CURRICULUM DOMAIN GENERALIZATION FOR COMPUTER VISION

by
SORMEH SERPOOSH

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
July 2025

CURRICULUM DOMAIN GENERALIZATION FOR COMPUTER VISION

Approved by:

Assoc. Prof. ÖZNUR TAŞTAN .....
(Thesis Supervisor)

Assoc. Prof. HÜSEYİN ÖZKAN .....

Prof. PINAR DUYGULU ŞAHİN .....

Date of Approval: JULY 17, 2025

ABSTRACT

CURRICULUM DOMAIN GENERALIZATION FOR COMPUTER VISION

SORMEH SERPOOSH

COMPUTER SCIENCE AND ENGINEERING MSc. THESIS, July 2025

Thesis Supervisor: Assoc. Prof. ÖZNUR TAŞTAN
Co-advisor: Prof. ERCHAN APTOULA

Keywords: Domain generalization, Domain Shift, Curriculum Learning, Progressive Feature Alignment, Feature Remixing, Computer Vision

Domain generalization aims to train models to perform well on unseen domains without access to data from those domains during training. ADRMX (Additive Disentanglement of Domain Features with Remix Loss) is an augmentation based design to improve generalization to unseen domains. ADMRX disentangles domain-invariant and domain-specific features via an additive architecture and applies a latent-space remix loss, mixing same-class representations across source domains to generate synthetic samples. Building on the ADRMX method, which mixes feature representations of same-class samples across different domains, this thesis introduces Progressive Feature Alignment (PFA). PFA is a curriculum-driven remixing strategy. Remixing proceeds from the closest to the most distant pairs of domains, with mixing coefficients dynamically adjusted based on class centroid distances across domains to prevent unrealistic blending of dissimilar features and reduce noise in the resulting synthetic examples. By organizing feature remixing according to semantic proximity, PFA enables a gradual adaptation to increasingly challenging shifts.
Under the leave-one-domain-out protocol on the PACS and OfficeHome benchmarks, PFA consistently outperforms ADRMX and other state-of-the-art domain generalization techniques, yielding especially strong gains on the more challenging OfficeHome dataset. These results demonstrate that a curriculum-driven approach to feature remixing can substantially enhance the robustness of computer vision models to complex domain variation, suggesting new directions for tackling severe shifts in unseen data.

The implementation of my method is available at: https://github.com/SormehSerp/PFA_

ÖZET

BILGISAYARLA GÖRÜ IÇIN MÜFREDAT TABANLI ALAN GENELLEME

SORMEH SERPOOSH

BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, TEMMUZ 2025

Tez Danışmanı: Doç. Dr. ÖZNUR TAŞTAN
Tez Eş Danışmanı: Prof. Dr. ERCHAN APTOULA

Anahtar Kelimeler: Alan Genellemesi, Alan Kayması, Müfredat Öğrenmesi, Kademeli Özellik Hizalaması, Özellik Karıştırma, Bilgisayarla Görü

Alan genellemesi, modellerin eğitim sırasında erişimi olmayan, görülmemiş alanlarda da iyi performans göstermesini amaçlar. ADRMX (Remix Kayıplı Alan Özelliklerinin Toplamsal Ayrıştırılması), görülmemiş alanlara genelleme yeteneğini artırmak için veri artırmaya dayalı bir yaklaşımdır. ADRMX, alanlardan bağımsız ve alana özgü özellikleri toplamsal bir mimariyle ayrıştırır ve gizil uzayda remix kaybı uygular; böylece kaynak alanlar arasında aynı sınıfa ait temsilleri karıştırarak sentetik örnekler üretir. Farklı alanlarda aynı sınıfa ait örneklerin özellik temsillerini karıştıran ADRMX yönteminden yola çıkarak, bu tez Kademeli Özellik Hizalaması (PFA) yöntemini sunar. PFA, müfredat temelli bir remix stratejisidir. Remix işlemi, en yakın alan çiftlerinden başlayarak en uzaklara doğru ilerler ve karıştırma katsayıları, alanlar arası sınıf merkezleri arasındaki mesafeye göre dinamik olarak ayarlanır; bu sayede çok farklı özelliklerin gerçekçi olmayan şekilde karışması önlenir ve ortaya çıkan sentetik örneklerdeki gürültü azaltılır. Özellik karıştırmayı anlamsal yakınlığa göre düzenleyen PFA, giderek zorlaşan alan değişimlerine kademeli bir uyum sağlar.

PACS ve OfficeHome veri setlerinde bir-alanı-dışarı-bırak protokolü altında, PFA sürekli olarak ADRMX ve diğer güncel alan genelleme tekniklerinden daha iyi performans göstermekte, özellikle zorlu OfficeHome veri setinde belirgin kazanımlar elde etmektedir. Bu sonuçlar, özellik karıştırmada müfredat temelli bir yak-

laşımın, bilgisayarla görü modellerinin karmaşık alan değişimlerine karşı dayanıklılığını önemli ölçüde artırabileceğini göstermekte ve görülmemiş verilerdeki ciddi değişimlerle başa çıkmak için yeni yönler önermektedir. Yöntemimin uygulamasına şu adresten ulaşılabilir: https://github.com/SormehSerp/PFA_

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to everyone who has supported me throughout this challenging yet rewarding journey.

First and foremost, I want to thank my advisor, Dr. Öznur Taştan. Her guidance, patience, and encouragement have been invaluable throughout this thesis. She consistently offered thoughtful advice and provided me with the freedom and confidence to explore my ideas. Her support has been crucial, especially given the many challenges I faced along the way.

I am deeply grateful to Dr. Erchan Aptoula, whose insights and feedback have profoundly shaped the quality of my work. He was always available to answer my questions, no matter what. His keen eye for details, I might have overlooked, and his suggestions during our meetings often led me to new directions and significantly improved my results.

Finally, I want to thank The Scientific and Technological Research Council of Turkiye (TUBITAK) for the provided computational support under Contract 121E452.

This journey has not been without personal and external difficulties, particularly given the situation in my home country, Iran. Navigating the intense workload of my thesis while simultaneously preparing Ph.D. applications was, at times, overwhelming. I am thankful for the understanding and compassion shown by my advisors, who supported me both academically and personally through these hardships. Their encouragement motivated me to persevere and give my best effort, even during the most demanding moments.

I also wish to thank the members of my thesis committee for their time and willingness to be part of this process.

Thank you all for being part of this journey.

dedicated to the field

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Motivation and Background

Domain generalization (DG) has emerged as one of the most critical challenges in computer vision, particularly because real-world applications often face severe performance drops when models encounter data drawn from domains not seen during training (Long et al., 2015b). Classical DG methods try to learn domain-invariant features or rely on various data augmentation techniques to mimic domain shifts (Torralba & Efros, 2011). However, many of these approaches either introduce unrealistic synthetic examples or fail to capture the subtle relationships between domains, especially when domain shifts involve complex stylistic variations.

This thesis builds upon an earlier method called ADRMX (Additive Disentanglement of Domain Features with Remix Loss) proposed by Demirel et al. (2023), which proposes a strategy for improving domain generalization by synthesizing new feature representations through the remixing of features between samples of the same class but from different domains. In ADRMX, for each mini-batch, feature representations are first extracted. Then, pairs of samples belonging to the same class but different domains are randomly selected, and their feature vectors are linearly combined. The idea behind this process is to generate synthetic examples that interpolate between different domains, thereby simulating domain shifts. However, one fundamental limitation of ADRMX is that it does not account for the semantic or statistical closeness of the domains being mixed. The selection of domain pairs for remixing is purely random, without any measure of how similar or dissimilar the domains are in terms of feature distributions. As a result, there is a significant risk that the method may combine features from domains that are substantially different in distributional characteristics. Such mismatched combinations can produce synthetic feature vectors that are unrealistic or lie outside the actual feature space

manifold, introducing artifacts that confuse the learning process rather than improving generalization. This limitation is particularly problematic in datasets with substantial domain differences due to variations in artistic style and/or background complexity, making naive remixing insufficient or even harmful for robust domain generalization.

Motivated by these limitations, this work proposes a more structured approach for feature remixing. My hypothesis was that remixing features between domains that are inherently closer in feature space would produce more meaningful synthetic samples, while still providing enough variability to improve generalization. Furthermore, I aimed for this process to evolve gradually during training, exposing the model first to easier domain shifts and only later to more challenging ones, inspired by curriculum learning. This led me to develop the Progressive Feature Alignment (PFA) method, which integrates domain distance measurements directly into the remixing strategy, resulting in a more controlled approach to synthetic feature generation.

Ultimately, the motivation for this work stems from a desire to make domain generalization not just more effective, but also more more adaptive to the nuances of different domain relationships. By taking into account the relative difficulty of domain relationships and embedding that into the training process, I believe this work takes a step toward more intelligent learning strategies for robust machine learning.

