Advanced ML PA4

Muhammad Huraira Anwar [1] **Adeen Ali Khan** [2]

Abstract

This report explores the concepts of **Unsupervised Domain Adaptation (UDA)** and **Domain Generalization (DG)**, focusing on their theoretical underpinnings, implementation, and practical performance across various datasets. UDA is investigated using methods like Domain Adaptation Network (DAN) and Domain Adversarial Neural Networks (DANN), which aim to align source and target domains through statistical and adversarial techniques. Experiments include popular datasets such as Office-31, Office-Home, and Digits datasets, analyzing the performance of pre-trained ResNet, DAN, and DANN under different domain shifts.

In the DG task, we examine the effectiveness of Invariant Risk Minimization (IRM) and its variants, including IB-IRM and Pareto IRM (PAIR). These methods are evaluated using datasets like RMNIST and PACS, assessing their ability to generalize across unseen target domains. Key metrics include OOD accuracy and the trade-offs between IID and OOD accuracy, tracked over training epochs.

Additionally, the principles of **Disentangled Representation Learning (DRL)** are studied through the β-VAE framework, with experiments on datasets like CelebA and DSprites. The disentanglement of generative factors is analyzed using quantitative metrics and visualizations, highlighting the impact of β on reconstruction quality and disentanglement effectiveness.

The report provides a comprehensive comparison of these methods, discussing their strengths, limitations, and the trade-offs encountered in achieving robust domain adaptation and generalization.

Introduction

Deep learning has revolutionized machine learning, enabling significant advancements across domains, particularly when trained and evaluated on data with similar distributions. Techniques such as transfer learning have become popular for leveraging pre-trained models, allowing their application to new domains with minimal labeled data. While these approaches achieve impressive performance in in-domain scenarios, they struggle with out-of-domain (OOD) data, where the training and testing distributions differ. Addressing this challenge is critical for developing robust models suitable for real-world applications, where data variability is inevitable.

In this work, we explore two major areas of domain adaptation and generalization: **Unsupervised Domain Adaptation (UDA)** and **Domain Generalization (DG)**. UDA focuses on adapting models trained on a labeled source domain to an unlabeled target domain, bridging the domain gap. We investigate representative UDA methods, including Domain Adaptation Network (DAN) and Domain Adversarial Neural Networks (DANN). These approaches align feature spaces between domains using statistical and adversarial techniques. To evaluate their performance, we use datasets such as Office-31, Office-Home, and Digits datasets, each introducing unique domain shifts and challenges.

For domain generalization, we examine **Invariant Risk Minimization (IRM)** and its extensions, IB-IRM and Pareto IRM (PAIR), which aim to identify invariant features across environments. Using datasets like RMNIST and PACS, we evaluate these methods' ability to generalize to unseen target domains. The emphasis is on understanding the trade-offs between in-domain (IID) and out-of-domain (OOD) performance, which is critical for assessing model robustness.

Beyond domain adaptation and generalization, we delve into **Disentangled Representation Learning (DRL)** using the β-VAE framework. By training on datasets such as CelebA and DSprites, we analyze the models' ability to disentangle generative factors like shape, orientation, and color. This helps in understanding how models capture meaningful representations and their potential limitations.

This study not only benchmarks multiple approaches for domain adaptation and generalization but also evaluates their effectiveness across various datasets and challenges. By comparing the performance of these methods, we aim to uncover insights into their inductive biases and provide a roadmap for developing more robust models capable of handling real-world variability.

1. Unsupervised Domain Adaptation

1.1. Methodology

This methodology focuses on implementing and evaluating UDA approaches using the Office-31 and Digits datasets,

Model	A → W	A → D	W → A	W → D	D → A	D → W	Avg
ResNet	65.53	61.24	17.64	35.94	15.16	25.03	**36.76**
DAN	51.45	50.60	39.26	97.59	39.26	93.20	**61.89**
DANN	25.47	38.17	72.66	46.97	25.38	36.59	**40.87**

Table 1. Comparison of ResNet-50, DAN , DANN model accuracies on unsupervised domain adaptation for the Office-31 Dataset (A-Amazon, W- Webcam, D-DSLR).

