Domain Generalization Using Robust Data Augmentation and Invariant Representation Learning

[*]Note: Sub-titles are not captured in Xplore and should not be used

Muhammad Uzair Ashraf
FA25-RAI-020
COMSATS University Islamabad
Islamabad, Pakistan
ashraf.uzair01@gmail.com

Mustehzar Jarri Butt
FA25-RAI-021
COMSATS University Islamabad
Islamabad, Pakistan
mustehzar2311@gmail.com

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In modern machine learning, among many important .challenges, there exist the challenge of domain generalization. This is challenge arises particularly when the AI systems are deployed increasingly in those environments that is different from the data on which they were trained on. When we talk about different domain in a single model, they can experience very severe performance degradations; this happens because of the variations such as lighting, texture, style, devices or environmental conditions etc. This issue is a big gap between training and real world distributions that limits the reliability and scalability of deep learning models across many applications, that includes computer vision, autonomous systems and medical imaging

Recent research presents techniques that can enable a model to learn representations can remain stable across multiple domains. Robust data augmentation, a technique that can expand domain diversity during training artificially, and invariant representation learning that can extract insensitive features to a superficial domain specific variant, are the two techniques that have gained prominence in this area. When put together these two approaches provide a solid foundation for developing models that can effectively generalize unseen environment without having to target a domain during its training.

There's been a lot of interest in domain generalization lately because it actually matters in real applications, and traditional supervised learning still falls short. Even with all the progress made so far, models still struggle to perform reliably when the environment changes a lot. That's why researchers are still exploring better ways to make models more adaptable without hurting their accuracy or efficiency.

II. Related Work

In order to improve generalization capability without requiring domain labels, data centric solutions such as augmentation strategies and representation regularization have been repoted in recent studies in this domain. For example, MixStyle has inspired further such as mixed domain feature modification. Based on this FIXED(Frustratingly Easy Domain Generalization with Mixup) suggests that robustness can be significanlty improved by implicitly mixing domain characteristics of samples, using simple mixup-based augmentation. The idea is to interpolate both features and the labels, it will reduce reliance on spurious correlations and would also promote smooth decision boundaries. FIXED achieves consistent improvement across DG standards without requiring domain supervision, proposing that for robust learning is not always necessary [1].

In another paper Normalization-Guided Augmentation (NormAUG) takes a different route by tweaking the normalization statistics inside the network. Instead of mixing samples like Mixup based strategy does, NormAUG changes the mean and variance of the activations during training. This helps the model rely less on fixed, domain-specific normalization layers. Compared to common augmentation methods, it tends to perform better because it forces the internal features to vary more, which improves how well the model handles new domains. Overall, it shows that you can boost robustness by adjusting the normalization behavior itself rather than only modifying the input data. [2]

Adding to these two approaches, Cross Domain Feature Augmentation (CDFA) introduces a more structured approach for augmentation. CDFA extracts style embeddings and reunited them with features from other images instead of random perturbations. It mimics realistic domain shift and results in controlled but spread out augmentation method. CDFA shows higher accuracy on unseen environment on test bench such as

Identify applicable funding agency here. If none, delete this.

PACS and VLCS and highlights the significance of modeling content and style independently in DG tasks. [3].

Lastly, there's this ICLR 2024 paper about Improving Domain Generalization with Domain Relations, and it tells us about whether knowing how the source domains relate to each other can help the model deal with new ones. They use this graph-based idea to capture what the domains have in common and what makes them different. It's idea is not like data augmentation methods; rather it tells us about understanding the structure behind the domains. By learning these similarity graphs, the model kind of figures out which patterns should stay the same even when it sees a completely new domain. The main idea is that actually thinking about the relationships between domains gives an advantage that simple augmentation methods can't really provide. [4].