1.2 Contributions

The main contribution of this thesis is the PFA method, which aims to improve domain generalization by guiding how features are remixed across domains during training. Instead of mixing features randomly, PFA uses the distances between domain-specific class centroids in the feature space to decide which domains should be mixed first, starting with those that are more similar and gradually including less similar pairs. This progression acts like a curriculum, helping the model adapt from easier to more difficult domain shifts. PFA also adjusts the mixing ratio based on how close the domains are, which helps avoid unrealistic synthetic features when domains differ too much. Through experiments, I show that PFA significantly improves accuracy compared to prior methods like ADRMX. I also analyze why some datasets present more challenges than some others, highlighting how domain differences can make naive remixing less effective unless guided by a structured approach

like PFA.

Together, these contributions advance the understanding of how controlled, distance-aware feature remixing can enhance domain generalization when trained on image datasets, offering a more principled alternative to purely random synthetic data generation.

The remainder of this thesis is organized as follows:
Chapter 2 presents related work, reviewing existing research on domain generalization, public datasets, evaluation metrics, and deep learning approaches relevant to this study. Chapter 3 describes the proposed methodology in detail, explaining the design and implementation of the PFA method. Chapter 4 discusses the experimental setup and reports the results obtained by evaluating PFA on benchmark datasets, comparing its performance to baseline and state-of-the-art methods. Chapter 5 provides an ablation study that analyzes the contributions of individual components within the PFA framework to overall performance. Finally, Chapter 6 concludes the thesis by summarizing key findings and suggesting directions for future research.

2. RELATED WORK

This chapter explores the background and key ideas that shape research in domain generalization, particularly within computer vision. It starts by introducing the public datasets that researchers commonly rely on, along with the metrics used to evaluate how well different methods cope with domain shifts. Next, it looks at major deep learning strategies for tackling domain generalization, focusing on approaches that either learn domain-invariant features or utilize data augmentation to improve robustness. Finally, the chapter discusses how these techniques are applied in the context of computer vision tasks, laying the groundwork for the methodology proposed later in this thesis.

2.1 Public Datasets for Domain Generalization Research

In my thesis, I work with two of the most widely used benchmarks: PACS and OfficeHome.

The PACS dataset (Li et al., 2017) includes images from four domains—Photo, Art Painting, Cartoon, and Sketch—all sharing the same seven object categories. What makes PACS particularly challenging is that the domains differ quite dramatically in visual style and texture, which is ideal for testing whether a model can learn domain-invariant features.

OfficeHome (Venkateswara et al., 2017), on the other hand, is a much larger and more complex benchmark. It consists of four domains as well—Art, Clipart, Product, and Real-World—with 65 object classes. Unlike PACS, which has fewer classes and more pronounced domain gaps, OfficeHome often presents more subtle shifts between domains. However, it also includes significant diversity in style, composition, and color palettes, making it a tougher test for domain generalization.

These datasets have become the standard for evaluating DG methods because they offer controlled but diverse scenarios that approximate real-world domain shifts. However, they are not without challenges. OfficeHome, in particular, has highly imbalanced classes and uneven domain sizes, which can complicate training and make fair comparisons challenging. Another difficulty is that visual differences across domains might be either too subtle or too extreme.

2.2 Evaluation Metrics and Validation Protocols

In domain generalization, the main performance metric is usually classification accuracy measured on a held-out domain. The standard evaluation protocol is leave-one-domain-out (LODO) validation, in which one domain is completely excluded during training and used solely for testing. This setup closely simulates real-world scenarios where a model encounters data from an unseen domain. The procedure is repeated for each domain in the dataset, ensuring that each domain serves once as the held-out test set. The resulting accuracies from each fold are then averaged to provide an overall measure of the model's generalization performance across domains.

In my work, I consistently follow LODO validation on both PACS and OfficeHome. This means that for each run, I train on three domains and test on the fourth. The reported result is either the accuracy on each individual held-out domain or the average accuracy across all domains, depending on the comparison.

While overall accuracy is the most commonly reported number, it is not always sufficient. Especially in datasets like OfficeHome with many classes and diverse domains, performance can vary significantly between domains. For this reason, I pay close attention to per-domain results to ensure that improvements are not due to outliers or single-domain effects.

Another important concern in DG research is avoiding leakage of target domain information during hyperparameter tuning. Using the test domain in any part of model selection can lead to overly optimistic results. Therefore, I carefully separate tuning processes from the held-out domain to ensure fair comparisons with prior work.

Overall, consistent validation protocols and transparent reporting are essential for showing that methods like PFA truly improve generalization rather than merely

fitting to the specifics of a given benchmark.

2.3 Model Architectures in Domain Generalization

Over the past decade, significant research has focused on developing model architectures that can learn representations robust to domain shifts. Unlike traditional supervised learning, DG requires models to perform well on unseen domains whose distributions differ from those seen during training. This challenge has motivated innovations in network design, feature disentanglement, and synthetic feature generation.

2.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have long been the backbone of visual recognition tasks and have been widely adopted in domain generalization research. Architectures like ResNet (He et al., 2016) are commonly used because their residual connections help stabilize training in deep networks, making them a strong foundation for DG pipelines. In DG, CNNs are often combined with additional mechanisms—such as adversarial training, domain classifiers, or feature normalization layers—to reduce reliance on domain-specific cues like texture or style.

However, pure CNN-based approaches can be limited in their ability to capture global dependencies, which sometimes leads to poor generalization across domains with significant style or structure changes. This has driven research toward architectures that combine CNNs with more sophisticated feature manipulation strategies.

2.3.2 Feature Disentanglement Models

A key line of research in DG has explored feature disentanglement, where models explicitly separate domain-invariant features (i.e., features tied to class semantics) from domain-specific features (e.g., styles, textures). The ADRMX framework (Demirel et al., 2023) is a notable example, introducing separate encoders to extract label-relevant and domain-specific features. By subtracting domain-specific components from the label features, ADRMX aims to isolate a domain-invariant representation, which is then used both for classification and feature-level augmentation.

This idea of generating synthetic samples by remixing domain-invariant and domain-specific features has influenced multiple DG methods, enabling models to simulate domain shifts without relying on target domain data.

2.3.3 Transformers in Domain Generalization

More recently, transformer-based models have started to make an impact in domain generalization tasks. Vision Transformers (ViTs) (Dosovitskiy et al., 2021) process images as sequences of patches, capturing global context more effectively than traditional CNNs. Their self-attention mechanism allows transformers to model long-range dependencies, which can be crucial for recognizing object shapes and semantics across diverse domains.

Several works have combined transformers with domain adaptation or domain generalization strategies. For example, some approaches integrate attention maps into domain-invariance pipelines, using transformers to highlight semantically important regions while suppressing domain-specific noise. However, transformers often require larger datasets and careful regularization to avoid overfitting, especially when applied to domain shifts.

2.3.4 Feature Augmentation and Remixing Architectures

Another active area in DG research involves architectures designed for feature-level augmentation. Instead of applying transformations directly to pixel data, these

methods manipulate feature representations in the latent space to generate synthetic samples that simulate domain shifts.

For instance, Demirel et al. (2023) introduced a feature-level remixing strategy, where domain-invariant features from one sample are combined with domain-specific features from another sample within the same class. This synthetic feature construction exposes the model to novel domain combinations during training, helping improve robustness to unseen domains.

Other works explore frequency-domain mixing (Yang & Soatto, 2020) or statistical feature alignment, further diversifying the set of possible domain shifts the model experiences during training. These architectures often build on classical backbones (like ResNet or transformers) but introduce additional modules specifically for synthetic feature generation.

2.3.5 Contrastive Learning and Feature Alignment

Contrastive learning (Ganin et al., 2016) has recently emerged as an important tool for DG. It encourages samples from the same class—but different domains—to cluster together in feature space, while pushing apart samples from different classes. This aligns well with the goals of DG, where the challenge is to learn class-consistent representations that are insensitive to domain-specific variations.

Several DG models (Khosla et al., 2020; Zhou et al., 2023; Hu et al., 2023; Balaji et al., 2018) integrate supervised contrastive loss into their architectures, either as a standalone objective or in combination with feature remixing and adversarial training. Such methods have demonstrated significant improvements, especially in datasets like PACS and OfficeHome, where style and appearance variations between domains can be substantial.

2.3.6 Summary

In summary, the field of domain generalization has witnessed a steady evolution from purely CNN-based models toward hybrid architectures that combine feature disentanglement, transformers, and feature-level augmentation. Techniques like ADRMX

have shown the value of generating synthetic features by mixing domain-specific and domain-invariant representations. Meanwhile, transformers and contrastive learning bring powerful tools for capturing global structure and aligning features across domains. These innovations collectively form the backdrop against which my own method is developed.

2.4 Deep Learning Approaches for Domain Generalization

DG has emerged as a significant area of research in computer vision and machine learning, motivated by the need to train models that can perform robustly on unseen domains whose distributions differ from those observed during training. Unlike domain adaptation, which assumes at least some access to target domain data, DG requires that models be trained entirely on source domains, making it substantially more challenging. This chapter reviews the landscape of deep learning approaches for DG, emphasizing methods relevant to computer vision, and provides context for the developments that motivate this thesis.