Model	M → S	M → U	S → M	S → U	U → M	U → S	Avg
ResNet	35.17	28.00	48.11	54.38	15.14	17.12	**32.98**
DAN	40.12	65.12	30.55	85.42	59.93	80.11	**60.20**
DANN	26.13	48.22	24.89	51.95	42.87	34.42	**38.08**

Table 2. Comparison of ResNet-50, DAN , DANN model accuracies on unsupervised domain adaptation for the Digits Datasets (M - MNIST, S-SVHN , U-USPS) .

aiming to transfer knowledge from a labeled source domain to an unlabeled target domain for high target domain accuracy. The Office-31 dataset includes three distinct domains: Amazon (A), DSLR (D), and Webcam (W), with six domain transfer tasks, such as $A \rightarrow W$ and $D \rightarrow A$, to evaluate UDA methods across different domain shifts. It consists of 4,652 images from 31 object categories. The Digits dataset includes MNIST (M), USPS (U), and SVHN (S), representing varying complexities in digit recognition tasks, with three domain transfer tasks $M \rightarrow U$, $U \rightarrow M$, and $S \rightarrow M$.

Three models are employed: the baseline pre-trained ResNet-50, which is used without domain adaptation techniques to act as a benchmark by training on the source domain and testing on the target domain; Domain Adaptation Network (DAN), which minimizes the Maximum Mean Discrepancy (MMD) between feature distributions of source and target domains using MMD layers in the latent space to align features; and Domain Adversarial Neural Networks (DANN), which employ adversarial learning to align source and target domain feature distributions by integrating a gradient reversal layer into the model for adversarial training to enforce domain-invariant features. The procedure involves implementing the models with pre-trained weights for ResNet-50 fine-tuned on the source domain, including MMD layers for DAN, and designing a gradient reversal layer with an adversarial training loop for DANN. The models are trained on the source domain of Office-31 and Digits datasets and evaluated on the target domain for each transfer task. Performance metrics, such as target domain accuracy, are reported, and a comparison is made across tasks, including visualization of domain alignment using t-SNE plots.

Hyperparameters are tuned for DAN, such as MMD kernel settings, and for DANN, such as the learning rate for adversarial optimization. Insights are drawn on how domain adaptation techniques reduce the gap between source and target domains, comparing the effectiveness of feature alignment (DAN) and adversarial learning (DANN) in improving target domain accuracy. The strengths and limitations of each approach are highlighted, including DANN's sensitivity to adversarial training and DAN's dependency on MMD kernel settings. This methodology ensures a systematic evaluation of UDA techniques on Office-31 and Digits datasets, providing critical insights into their effectiveness and areas for further improvement.

1.2. Discussion

The learning parameter used was **0.001** across all models. DAN and DANN models were comparatively shallower models (with less number of layers and consequently fewer parameters, which may limit their learning capacity, especially when dealing with complex datasets like those requiring transfer learning) as compared to ResNet50 pretrained model, hence both of them were trained for **10** epochs each, while ResNet50 was only finetuned for **2** epochs (it starts to overfit after more than 3 epochs).

For DAN, gamma values for MMD loss were selected from a range of values of 2 to the power of [**-8,9**]. The lambda parameter was set to **1.0**, and the MMD loss regularization parameter was set to **5.0** to help it focus on rigorous domain alignment. For DANN, in the GRL, to calculate lambda, gamma was initialised to **10** and ρ was calculated per each batch, depending on epoch number, batch index, and total number of data samples.

As presented in the tables, on similar domains, like MNIST and USPS, or when trained on a dataset with more examples (like Amazon which has 3-4 times more examples than Webcam and DSLR), the ResNet model performs reasonably good . However, on domains with even a slight shift or lesser examples, its accuracy drops considerably. Both, the other

models however focus on aligning the domains, but their drawbacks are either the multiple terms in the loss or the adversarial setup in case of DANN, which dont result in as smooth convergence, and hence the accuracies in some domain setups underperform than ResNet.