In Conclusion these four papers look at domain generalization from pretty different angles, but they still kind of connect. One paper sticks to a simple mixup-style idea (FIXED), another tries messing with the statistics of the features (NormAUG), another focuses on mixing semantic features in a smarter way (CDFA), and the last one looks at how different domains relate to each other using graphs. If you put them together, you basically get a rough picture of how people are currently approaching DG in places like NeurIPS, ICML, and ICLR, from basic augmentation tricks to more structural or relational methods.

TABLE I

	Comparative Summary of Related Work		
Paper	***Key Idea***	***Strengths***	***Limitations***
FIXED (NeurIPS 2024)	Mixup based interpolation across domain	Extremely simple; strong baselines; computationally cheap	Depends heavily on mixup quality; limited control over domain shifts
NormAUG (NeurIPS 2023)	Augmentation via normalization layer statistics	Lightweight; directly manipulates internal features	May be sensitive to normalization design and architecture
Cross-Domain Feature Augmentation (NeurIPS 2024)	Feature space perturbations simulate unseen domains	Semantic consistency; flexible control over shifts	Requires stable feature embeddings; can introduce noise
Domain Relations (ICLR 2024)	Modeling relationships among source domains	Captures structured domain similarity; strong theoretical grounding	Suffers when source domains are very few or imbalanced

III. Research Gap

Even after there is a lot of progress made and shown in these paper but there still remains some gaps

- **Limited synergy between augmentation and relational approaches**: Existing methods usually rely on either data/feature augmentation or modeling relationships between domains, but there are very few models that combine these two complementary perspectives. A hybrid method could leverage domain relations to guide augmentations more intelligently.
- **Changing features doesn't always keep the meaning of the data the same across different tasks:**Even though feature space augmentation is promising, high level features may become unstable across different architectures or datasets. That makes sure that semantic preservation remains difficult.
- **Most methods assume multiple diverse source domains:**Real world settings often have only one or two domains available, causing relational or statistical augmentation techniques to underperform. Adaptive approaches, under extremely limited domain diversity are still lacking.
- **Computational simplicity vs. robustness trade off remains unresolved:**Lightweight methods like mixup-based perform surprisingly well, but their limits and failure cases are not fully understood. More clarity is needed on when simple augmentation is sufficient and when sophisticated modeling becomes necessary.

IV. Problem Statement

In the referenced papers, we are shown quite astounding progress from the recent domain generalization methods, but despite all that the models' performance still fail when it comes to unseen environments that differ from training domains. These approaches rely, either on augmentation-based techniques or on modeling structural relationships between domains, but they are rarely combine both aspects. On top of that, most datasets that are mostly used are limited in domain diversity aspect, which causes many DG strategies to underperform.

The problem stated is to implement an approach that can improve model robustness across unseen domains by using techniques such as Empirical Risk Minimization (ERM) combined with Mixup-based augmentation.

V. Methodology

This study shows how Empirical Risk Minimization (ERM) and Mixup-augmented ERM perform in a Domain Generalization (DG) area. To simulate multiple source domains and one unseen target domain, we trained on two distinct grayscale image datasets MNIST and FashionMNIST , while KMNIST dataset was used as the unseen target domain. All models are trained from scratch using a unified pipeline designed for fair and reproducible comparison.

The datasets undergo a shared pre-processing pipeline consists of grayscale normalization, pixel scaling, and resizing to 224×224 to ensure compatibility with the ResNet-18 architecture. The MNIST and FashionMNIST training splits are concatenated into a single multi-domain training set, reflecting the DG assumption that a model learns to extract domain-invariant features from diverse but related sources.

KMNIST is never seen during training, making it an ideal unseen-domain benchmark.

A modified ResNet-18 architecture serves as the backbone model. Because the datasets contain single-channel images, the initial convolution layer is adjusted to accept 1-channel input. The final fully connected layer is replaced with a 10-class output layer corresponding to the shared label space across all datasets.