2.4.1 Learning Domain-Invariant Representations

One of the foundational goals in DG research is to learn representations that remain consistent across varying domains. Early approaches aimed at aligning global feature distributions between source domains to mitigate domain discrepancies. Techniques like domain adversarial training (Ganin et al., 2016) encourage the model to extract features indistinguishable across domains by employing adversarial losses. Similarly, Maximum Mean Discrepancy (MMD) minimization (Long et al., 2015a) and CORAL (Sun & Saenko, 2016a) focus on aligning statistical properties, such as feature covariances.

Later, Li et al. (2018) proposed conditional invariant representation learning, which extends these ideas by considering class-conditioned distributions, ensuring that domain-invariant features also preserve class semantics.

2.4.2 Data Augmentation and Synthetic Feature Generation

An increasingly prominent direction in DG is to create synthetic examples that simulate domain shifts, improving the model's robustness to unseen domains. MixUp (Zhang et al., 2018) is a widely known technique that linearly interpolates between pairs of training examples and their labels. While originally proposed for regularization, MixUp's interpolation introduces variations that can mimic domain shifts.

MixStyle (Zhou et al., 2021) represents a significant evolution, perturbing feature-level statistics (such as mean and variance) to produce new styles during training. This method has shown remarkable success in visual domain generalization benchmarks like PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017).

Yang & Soatto (2020) proposed Fourier Domain Adaptation (FDA), a frequency-domain approach that replaces low-frequency components between images, generating synthetic views that preserve object structure while simulating different domain appearances. Such methods are particularly effective for vision tasks where style differences between domains can be stark, as in the PACS (Li et al., 2017) dataset.

While synthetic augmentation has proven effective, challenges remain. Poorly controlled mixing can generate unrealistic or semantically inconsistent samples, underscoring the importance of careful design in augmentation strategies. (Zhang et al., 2018; Shi et al., 2023; Yun et al., 2019; Geirhos et al., 2019)

2.4.3 Feature Disentanglement and Remixing

A separate but related strategy is feature disentanglement, where models explicitly separate domain-invariant content from domain-specific style features. For instance, ADRMX (Demirel et al., 2023) employs dual encoders: one focused on extracting label-related features and another capturing domain-specific variations. By subtracting domain information from label representations, ADRMX generates semantically meaningful and domain-agnostic features.

Beyond disentanglement, ADRMX incorporates feature remixing, synthesizing new samples by blending features across domains within the same class. This synthetic data acts as a bridge, helping the model learn representations robust to domain variations. Similar ideas appear in MixStyle, which also aims to improve generalization by mixing information across domains, but instead of explicitly encoding

domain and label features as ADRMX does, MixStyle (Zhou et al., 2021) operates by perturbing feature statistics, such as mean and variance, to simulate style shifts during training.

2.4.4 Contrastive and Self-Supervised Learning in Domain Generalization

Contrastive learning has emerged as a key pillar in DG, driven by its ability to encourage clustering of semantically similar samples across domains. Supervised contrastive learning (Khosla et al., 2020) enforces proximity between samples of the same class while pushing apart samples from other classes. In DG, this approach helps unify representations from different domains, promoting domain-invariant feature learning.

Domain Contrastive Learning (Zhou et al., 2023) extends this by explicitly constructing positive pairs from different domains, leveraging contrastive objectives to extract domain-invariant signals. Memory banks and momentum encoders (He et al., 2020) further stabilize contrastive signals during training, enhancing generalization.

2.4.5 Transformers and Attention Mechanisms

Transformer-based architectures (Dosovitskiy et al., 2021) have recently become prominent in DG research, particularly because of their capacity to model global context and long-range dependencies. Unlike CNNs, transformers process images as sequences of patches, allowing them to capture relationships across distant spatial regions. This makes them particularly useful for detecting consistent object structures across domains.

Hybrid models like DeiT (Touvron et al., 2021) combine convolutional feature extraction with transformer-based reasoning, achieving impressive results in various vision tasks. Additionally, attention maps in transformers offer the potential to isolate domain-invariant regions, contributing another layer of robustness.

Nevertheless, transformers bring challenges, including significant computational costs and sensitivity to training data scale and diversity.

2.4.6 Curriculum Learning Strategies

Curriculum learning (CL) has been adapted to DG as a means to progressively expose the model to increasing levels of difficulty. Rather than relying on random or uniform sampling, curriculum methods arrange training data from "easy" to "hard" based on metrics such as feature distance, domain gap, or entropy.

Several recent innovations exemplify this approach. Wang et al. (2024) proposed Ladder Curriculum Learning (LCL), which sorts data both at inter-domain and intra-domain levels, allowing for smoother progression through increasingly complex samples. Furthermore, Wang et al. (2024) introduced Curriculum Learning-based Domain Generalization (CLDG), which leverages frequent classes to help the model learn rarer ones, effectively addressing category imbalance. Additionally, Jiang et al. (2023) developed the Momentum Difficulty Framework (MoDify), which dynamically balances sample difficulty with the model's evolving competence during training.

These curriculum methods aim to mitigate sudden learning shifts and enhance convergence, especially in highly heterogeneous domains of datasets.

While these methods share the overall principle of progressing from easier to harder examples, my proposed PFA method differs in several key aspects. For example, Ladder Curriculum Learning (LCL) (Wang et al., 2024) operates primarily at the sample level, sorting individual instances both within and across domains based on learned difficulty scores. In contrast, PFA works at the domain-pair level, using centroid distances computed for each class to quantify how similar or different domains are. Instead of directly ranking samples, PFA sorts domain pairs according to these statistical distances and progressively introduces more distant (thus more challenging) domain pairs during training. Moreover, PFA integrates a dynamic remixing mechanism where the mixing ratio between features from two domains depends on their normalized centroid distance, whereas LCL does not explicitly perform feature-level remixing. This makes PFA distinct in targeting feature-level domain alignment via a curriculum that respects domain similarities, rather than solely focusing on sample-level difficulty.

2.4.7 Meta-Learning for Domain Generalization

Meta-learning has emerged as another powerful strategy for DG. It simulates domain shifts during training by dividing source domains into meta-train and meta-test

splits. The goal is to optimize models not merely for the training data but for rapid adaptation to unseen distributions.

Methods like MLDG (Gulrajani & Lopez-Paz, 2020) and MetaReg (Balaji et al., 2018) learn regularizers and model parameters that generalize effectively beyond the observed source domains. These approaches have demonstrated significant improvements in domain generalization performance, though they often demand increased computational resources.

2.5 Domain Generalization in Computer Vision

Domain generalization in computer vision encompasses a diverse range of tasks, including object recognition, scene classification, and medical imaging, where domain shifts are frequent and sometimes severe. Datasets such as PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017) have become benchmarks for testing DG algorithms, providing domains with distinct styles, such as photo, art painting, cartoon, and sketch.

2.5.1 Architectural Foundations in Vision DG

CNNs, particularly architectures like ResNet (He et al., 2016), remain a backbone for many DG pipelines due to their stability and strong feature extraction capabilities. Yet, CNNs often capture superficial domain-specific patterns, such as texture, rather than more robust shape-based cues.

This limitation has driven research into alternative architectures, including transformers, which excel in capturing global context. The integration of attention mechanisms into DG architectures has allowed for dynamic focus on domain-invariant regions, enhancing robustness.

2.5.2 Hybrid Approaches and Emerging Trends

Recent research has increasingly embraced hybrid models, combining CNNs for local detail extraction with transformers for global reasoning. This hybridization seeks to balance computational efficiency with representational flexibility.

Furthermore, frequency-domain augmentations and feature remixing techniques remain central to DG research, offering methods to simulate domain shifts in a controlled manner.

2.6 Summary

In summary, domain generalization in computer vision has evolved into a sophisticated research field blending classical ideas like domain-invariant learning with modern innovations such as transformers, contrastive learning, and curriculum strategies. Despite significant advancements, many challenges remain, especially concerning scalability and robustness across diverse datasets like PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017). The insights from these diverse strategies form the basis for the methods proposed in this thesis.

Emerging research explores combining domain generalization with large-scale pretraining and foundation models. Works like CLIP (Radford et al., 2021) and DINOv2 (Oquab et al., 2023) suggest that models pre-trained on diverse and large datasets inherently learn features transferable across domains. Adapter-based strategies (Lee et al., 2024) allow lightweight fine-tuning for DG without catastrophic forgetting. Meanwhile, multimodal settings such as Visual Question Answering (VQA), face new challenges due to domain shifts in both vision and language. Datasets like VQA-GEN have been proposed to benchmark these multimodal DG tasks (Unni et al., 2023).

Domain generalization remains a rapidly advancing field, spanning sophisticated data augmentation, deep representation learning, and curriculum-inspired frameworks. The synergy of these methods pushes the limits of robust AI systems capable of handling real-world distribution shifts.