However, on average, both DAN and DANN perform better than ResNet due to their ability to reduce the domain discrepancy through domain adaptation techniques. By aligning feature distributions between the source and target domains, these models achieve improved generalization on the target domain, especially when there is significant domain shift. DAN achieves this alignment using the Maximum Mean Discrepancy (MMD) loss, which ensures that the learned representations are domain-invariant, while DANN leverages adversarial learning to achieve similar domain alignment. Although these approaches introduce additional complexity, their effectiveness in handling domain shifts leads to better overall performance compared to the ResNet model, which lacks explicit domain adaptation mechanisms.

1.3.1. DAN vs DANN

Domain Adaptation Network (DAN) and Domain Adversarial Neural Network (DANN) represent two distinct yet complementary approaches to domain adaptation, differing primarily in their structural design and loss functions.

DAN utilizes the Maximum Mean Discrepancy (MMD) metric to align feature distributions between source and target domains in the latent space. By minimizing MMD across layers of the network, DAN reduces the domain shift, allowing for a more effective transfer of learned features. The key idea is to bring the distributions closer using a statistical divergence measure, ensuring that the feature extractor generates domain-invariant features without explicitly interacting with the classification head during adaptation. Structurally, DAN is designed with multiple MMD loss layers integrated into its architecture, each contributing to aligning the feature space progressively.

On the other hand, DANN employs an adversarial training framework to achieve domain invariance. The network incorporates a domain classifier and a gradient reversal layer (GRL) in addition to the feature extractor and classification head. The GRL plays a pivotal role by reversing the gradient direction of the domain classifier during backpropagation, encouraging the feature extractor to generate representations that are indistinguishable between the source and target domains. Unlike DAN, which uses a divergence measure for alignment, DANN relies on adversarial loss, where the domain classifier competes against the feature extractor, thereby implicitly aligning the domains.

While DAN focuses solely on statistical measures like MMD for domain alignment, DANN introduces a dynamic adversarial process that adds flexibility but can lead to instability during training. The choice between these methods depends on the complexity of the domain shift and the robustness of the training setup. For instance, DAN may perform better when the domain shift is subtle and quantifiable, whereas DANN excels in scenarios with complex, non-linear shifts requiring a more flexible alignment strategy.

1.3.2. Code pipeline used for DAN vs DANN

The code pipelines for DAN and DANN differ significantly due to their underlying mechanisms for domain adaptation. DAN leverages the Maximum Mean Discrepancy (MMD) to align source and target feature distributions. Its pipeline involves incorporating MMD layers into the deep network, typically at multiple feature extraction stages. These layers compute the divergence between domain features and minimize it during backpropagation. The training process is relatively straightforward, with the key challenge being the careful tuning of MMD kernel parameters to ensure effective domain alignment without overfitting.

On the other hand, DANN employs adversarial learning via a gradient reversal layer (GRL). The GRL is inserted into the network, acting as a bridge between the feature extractor and a domain classifier. During training, the GRL flips the gradients from the domain classifier, forcing the feature extractor to learn domain-invariant features. This adversarial setup makes DANN's pipeline more complex and sensitive to hyperparameter choices, particularly for adversarial training stability. While both pipelines are effective for domain adaptation, DAN's statistical approach is computationally stable but dependent on kernel settings, whereas DANN's adversarial approach offers strong theoretical guarantees for domain-invariant features but requires careful handling of optimization dynamics.

1.3.3. ResNet

Performance drops are observed from **13%** to **35%** when comparing accuracy on source domain and on target domains. On average, between all the domains, the drop in performance is about **20%** which is quite a significant drop, considering that the highest accuracy is just **60%**.

The feature distributions of source and target domains are visualized using TSNE. As we can see, the domain shift is quite profound, even after finetuning the model to source domain for a few epochs, the domains are not aligned at all. The clusters on the left side represent the source domain, while the clusters on the right side represent the target domain. They are clearly separable by a line.

