.121 ± 0.013
IPAD_OFFICE2	0.048 ± 0.018	0.048 ± 0.018	0.137 ± 0.015	0.134 ± 0.011	0.134 ± 0.014	0.133 ± 0.016	0.130 ± 0.013
IPHONE_BALCONY1	0.048 ± 0.016	0.048 ± 0.016	0.141 ± 0.017	0.139 ± 0.017	0.146 ± 0.021	0.132 ± 0.019	0.139 ± 0.020
IPHONE_BEDROOM1	0.051 ± 0.020	0.051 ± 0.020	0.133 ± 0.016	0.130 ± 0.012	0.130 ± 0.016	0.126 ± 0.014	0.126 ± 0.013
IPHONE_LIVINGROOM1	0.046 ± 0.018	0.046 ± 0.018	0.126 ± 0.013	0.125 ± 0.007	0.125 ± 0.012	0.126 ± 0.016	0.123 ± 0.010
PRODUCED	0.048 ± 0.018	0.048 ± 0.018	0.134 ± 0.017	0.130 ± 0.009	0.139 ± 0.018	0.129 ± 0.019	0.128 ± 0.013
Average	0.049 ± 0.018	0.049 ± 0.018	0.134 ± 0.013	0.132 ± 0.010	0.134 ± 0.012	0.128 ± 0.014	0.127 ± 0.011

C Loss Curve Convergence Visualization

This appendix shows the loss curves during model training on the bird detection dataset, using its original size (Experiment 1).

The real training loss used in backpropagation is depicted in Figure 19. Because some methods operate on higher loss values, the comparison is difficult. We normalize these methods by dividing their loss values by the number of domains (Figure 20). The methods depict similar results towards the end, which may be difficult to visualize. For this reason, we provide Figure 21 showing a zoomed version of the plot including only the last 3 epochs.

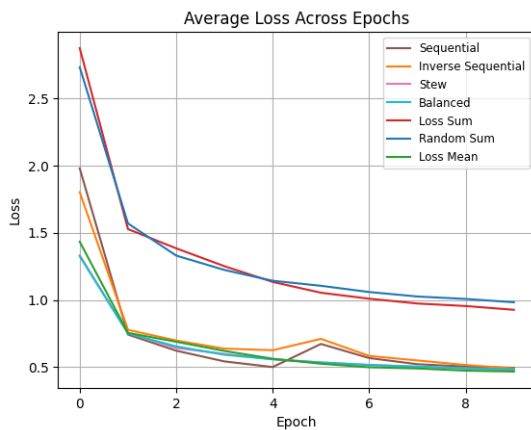


Figure 19. Average domain loss across epochs. Source: Author.

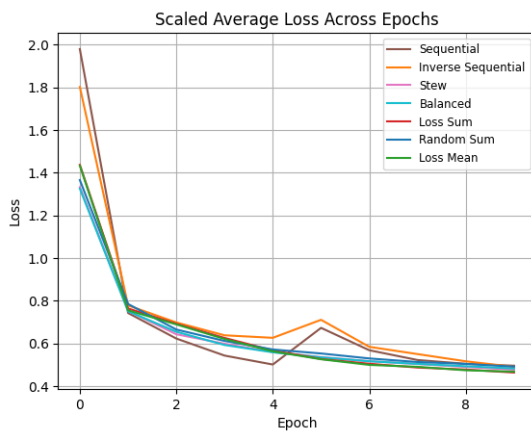


Figure 20. Average domain loss across epochs. Methods that operate on a higher loss were normalized for comparison purposes (this normalization consists of dividing the loss by the number of domains). Source: Author.

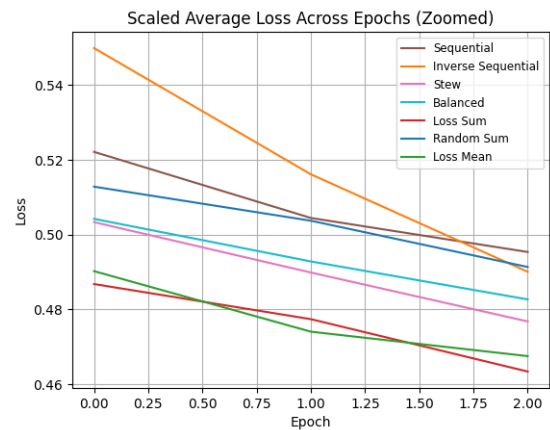


Figure 21. Average normalized domain loss for the last 3 epochs. Source: Author.