3. METHODOLOGY

This chapter describes the framework and methodology I developed over the course of my research. It follows the gradual development of my work, starting from the ADRMX framework and understanding its limitations, to designing my PFA method.

The detailed steps of the proposed PFA method are described in Algorithm 1. Here, $\mathbf{f}_c^{d_1}$ and $\mathbf{f}_c^{d_2}$ denote the feature vectors sampled from class c in domains d_1 and d_2, respectively.

Algorithm 1 Progressive Feature Alignment (PFA)

Require: Labeled data from multiple source domains $\{\mathcal{D}_d\}_{d=1}^{D}$, number of classes C, curriculum schedule $\mathcal{S}$, remix ratio range $[\alpha_{\min}, \alpha_{\max}]$, , training steps T (5000 steps per fold in practice)

Compute class centroids $\boldsymbol{\mu}_{c,d}$ for each domain d and class c
for each training step $t = 1$ to T **do**
  Determine allowed domain pairs based on curriculum schedule $\mathcal{S}(t)$
  for each allowed domain pair (d_1, d_2) **do**
    for each class $c = 1$ to C **do**
      Compute Euclidean distance $D_c(d_1, d_2) = \|\boldsymbol{\mu}_{c,d_1} - \boldsymbol{\mu}_{c,d_2}\|_2$
      Compute remix ratio α via Eq. (3.10) where δ controls deviation from equal mixing.
      Generate synthetic feature:

$$\mathbf{f}_{\text{syn}} = \alpha \cdot \mathbf{f}_c^{d_1} + (1 - \alpha) \cdot \mathbf{f}_c^{d_2}$$

      Add $\mathbf{f}_{\text{syn}}$ to training batch
    end for
  end for
  Update model parameters using the batch (including synthetic samples)
end for

3.1 Additive Disentanglement of Domain Features with Remix Loss

(ADMRX)

My work builds upon ADRMX (Demirel et al., 2023), which is a feature-level data augmentation strategy for domain generalization. ADRMX uses ResNet-50 (He et al., 2016) as its backbone. I will briefly review the key ideas of ADRMX before introducing my own method.

In ADRMX, two separate feature encoders are trained:
- A **label encoder** $\mathcal{F}_{\text{label}}$ to extract class-specific features.
- A **domain encoder** $\mathcal{F}_{\text{domain}}$ to extract domain-specific features.

Given an input image $\mathbf{x}$, these encoders produce:

$$\mathbf{z}_{\text{label}} = \mathcal{F}_{\text{label}}(\mathbf{x}) \tag{3.1}$$

$$\mathbf{z}_{\text{domain}} = \mathcal{F}_{\text{domain}}(\mathbf{x}) \tag{3.2}$$

From these, ADRMX defines the domain-invariant feature as:

$$\mathbf{z}_{\text{inv}} = \mathbf{z}_{\text{label}} - \mathbf{z}_{\text{domain}} \tag{3.3}$$

To perform feature-level augmentation, ADRMX takes:
$\mathbf{z}_{\text{inv}}^{(i)}$ from one sample i, and $\mathbf{z}_{\text{domain}}^{(j)}$ from another sample j of the same class but a different domain and sums them:

$$\mathbf{z}_{\text{remix}} = \mathbf{z}_{\text{inv}}^{(i)} + \mathbf{z}_{\text{domain}}^{(j)} \tag{3.4}$$

The synthetic feature $\mathbf{z}_{\text{remix}}$ is then used as a new training example.

While ADRMX demonstrates good performance on simpler datasets such as PACS (Li et al., 2017), it struggles on more complex datasets like OfficeHome (Venkateswara et al., 2017), where domain discrepancies are bolder. Experiments show that ADRMX achieves an average accuracy of only 68.3% on OfficeHome, com-

pared to 75.2% achieved by the proposed PFA method. This performance gap can be due to the significant variations in representation across domains in OfficeHome. For example, the Clipart domain often contains simplified drawings with flat colors, while the Real-World domain consists of high-resolution photos with complex textures and lighting. Mixing features from such dissimilar domains may produce synthetic examples that lie outside the true data manifold, thereby degrading model performance. This observation is consistent with prior analyses that emphasize the challenge of domain shifts involving large stylistic gaps.

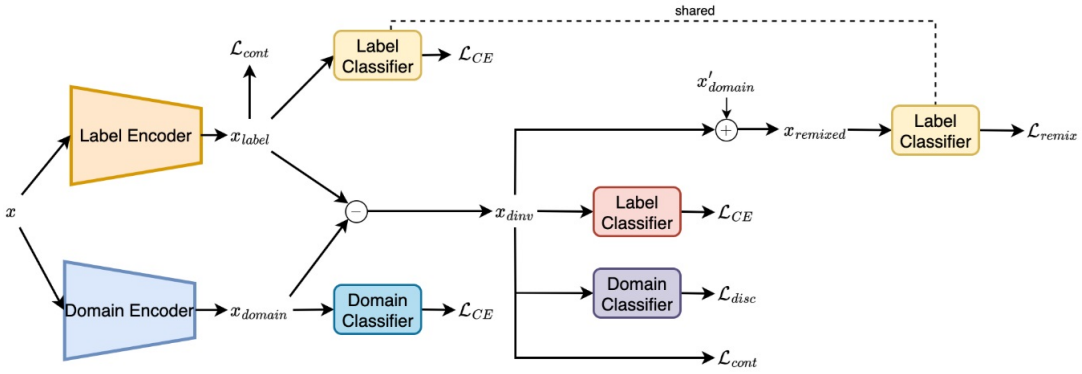

Figure 3.1 Architecture of the ADRMX model (Demirel et al., 2023). The framework extracts features from input images using a shared backbone, separates domain-specific and class-specific representations through dual branches, and generates synthetic training examples by linearly mixing feature vectors of the same class sampled from different domains. This remixing strategy aims to simulate domain shifts to enhance generalization to unseen target domains.

3.2 Progressive Feature Alignment (PFA)

3.2.1 Motivation and Overview

My intuition was that not all domain pairs should be mixed equally. Mixing features from highly similar domains is less risky than mixing those from domains with large discrepancies. Thus, I wanted to define a curriculum where remixing starts with "easy" domain pairs (small gaps) and gradually progresses to "harder" pairs (large gaps). This idea is inspired by curriculum learning (Bengio et al., 2009).

Overall, my approach introduces two main innovations:

- A quantitative measure of domain distance
- Dynamic weighting of the remixing strength based on the pairwise domain distance

3.2.2 Centroid-Based Domain Distance and Curriculum Schedule

For each class c and each domain d, I compute a centroid vector:

$$\boldsymbol{\mu}_{c,d} = \frac{1}{N_{c,d}} \sum_{i=1}^{N_{c,d}} \mathcal{F}_{\text{domain}}\big(\mathbf{x}_{i,d}\big) \tag{3.5}$$

where $\mathbf{x}_i$ belongs to class c in domain d, and $N_{c,d}$ is the number of such samples.

Given two domains d_1 and d_2, I compute the pairwise distance for class c as:

$$D_c(d_1, d_2) = \left\| \boldsymbol{\mu}_{c,d_1} - \boldsymbol{\mu}_{c,d_2} \right\|_2 \tag{3.6}$$

To normalize across classes and datasets, I divide by the intra-class standard deviation:

$$D_c^{\text{norm}}(d_1, d_2) = \frac{D_c(d_1, d_2)}{\sigma_c} \tag{3.7}$$

where:

$$\sigma_c = \sqrt{\frac{1}{M} \sum_{d} \sum_{i=1}^{N_{c,d}} \left\| \mathcal{F}_{\text{domain}}(\mathbf{x}_{i,d}) - \boldsymbol{\mu}_{c,d} \right\|_2^2} \tag{3.8}$$

Here, $M = \sum_d N_{c,d}$ is the total number of samples from class c across all domains.

Then, for each class c, I sort the domain pairs by D_c^{norm}. Denote the sorted list as:

$$\mathcal{P}_c = \{(d_1, d_2)\}_{\text{sorted by } D_c^{\text{norm}}} \tag{3.9}$$

Initially, remixing only uses the domain pairs with the smallest distances. As training progresses, I gradually introduce pairs with larger D_c^{norm}.

My curriculum schedule is governed by several hyperparameters that define how domain pairs are selected and how remixing is applied during training. Thresholds τ_1 and τ_2 determine which domain pairs are included at each stage of training, while the threshold τ controls how sharply the remix ratio α responds to differences in domain distance. The remix deviation parameter δ influences how far α can shift away from equal mixing, and the update frequency parameter `pair_update_freq` specifies how often centroid distances and domain pair selections are recomputed as training progresses.