1.3.4. DAN vs ResNet

The TSNE plot obtained from the DAN model demonstrates better domain alignment compared to the baseline ResNet. DAN utilizes the Maximum Mean Discrepancy (MMD) loss aim to address domain adaptation by explicitly reducing the

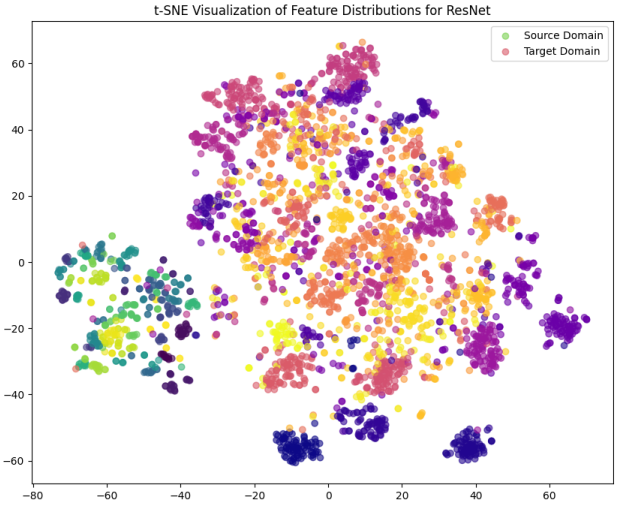

Figure 1. Plot showing feature distributions using TSNE to highlight domain shift in case of ResNet

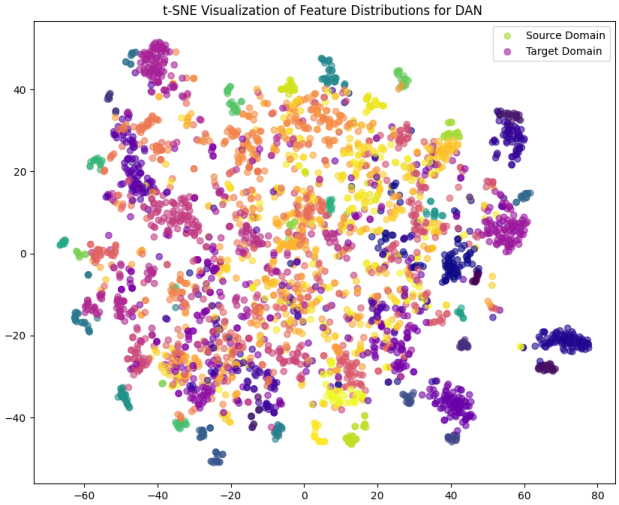

Figure 2. Plot showing feature distributions using TSNE to highlight domain alignment and successful adaptation in case of DAN

discrepancy between the feature distributions of the source and target domains.

In these models, the feature extractor network, typically based on pre-trained architectures like AlexNet is used to extract domain-invariant features. The MMD loss serves as a regularization term that measures the difference between the feature distributions of the two domains and penalizes any discrepancies. By minimizing this loss, the model encourages the learned features to be more domain-invariant, leading to improved generalization on the target domain. The effectiveness of this approach is demonstrated through better alignment in TSNE plots, where the features from a DAN show reduced domain shifts compared to baseline models, confirming the value of MMD-based adaptation strategies.

The figure depicting the MMD loss with respect to epochs shows a fluctuating but generally decreasing trend, while the model accuracy steadily increases and stabilizes over time. This relationship highlights the dynamic interplay between domain alignment and model learning during training. The fluctuations in MMD loss are common in domain adaptation tasks, as the model alternates between minimizing the discrepancy between source and target feature distributions and optimizing classification performance. Early in training, the feature representations are less aligned, leading to higher MMD values. As the model learns, MMD loss decreases, indicating better alignment, although fluctuations may still occur due to variations in mini-batches and the high-dimensional nature of the data.

The steady increase and eventual stabilization of accuracy suggest that the classifier becomes more confident and accurate as domain alignment improves. However, the fluctuations in MMD loss do not directly translate to accuracy fluctuations because MMD measures alignment between domains rather than classification performance. The decreasing trend in MMD loss complements the rising accuracy, as improved domain alignment facilitates better generalization on the target domain. The overall behavior demonstrates the effectiveness of the Deep Adaptation Network in simultaneously reducing domain discrepancy and enhancing classification performance.