Two training strategies are evaluated. In the ERM, batches are sampled from the mixed MNIST + FashionMNIST dataset and optimized using cross-entropy loss. In the Mixup variant, samples are interpolated using the Mixup equation:

$$x' = \lambda x_i + (1 - \lambda)x_j \tag{1}$$

$$y' = \lambda y_i + (1 - \lambda)y_j \tag{2}$$

where $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ This technique encourages the model to learn smoother decision boundaries and potentially more robust representations.

After training the models, models are evaluated on three separate domains: MNIST, FashionMNIST, and the unseen KMNIST dataset. Evaluation includes accuracy scores, loss and accuracy curves over epochs, as well as confusion matrices and classification reports. These metrics collectively allow us to compare how well each training strategy generalizes beyond its training domains.

VI. Experimental Results

A. Training Behavior

Training curves show that ERM converges more quickly and achieves higher training accuracy than Mixup. Mixup exhibits slower convergence, as expected due to the additional regularization introduced by sample interpolation. The training dynamics for both ERM and Mixup are shown in "Fig. 1" and " Fig. 2

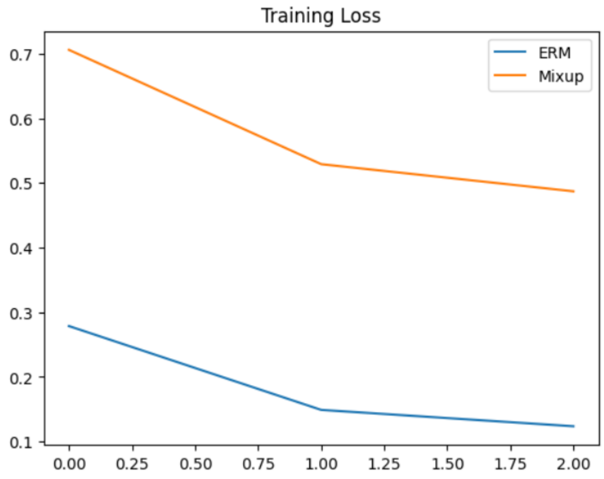

Fig. 1. Training Loss.

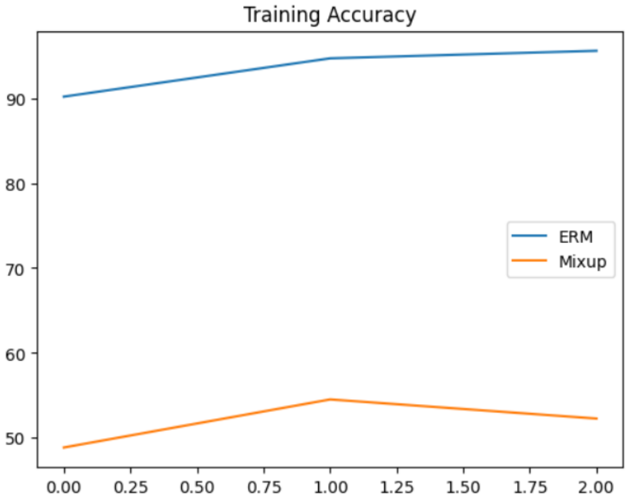

Fig. 2. Training Accuracy.

B. Performance on Source Domains

Both ERM and Mixup achieve strong accuracy on MNIST and FashionMNIST:

- ERM: MNIST 98.73%, FashionMNIST 91.02%
- Mixup: MNIST 99.07%, FashionMNIST 91.85%

Mixup slightly improves MNIST generalization but does not significantly outperform ERM on FashionMNIST. As shown in Figure. 3 and Table II