The curriculum function $\mathcal{S}(t)$ determines which domain pairs are eligible for remixing at each step t. It operates based on two training thresholds, $T_1 = 1500$ and $T_2 = 3500$, which divide the training schedule into three progressive stages:
Steps 0–T_1 only for pairs with $D_c^{\text{norm}} \leq \tau_1$; steps T_1–T_2 include pairs up to τ_2, and finally from step T_2 onwards, all domain pairs will be included.

The specific values used for these hyperparameters are detailed in Section 4.2.3.

3.2.3 Dynamic Feature Remixing Strategy

Instead of mixing features equally, I introduce a dynamic weighting factor:

$$\alpha_{c,d_1,d_2} = \begin{cases} 0.5 + \delta \left(1 - \dfrac{D_c^{\text{norm}}(d_1, d_2)}{\tau}\right), & \text{if } d_2 > d_1 \\ 0.5 - \delta \left(1 - \dfrac{D_c^{\text{norm}}(d_1, d_2)}{\tau}\right), & \text{if } d_2 < d_1 \end{cases} \tag{3.10}$$

where:

- $\delta \in [0, 0.5]$ controls how far α moves away from 0.5.
- $D_c^{\text{norm}}(d_1, d_2)$ is the normalized centroid distance between the same class across domains d_1 and d_2.
- τ is a threshold hyperparameter determining how quickly α departs from 0.5 as domain distance increases.

A closer examination of Eq. (3.10) reveals how the hyperparameters δ and τ jointly influence the dynamics of remixing strength. More particularly, δ determines how far the remix ratio α_{c,d_1,d_2} can deviate from an even 0.5 split, thereby setting the maximum extent of feature mixing. Larger δ values allow stronger remixing toward one domain, while smaller values keep the mixing more balanced. Meanwhile, τ acts as a scaling threshold for domain distance: smaller values of τ make α_{c,d_1,d_2} more sensitive to even small differences in D_c^{norm}, causing remix ratios to shift rapidly as domain pairs become less similar. Together, these parameters enable PFA to progressively adjust remixing strength in a controlled manner, ensuring that synthetic features remain realistic and semantically meaningful as the curriculum advances.

Given two features $\mathbf{z}_{\text{inv}}^{(i)}$ from domain d_1 and $\mathbf{z}_{\text{domain}}^{(j)}$ from domain d_2, the remixed feature becomes:

$$\mathbf{z}_{\text{remix}} = \alpha_{c,d_1,d_2} \cdot \mathbf{z}_{\text{domain}}^{(j)} + \left(1 - \alpha_{c,d_1,d_2}\right) \cdot \mathbf{z}_{\text{inv}}^{(i)}. \tag{3.11}$$

This ensures that when domains are close ($D_c^{\text{norm}} \approx 0$), α approaches either 1 or 0, leading to strong remixing from one domain into another. When domains are far apart, α is pushed closer to either extreme (0 or 1) and avoids being near 0.5, preventing unrealistic half-and-half synthetic samples.

3.2.4 Centroid Distance and Remix Implementation

The first step in the proposed PFA method involves computing **class-specific centroids** for each domain, providing a compact representation of each class in the feature space.

Given a mini-batch of data, let each sample be denoted as a tuple (f_i, y_i, d_i), where:

- $f_i \in \mathbb{R}^F$ is the feature vector of sample i
- $y_i \in \{1, \ldots, K\}$ is the class label
- $d_i \in \{1, \ldots, D\}$ indicates the domain

For each domain d and class c, we define the the centroid vector $\boldsymbol{\mu}_{c,d}$ as the mean of all feature vectors belonging to class c in domain d. Formally,

$$\boldsymbol{\mu}_{c,d} = \frac{1}{N_{c,d}} \sum_{\substack{i=1 \\ y_i=c,\, d_i=d}}^{N} \mathbf{f}_i \tag{3.12}$$

where:

- $N_{c,d}$ is the number of samples of class c in domain d within the current mini-batch
- N is the total number of samples in the mini-batch

The centroid vector $\boldsymbol{\mu}_{c,d}$ captures the feature-space representation of class c specific to domain d. This step is crucial in my method, as it allows the model to analyze inter-domain differences for each class and later guides the curriculum strategy.

After obtaining the class-specific centroids for each domain, the next stage in the proposed PFA method is computing the pairwise Euclidean distances between centroids of the same class across different domains.

Let the centroids of class c in domains d_1, d_2, d_3 be denoted as $\boldsymbol{\mu}_{c,d_1}$, $\boldsymbol{\mu}_{c,d_2}$, and $\boldsymbol{\mu}_{c,d_3}$. For each class c, we compute:

$$d_{12}^c = \left\| \boldsymbol{\mu}_{c,d_1} - \boldsymbol{\mu}_{c,d_2} \right\|_2 \tag{3.13}$$

$$d_{13}^c = \left\| \boldsymbol{\mu}_{c,d_1} - \boldsymbol{\mu}_{c,d_3} \right\|_2 \tag{3.14}$$

$$d_{23}^c = \left\| \boldsymbol{\mu}_{c,d_2} - \boldsymbol{\mu}_{c,d_3} \right\|_2 \tag{3.15}$$

TThis process generates three distances per class, leading to $3 \times K$ distances in total for K classes. For our PACS and Office-Home datasets, $K = 7$. These distances represent the inter-domain variability for each class. To enable a curriculum-driven

learning approach, I sort the distances in an ascending order for each class, identifying which class-domain pairs are most similar and should be prioritized during earlier training stages. This forms the core idea of the progressive curriculum applied in PFA.

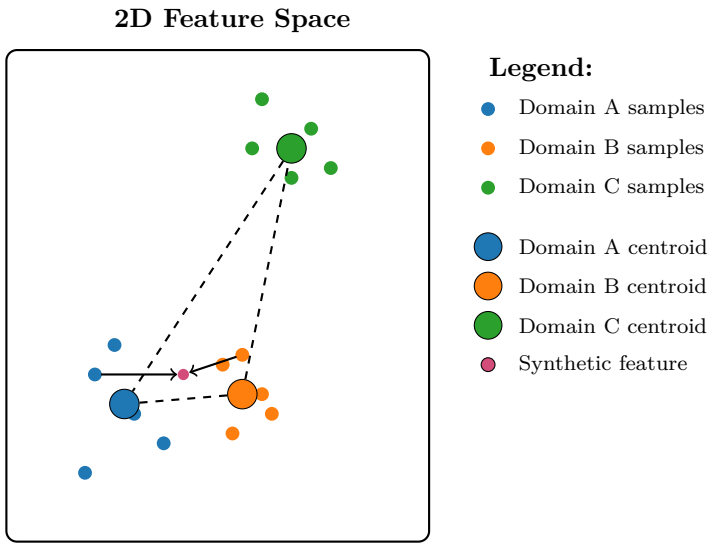

Figure 3.2 illustrates centroid-based distance computation and feature remixing in PFA. Dots represent samples of a single class from three domains (A: blue, B: orange, C: green), with solid-colored markers indicating their corresponding centroids. Dashed lines represent centroid-to-centroid distances used to sort domain pairs by inter-domain similarity. In this example, one edge of the triangle is significantly shorter than the others, indicating that the two connected domains are semantically closer in feature space. As shown, the synthetic sample (purple dot) is generated by remixing these two closer domains, supporting curriculum-driven training by prioritizing easier domain pairs in earlier stages. The same computation is performed independently for all classes in the dataset.

While Figure 3.2 illustrates the general concept of centroid distances among three domains, the specific domain pairs used for remixing are determined separately for each LODO fold as described below.

Because my experiments follow a leave-one-domain-out (LODO) protocol, each training run uses only three of the four available domains. Consequently, in each fold, only those three domains are included in centroid computations and distance calculations. It would be inconsistent to compute domain distances globally across all four domains, because the left-out domain has no available features during training and does not contribute to the remixing process in that fold. Therefore, I compute normalized centroid distances separately for each fold, ensuring that the curriculum schedule and remixing strategies are fully aligned with the actual domains seen during training.

For each fold, there are exactly three domain pairs. Table 3.1 provides an example of the average normalized centroid distances computed across folds on the OfficeHome dataset. These values illustrate how domain pairs are classified as "easy" or "hard" for remixing in the curriculum. A threshold of $\tau = 0.7$ is used to distinguish domain pairs suitable for early-stage remixing from those introduced later in training.

Table 3.1 Normalized centroid distances between domain pairs computed separately for each leave-one-domain-out fold on the OfficeHome dataset. Distances below the threshold $\tau = 0.7$ are considered "easy" and included in early remixing, while higher distances are progressively introduced in later stages.

Left-out Domain	**Domain Pair**	**Normalized Distance**
Art	Clipart–Product	0.55
Art	Clipart–Real	0.72
Art	Product–Real	0.61
Clipart	Art–Product	0.48
Clipart	Art–Real	0.65
Clipart	Product–Real	0.58
Product	Art–Clipart	0.68
Product	Art–Real	0.79
Product	Clipart–Real	0.82
Real	Art–Clipart	0.52
Real	Art–Product	0.57
Real	Clipart–Product	0.71

3.2.5 Model Architecture

My proposed model extends the ADRMX baseline (Demirel et al., 2023), which builds on a dual-feature architecture to disentangle domain-specific and class-specific features. A ResNet-50 backbone (He et al., 2016) is employed for feature extraction, initialized with ImageNet-pretrained weights. PFA further introduces dynamic remixing based on class-wise centroids, promoting the synthesis of new feature prototypes that bridge domain gaps.

The remixing process leverages a distance-based weighting mechanism, modulating remix intensity via the parameter τ, that will be explained in detail later in Experiments and Results chapter. In addition, supervised contrastive losses (Khosla et al., 2020) are applied both before and after remixing, enhancing intra-class compactness and inter-class separability.

3.3 Summary

In summary, my PFA method extends ADRMX by introducing a progressive curriculum over domain pairs and dynamic remix weighting. By quantifying domain similarity and adjusting remixing accordingly, PFA improves stability and domain generalization, especially for datasets with large intra-class and inter-domain variation like OfficeHome.

4. EXPERIMENTS & RESULTS

This chapter describes the experimental evaluation of the proposed PFA method. It introduces the datasets used, outlines the experimental setup, presents the results compared to existing methods, and discusses the implications of these findings.

4.1 Datasets

This thesis utilizes two widely adopted benchmark datasets for domain generalization in computer vision: PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017). Both datasets provide diverse domain shifts, enabling rigorous testing of model generalization performance.

Table 4.1 Statistics of PACS and OfficeHome datasets used in this thesis.

Dataset	**# Domains**	**# Classes**	**# Images**	**Description**
PACS	4	7	9,991	Photo, Art Painting, Cartoon, Sketch
OfficeHome	4	65	15,588	Art, Clipart, Product, Real-World

4.1.1 PACS

The PACS dataset (Li et al., 2017) comprises images from four distinct domains: Photo, Art Painting, Cartoon, and Sketch. It contains seven semantic classes: dog, elephant, giraffe, guitar, horse, house, and person. Each domain differs significantly

in terms of style and texture, as illustrated in Figure 4.1. This stylistic variability makes PACS an excellent testbed for algorithms aiming to disentangle semantic information from domain-specific appearance.

Figure 4.1 Example images from the PACS dataset (Li et al., 2017) across four domains. From left to right: Photo, Art Painting, Cartoon, and Sketch.

4.1.2 OfficeHome

The OfficeHome dataset (Venkateswara et al., 2017) contains four domains: Art, Clipart, Product, and Real-World. It includes 65 object categories that cover a diverse range of office-related and household items. Compared to PACS, OfficeHome presents significantly more challenging shifts in style, as well as a notable class imbalance. Figure 4.2 provides visual samples illustrating domain diversity.

For experiments, the leave-one-domain-out strategy is adopted. Each run involves training on three domains and testing on the fourth, ensuring the evaluation reflects unseen domain conditions.

Figure 4.2 Example images from the OfficeHome dataset (Venkateswara et al., 2017) across four domains. From left to right: Art, Clipart, Product, and Real-World.

4.2 Experiments

This chapter presents an extensive experimental evaluation of the proposed PFA method for domain generalization. Experiments were conducted on two widely adopted benchmark datasets: PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017). These datasets pose significant challenges due to diverse domain shifts arising from variations in style, content, and image statistics. The goal is to assess the effectiveness of PFA in improving generalization performance under out-of-distribution conditions.

4.2.1 Implementation Details

All experiments were conducted using a ResNet-50 backbone (He et al., 2016) implemented in PyTorch. The experiments were conducted on a server equipped with two NVIDIA RTX 3090 GPUs, each with 24GB of VRAM. Models were trained with a batch size of 16. Optimization used the Adam optimizer with an initial learning rate of $1e^{-4}$, $\beta_1 = 0.9$, and weight decay of $1e^{-6}$.

PFA was compared against ADRMX (Demirel et al., 2023) and several established DG techniques, including MixStyle (Zhou et al., 2021), FDA (Yang & Soatto, 2020),

and RSC (Huang et al., 2020). The leave-one-domain-out protocol was repeated for each dataset, and experiments were conducted three times with different seeds to ensure robustness.

4.2.2 Training Procedure

I integrate PFA into the ADRMX framework. At each training step, I compute features for minibatches, select domain pairs according to the current curriculum stage, and generate remixed synthetic features. The losses I compute include cross-entropy for label classification, a contrastive loss to enforce class clustering, a domain discrimination loss, and a remixing loss derived from the synthetic features. The curriculum stages are defined using thresholds $\tau_1 = 0.6$ and $\tau_2 = 0.75$, while the remix ratio α is computed dynamically via Eq. (3.10), using $\delta = 0.2$ and $\tau = 0.7$. I also periodically update centroids and recompute D_c^{norm} every $K = 20$ steps (the `pair_update_freq` hyperparameter), so that my curriculum adapts as feature spaces evolve.

4.2.3 Hyperparameter Tuning

I tuned several key hyperparameters in my experiments to ensure an effective training. Specifically, I set the pair update frequency (`pair_update_freq`) to refresh the domain pairs and recompute centroid distances every 20 steps. The remix loss weight (λ_{rmxd}) was set to 1.0 to balance its contribution alongside other losses, while the contrastive loss weight (λ_{cnt}) was fixed at 0.5 to promote class-consistent feature clustering. For the dynamic remixing mechanism, I set the remix decay threshold (τ) to 0.7, which determines how sharply the remix weight α shifts away from equal mixing as domain distances increase. I also introduced a remix aggressiveness parameter (δ) with a value of 0.2, allowing moderate deviation from balanced mixing without generating unrealistic synthetic features. Additionally, to define the curriculum stages for progressively introducing more distant domain pairs, I selected thresholds $\tau_1 = 0.6$ and $\tau_2 = 0.75$. These values were chosen to ensure that early training focuses on remixing among closely related domains, while later steps gradually incorporate more challenging domain pairs as well. Remix repetitions per centroid pair were kept at one for computational efficiency. These settings were determined through empirical testing to balance remixing diversity with training

stability, particularly on datasets such as OfficeHome (Venkateswara et al., 2017) with large domain gaps.

Due to memory constraints, I used a batch size of 16. The optimizer employed was Adam, with a learning rate of 10^{-4}, $\beta_1 = 0.9$, and a weight decay of 1×10^{-6}. Training followed the leave-one-domain-out protocol on both datasets, running four splits per dataset for 5000 steps per split. The learning rate was reduced by a factor of 0.5 if the validation loss plateaued for 500 consecutive steps.

4.2.4 Comparative Methods

For a comprehensive evaluation, PFA was compared against:

- **ADRMX baseline** (Demirel et al., 2023): domain-specific and label-specific feature disentanglement with feature remixing.
- **MixStyle** (Zhou et al., 2021): feature statistic perturbation applied with $\alpha = 0.1$ and probability $p = 0.5$.
- **RSC** (Huang et al., 2020): representation self-challenging, dropping dominant features to encourage generalization.
- **FDA** (Yang & Soatto, 2020): frequency domain adaptation, swapping low-frequency components across domains.
- **ERM**: Empirical risk minimization, serving as the naïve baseline without domain generalization components.

4.2.5 Evaluation Metrics

Performance was assessed using the average classification accuracy across all target domains in each leave-one-domain-out setting. For PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017), results are reported as mean accuracy over four splits, following the evaluation standards in prior works.

4.2.6 Benefits and Observations

On the PACS (Li et al., 2017) dataset, PFA consistently outperformed ADRMX. On OfficeHome (Venkateswara et al., 2017), it achieved significant gains because it avoids early mixing of highly distant domains, which had previously led to unstable training by synthesizing noisy samples. Additionally, dynamic weighting through the parameter α effectively prevented the synthesis of unrealistic features. Overall, PFA introduced a structured approach to progressively remixing samples originating from different domains, which helped the model generalize to unseen domains in a more robust manner.

4.2.7 Limitations

While PFA demonstrates improved generalization performance, several practical limitations remain. Feature remixing increases computational overhead due to the need for pairwise centroid operations. In OfficeHome, large class counts and imbalance can amplify computational cost, occasionally requiring careful tuning of batch size and memory allocation. Future work may investigate adaptive mechanisms to mitigate such effects.

4.3 Results and Discussion

4.3.1 Results on PACS and OfficeHome

To evaluate the effectiveness of my proposed method (PFA), I compared it against several domain generalization baselines and state-of-the-art algorithms on two benchmark datasets: PACS and Office-Home. Tables 4.2 and 4.3 report the average accuracy obtained by each method.

Table 4.2 Comparison of different algorithms on the PACS dataset using a ResNet-50 backbone.