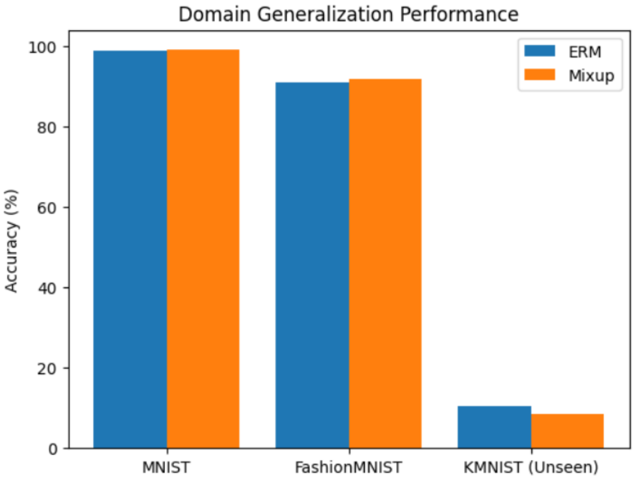

Fig. 3. Domain Generalization Performance.

C. Generalization to Unseen Domain (KMNIST)

The KMNIST dataset represents a substantial distribution shift relative to the training domains. Both models perform poorly:

- ERM: 10.63%
- Mixup: 8.40%

This shows that simple ERM or Mixup alone is insufficient for strong generalization when the unseen domain differs drastically from the source domains.

D. Confusion Matrix

Both ERM and Mixup models exhibit strong diagonal dominance on source domains (MNIST and FashionMNIST), indicating successful learning of discriminative features (). However, performance degrades substantially on the unseen KMNIST domain, with increased off-diagonal scatter and weakened diagonal strength.

The confusion matrices reveal systematic misclassification patterns on KMNIST, where predictions cluster around specific incorrect classes. This suggests the models rely on low-level stroke patterns rather than semantic understanding. While Mixup shows marginal improvement over ERM, both approaches exhibit qualitatively similar confusion structures, indicating that interpolation-based augmentation provides limited benefits for extreme domain shifts between different writing systems (Latin/fashion symbols vs. Japanese Hiragana). The shared label space (0-9) does not ensure semantic transferability across fundamentally different visual domains.

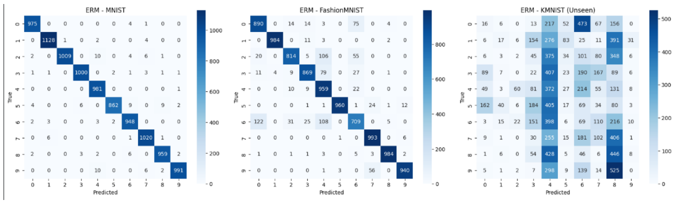

Fig. 4. Confusion Matrices for ERM model on All Datasets.

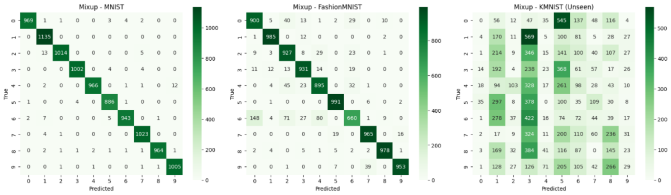

Fig. 5. Confusion Matrices for MixUP model on All Datasets.

TABLE II
Comparative Summary of Models' Accuracy

	Comparative Summary of Models' Accuracy		
Models	***ERM***	***Mixup***	***Difference***
MNIST	98.73%	99.07%	+0.34%
Fashion MNIST	91.02%	91.85%	+0.83
KMNIST (Unseen)	10.63%	8.40%	-2.23%

References

[1] K. Xu, L. Wang, and J. Huang, "Frustratingly Easy Domain Generalization with Mixup," NeurIPS, 2024.

[2] Qi, L., Yang, H., Shi, Y. and Geng, X., 2024. Normaug: Normalization-guided augmentation for domain generalization. IEEE Transactions on Image Processing, 33, pp.1419-1431.

[3] Y. Chen, D. Li, and X. Wang, "Cross-Domain Feature Augmentation for Domain Generalization," NeurIPS, 2024.

[4] Yao, Huaxiu, et al. "Improving domain generalization with domain relations." arXiv preprint arXiv:2302.02609 (2023).