Algorithm	**PACS**	**Paper**
ERM	0.812	
IDCL2	0.820	(Li & Wang, 2023)
IDCL3	0.825	(Li & Wang, 2023)
DSU	0.842	(Chen et al., 2022)
DSU + LCL	0.854	(Wang et al., 2024)
DSU++	0.852	(Li et al., 2023)
DANN	0.814	(Ganin et al., 2016)
CORAL	0.820	(Sun & Saenko, 2016b)
Mixup	0.809	(Zhang et al., 2018)
GroupDRO	0.806	(Sagawa, 2022)
ANDMask	0.799	(Schulz, 2020)
RSC	0.820	(Huang et al., 2020)
DSU++ + LCL	0.860	(Chen & Liu, 2024)
MixStyle w/ domain label	0.837	(Zhou et al., 2021)
Deep Variational Encoding Network	0.720	(Kim, 2023)
ADRMX	0.830	(Demirel et al., 2023)
XDED	0.838	(Kyungmoon Lee, 2022)
PFA	**0.886**	my method

Table 4.3 Comparison of different algorithms on the Office-Home dataset using a ResNet-50 backbone.

Algorithm	**Office-Home**	**Paper**
ERM	0.629	
IDCL2	0.706	(Li & Wang, 2023)
IDCL3	0.708	(Li & Wang, 2023)
LCL	0.718	(Li & Wang, 2023)
DSU	0.661	(Chen et al., 2022)
DANN	0.616	(Ganin et al., 2016)
CORAL	0.633	(Sun & Saenko, 2016b)
Mixup	0.621	(Zhang et al., 2018)
GroupDRO	0.627	(Sagawa, 2022)
ANDMask	0.616	(Schulz, 2020)
RSC	0.637	(Huang et al., 2020)
DSU++ + LCL	0.723	(Chen & Liu, 2024)
MixStyle w/ domain label	0.655	(Zhou et al., 2021)
ADRMX	0.683	(Demirel et al., 2023)
XDED	0.650	(Kyungmoon Lee, 2022)
PFA	**0.752**	my method

For further examination, table 4.4 presents the class-wise accuracy of the proposed PFA method on the PACS dataset using the ResNet-50 backbone. PFA achieves strong performance across most object categories, particularly in semantically complex classes where cross-domain variation is more pronounced. For instance, PFA achieves 89.3% accuracy on the Elephant class and 87.4% on the Person class, reflecting the benefits of curriculum-based feature remixing. Other classes such as Giraffe (88.1%), Horse (83.9%), and Dog (85.7%) also show robust generalization across domains. Meanwhile, slightly lower accuracy is observed in classes like Guitar (80.5%) and House (78.6%), which are more sensitive to background and style variations. Overall, the class-wise results highlight PFA's effectiveness in promoting semantic consistency while addressing domain shifts.

Table 4.4 Class-wise accuracy (%) of the PFA method on the PACS dataset using the ResNet-50 backbone.

Class	Dog	Elephant	Giraffe	Guitar	Horse	House	Person
Accuracy (%)	85.7	89.3	88.1	80.5	83.9	78.6	87.4

Finally, before concluding the results, I also experimented with a simple extension to the original curriculum design by introducing a little randomness during the sample selection process. The idea behind this was to not rely strictly on centroid distances at every stage, but instead allow a small proportion of random sample pairs to be remixed along with the regular ones, just to see if that would help the model generalize better, especially in more visually abstract domains like Sketch or Clipart. Specifically, I injected 10% random pairs during the first stage of the curriculum and 20% in the second stage, while keeping the final stage fully structured. This setting was not tuned heavily, but chosen heuristically to simulate what a slightly more relaxed version of the curriculum might look like. The results shown below in tables 4.5 and 4.6 demonstrate that this variant does not harm the original performance, and in some domains, actually gives a small boost, which suggests that a touch of randomness can work as a regularizer in these types of domain generalization setups.

Table 4.5 Overall accuracy (%) on the PACS dataset comparing PFA and its variant with random sample injection. Results are averaged over all leave-one-domain-out folds using ResNet-50.

Method	**Photo**	**Art**	**Cartoon**	**Sketch**
PFA	91.4	89.2	88.5	85.2
PFA (Randomness Included)	91.2	88.7	88.9	85.5

Table 4.6 Overall accuracy (%) on the OfficeHome dataset comparing PFA and its variant with random sample injection. Results are averaged over all leave-one-domain-out folds using ResNet-50.

Method	**Art**	**Clipart**	**Product**	**Real World**
PFA	73.4	70.1	78.6	78.7
PFA (Randomness Included)	73.9	70.9	77.9	78.2

4.4 Ablation Study

This chapter presents a thorough ablation study to analyze the contribution of each critical design choice in my proposed **(PFA)** method, developed as an extension to the ADRMX baseline by examining the impact of each component on the model's domain generalization performance.

4.4.1 Motivation for Ablation

These studies help quantify the gains brought by PFA. Through my experiments, it became clear that the success of PFA depends on several key design factors. One crucial aspect is whether remixing is performed randomly or scheduled based on centroid distances, as this choice significantly influences how synthetic samples contribute to domain alignment. Another important factor is the use of dynamic remix weights rather than fixed mixing ratios, which allows the model to adjust the blending of features more precisely according to inter-domain similarities. Additionally, the frequency at which centroid distances are recalculated during training affects how well the curriculum remains aligned with the evolving feature space. The sensitivity to the threshold τ, which controls remix strength, also plays a substantial role, as overly aggressive or conservative remixing can either destabilize learning or limit the method's effectiveness. Finally, the influence of the contrastive loss weight λ_{cnt} proves affective, impacting the balance between enforcing intra-class compactness and maintaining inter-class separation.

These studies help quantify the gains brought by PFA and provide valuable insight into the method's practical deployment and tuning.

4.4.2 Effect of Curriculum vs. Random Remixing

A core novelty in PFA is that remixing is no longer random. Instead, centroid distances between same-class domains are computed at intervals during training. Pairs of domains closer in feature space are remixed first, progressing toward more distant pairs as training advances. To isolate the impact of this curriculum strategy, I compared:

- ADRMX baseline (random pairing, random remixing)
- PFA with random pairing and remixing
- PFA with curriculum-driven pairing

Table 4.7 shows the average classification accuracy on PACS and OfficeHome datasets.

Table 4.7 Impact of curriculum-driven remixing vs. random remixing.

Method	**PACS (%)**	**OfficeHome (%)**
ADRMX Baseline	83.0	68.3
PFA (Random Remixing)	84.0	69.7
PFA (Curriculum Remixing)	**88.6**	**75.2**

The results show that even random remixing can improve over ADRMX. However, PFA's curriculum brings a notable further boost, particularly for OfficeHome, where domain distances are larger.

4.4.3 Effect of Dynamic Remixing Strength

Unlike ADRMX, which implicitly mixes features with a roughly fixed ratio, PFA dynamically adjusts the remix weight α depending on how close or distant two domains are in the feature space. For closer domains, remixing is strong (α closer to 0.5), while distant domains receive highly biased remixing to avoid unrealistic synthetic features.

I compared:

- Curriculum remixing with fixed $\alpha = 0.5$
- Curriculum remixing with dynamic α

Results are shown in Table 4.8.

Table 4.8 Effect of dynamic remixing strength vs. fixed remixing ratio.

Remixing Strategy	**PACS (%)**	**OfficeHome (%)**
Curriculum remixing, fixed $\alpha = 0.5$	85.1	68.8
Curriculum remixing, dynamic α	**88.6**	**75.2**

Dynamic weighting prevents extreme or unrealistic interpolations between very dissimilar domains, resulting in better generalization, especially on OfficeHome.

4.4.4 Pair Update Frequency

My method recalculates centroid distances every K steps, ensuring the curriculum remains aligned with the evolving feature space. Updating too frequently can cause instability, while updating too slowly might leave the curriculum stale.

I tested:

- Updates every 5 steps
- Updates every 20 steps (default)
- Updates every 50 steps

Table 4.9 reports results.

Table 4.9 Impact of pair update frequency on domain generalization.

Update Frequency (steps)	**PACS (%)**	**OfficeHome (%)**
5	85.3	69.4
20 (default)	**88.6**	**70.8**
50	84.9	69.6

The best performing result is achieved with 20 steps, balancing timely updates and training stability.

4.4.5 Threshold τ for Dynamic Remixing

The threshold τ governs how rapidly the remix weight α decays for distant domain pairs. A low τ leads to very cautious remixing (few synthetic examples), while a high τ risks unrealistic blends.

I tested $\tau \in \{0.3, 0.5, 0.7, 0.9\}$. Results are in Table 4.10.

Table 4.10 Effect of τ on domain generalization performance.

τ	**PACS (%)**	**OfficeHome (%)**
0.3	84.6	69.3
0.5	85.1	69.0
0.7	**88.6**	**75.2**
0.9	85.3	69.7

A moderate threshold of 0.7 yielded the best balance between meaningful remixing and avoiding unrealistic feature synthesis.

4.4.6 Effect of Contrastive Loss Weight λ_{cnt}

Contrastive loss complements remixing by ensuring that samples of the same class cluster tightly even after synthetic mixing. I evaluated:

- No contrastive loss ($\lambda_{\text{cnt}} = 0.0$)
- Default weight ($\lambda_{\text{cnt}} = 0.5$)
- Higher weight ($\lambda_{\text{cnt}} = 1.0$)

Results are shown in Table 4.11.

Table 4.11 Impact of contrastive loss weight on domain generalization.

λ_{cnt}	**PACS (%)**	**OfficeHome (%)**
0.0	83.7	69.5
0.5	**88.6**	**75.2**
1.0	85.3	70.4

Moderate contrastive strength helped the most. Too high a weight risked overwhelming other losses.

4.4.7 Summary of Ablation Findings

From these experiments, several important conclusions can be drawn. PFA's curriculum-based pairing proves critical and consistently outperforms random remixing, highlighting the benefit of progressively introducing more challenging domain pairs during training. The use of dynamic remix weights is essential for preventing unrealistic synthetic samples, especially when mixing features from distant domains. Furthermore, updating centroids approximately every 20 steps offers the best trade-off between maintaining up-to-date domain relationships and ensuring training stability. A moderate threshold value τ around 0.7 strikes a balance between encouraging remix diversity and producing semantically realistic synthetic samples. Finally, incorporating a moderate contrastive loss significantly enhances feature clustering, an effect particularly important on the OfficeHome dataset.

Overall, these ablation studies clearly demonstrate how each component of PFA contributes to more robust domain generalization compared to ADRMX, especially under strong domain shifts.

5. CONCLUSION& FUTURE WORK

5.1 Conclusion

The experiments presented in this thesis demonstrate the effectiveness of the PFA method in addressing the challenge of domain generalization. Domain generalization aims to develop models that perform well on unseen target domains, a problem that remains significant due to the diverse ways domains can differ in real-world applications. The core issue addressed in this work is that naive feature remixing, as used in prior approaches like ADRMX (Demirel et al., 2023), does not account for how similar or different domains truly are, potentially generating unrealistic synthetic samples that can harm the training procedure.

The proposed PFA method tackles this problem by introducing a structured, distance-aware approach to feature remixing. Specifically, PFA computes distances between class centroids across domains and uses this information to guide which domain pairs should be remixed, prioritizing those that are closer in feature space. Additionally, PFA employs a dynamic weighting mechanism that adjusts the remixing ratio based on the degree of domain similarity, helping to avoid blending features from domains that are too dissimilar. Finally, by incorporating a curriculum-style training strategy, PFA gradually exposes the model to increasingly challenging domain shifts, promoting more robust learning.

Experimental results on both PACS and OfficeHome benchmarks confirm that PFA consistently outperforms ADRMX and several other competitive methods, validating its core design principles and demonstrating its potential as a practical and lightweight solution for domain generalization. These findings suggest that integrating distance-aware strategies into feature remixing can significantly enhance a model's ability to generalize across diverse domains, contributing a valuable direc-

tion for future research in this field.

5.2 Future Work

While the experiments in this thesis demonstrate the effectiveness of PFA for domain generalization, several promising directions remain open for future research.

One important avenue is the extension of PFA to other backbone architectures. The current implementation focuses on ResNet-50 (He et al., 2016), but integrating PFA into modern architectures such as Vision Transformers (ViT) (Dosovitskiy et al., 2021), Swin Transformers (Liu et al., 2021), and hybrid CNN-Transformer models could reveal how global attention mechanisms might further enhance domain generalization.

Another direction concerns scalability to larger datasets. Evaluating PFA on more extensive and diverse benchmarks beyond PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017), such as DomainNet (Peng et al., 2019) or newly proposed large-scale DG datasets, would provide stronger evidence of PFA's robustness and its capacity to handle more severe domain shifts.

There is also significant potential in developing dynamic curriculum learning strategies. Currently, PFA uses a curriculum by controlling remix strength through distance-based weighting. Future research could build on this by designing a fully dynamic curriculum that adapts the remixing difficulty in response to the model's performance or confidence, potentially leading to more efficient learning.

Another promising direction is integrating PFA with self-supervised learning frameworks. Recent trends in domain generalization emphasize the benefits of combining self-supervised pretraining with DG-specific strategies. Exploring how PFA can be incorporated into frameworks such as MoCo (He et al., 2020) or SimCLR (Chen et al., 2020) could further enhance the robustness and generalization capacity of learned representations.

Finally, while PFA remains lightweight from an architectural perspective, its pairwise centroid computations could become computationally expensive in large-scale applications. Future work might focus on optimizing this step or devising approximate methods to ensure PFA remains practical for real-time or resource-constrained deployment.

Overall, these directions suggest that PFA holds substantial potential for broader applicability, laying the groundwork for future advances in robust and generalizable computer vision systems.

BIBLIOGRAPHY

Balaji, Y., Sankaranarayanan, S., & Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *ICML*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, (pp. 1597–1607).

Chen, Y. & Liu, Q. (2024). Ladder curriculum learning for domain generalization in cross-domain classification. In *CVPR*.

Chen, Z., Xu, Y., Jia, Z., Zhao, Z., Wang, L., & Hsieh, C.-J. (2022). Uncertainty modeling for out-of-distribution generalization. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

Demirel, B., Aptoula, E., & Ozkan, H. (2023). ADRMX: Additive Disentanglement of Domain Features with Remix Loss. arXiv preprint arXiv:2308.06624.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. In *Journal of Machine Learning Research*, volume 17, (pp. 1–35).

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*.

Gulrajani, I. & Lopez-Paz, D. (2020). In search of lost domain generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 9729–9738).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770–778).

Hu, R., Liao, Z., & Xia, G.-S. (2023). Contrastive channel-wise style disentanglement for domain generalization. In *CVPR*.

Huang, R., Wang, S., Nallapati, R., & Xiang, B. (2020). Self-challenging improves cross-domain generalization. In *ECCV*.

Jiang, X., Huang, J., Jin, S., & Lu, S. (2023). Momentum difficulty framework for domain generalization. arXiv preprint arXiv:2309.00844.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. In *NeurIPS*.

Kim, J. e. a. (2023). Domain generalization via encoding and resampling in a unified latent space. *ICML*, *XX*(YY), ZZ.

Kyungmoon Lee, Sungyeon Kim, S. K. (2022). Cross-domain ensemble distillation for domain generalization. *CVPR*, *XX*, YY.

Lee, S., Oh, J., & Kim, H. J. (2024). Adapter-based domain generalization for large foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. (2018). Domain generalization via conditional invariant representations. In *AAAI*.

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

Li, X., Hu, Z., Liu, J., Ge, Y., Dai, Y., & Duan, L.-Y. (2023). Modeling uncertain feature representation for domain generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Li, X. & Wang, Y. (2023). Inter-domain curriculum learning for domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(6), 7654–7667.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015a). Learning transferable features with deep adaptation networks. In *ICML*.

Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015b). Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, (pp. 97–105). JMLR.org.

Oquab, M., Darcet, T., Lavril, T., Szafraniec, M., Vincent, P., Joulin, A., Mairal, J., Labatut, P., & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193.

Peng, X., Bai, Q., Xia, X., Huang, Z., & Saenko, K. (2019). Moment matching for multi-source domain adaptation. In *ICCV*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sagawa, S. e. a. (2022). Pac generalization via invariant representations. In *ICML*.

Schulz, M. e. a. (2020). Learning explanations that are hard to vary. In *NeurIPS*.

Shi, M., Xie, F., Yang, J., Zhao, J., Liu, X., & Wang, F. (2023). Cutout with patch-loss augmentation for improving generative adversarial networks against instability. *Journal of Visual Communication and Image Representation*, *95*, 103975.

Sun, B. & Saenko, K. (2016a). Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, (pp. 443–450).

Sun, B. & Saenko, K. (2016b). Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*.

Torralba, A. & Efros, A. A. (2011). Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1521–1528). IEEE.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *ICML*.

Unni, V., Jere, S., Das, A., Parikh, D., & Batra, D. (2023). Vqa-gen: A visual question answering benchmark for domain generalization. *arXiv preprint arXiv:2311.00807*, *abs/2311.00807*(1), 1–10.

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 5385–5394).

Wang, X., Luo, S., & Gao, Y. (2024). Ladder curriculum learning for domain generalization in cross-domain classification. *IEEE Access*, *12*, 95356–95367.

Wang, Y., Gao, J., Wang, Q., Yang, X., & Du, J. (2024). Curriculum learning-based domain generalization for cross-domain fault diagnosis with category shift. *Mechanical Systems and Signal Processing*, *212*, 111295.

Yang, Y. & Soatto, S. (2020). Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (pp. 6023–6032).

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *ICLR*.

Zhou, K., Gong, X., Song, Y.-Z., & Xiang, T. (2023). Domain contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 3596–3606).

Zhou, K., Yang, Y., Hospedales, T. M., & Xiang, T. (2021). Domain generalization with mixstyle. In *International Conference on Learning Representations (ICLR)*.