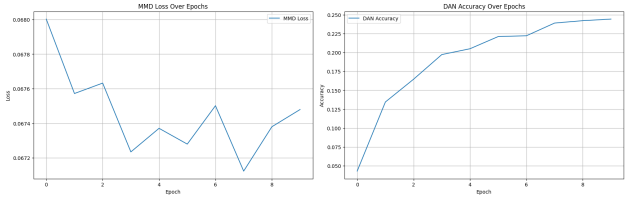

Figure 3. Plot showing MMD loss with respect to epochs and model accuracy with respect to epochs

1.3.5. DANN vs ResNet

The TSNE plot obtained from the DANN model demonstrates better domain alignment compared to the baseline ResNet. This improvement highlights DANN's ability to align the feature spaces of the source and target domains more effectively. In contrast, the ResNet exhibits a lack of domain alignment, resulting in overlapping and less distinguishable features between the two domains. The DANN's gradient reversal layer (GRL) helps enforce feature representations that are less sensitive to domain shifts, leading to clearer separability and more distinct clusters in the TSNE plot.

Moreover, the adversarial interplay between the feature ex-

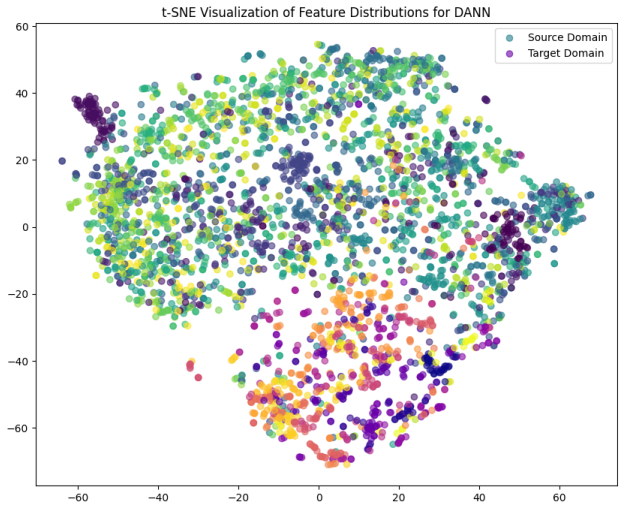

Figure 4. Plot showing feature distributions using TSNE to highlight domain alignment and successful adaptation in case of DANN

tractor and domain classifier in DANN provides insights into how the GRL facilitates domain alignment. It reverses the gradient during backpropagation, effectively discouraging the domain classifier from distinguishing between source and target domains. By learning domain-invariant features, the feature extractor aims to minimize the domain discrepancy. However, this process introduces potential trade-offs, including instability during training due to adversarial optimization. The adversarial nature of the DANN framework can lead to gradient conflicts, potentially causing slower convergence or oscillations in model performance. In some cases, these instabilities manifest as suboptimal adaptation, especially when hyperparameter settings, such as the adversarial weight or the strength of the GRL, are not carefully tuned.

Moreover, the GRL's contribution to domain alignment might result in a bias towards the most discriminative features rather than comprehensive domain-invariant ones. This can affect the model's generalization ability, potentially leading to underfitting or overfitting, depending on the dataset and domain shift characteristics. Therefore, careful tuning of hyperparameters like gamma (to fix it at **10**) and regularization parameters becomes crucial to mitigate these trade-offs and ensure effective domain alignment in DANN-based models. This usually depends on what the goal is, i.e. classification on target domain is more crucial than differentiating between the two domains.

2. Domain Generalization

Methodology

In this task we address the challenge of domain generalization using Invariant Risk Minimization (IRM) and its variants on the Rotated MNIST dataset. The primary objective is to evaluate the effectiveness of these methods in improving out-of-domain (OOD) generalization while maintaining in-domain (IID) accuracy. We create multiple training environments by applying rotations of varying angles to the MNIST dataset and use a simple multi-layer perceptron (MLP) as the backbone model. During training, three IRM-based methods are employed: (1) **IRM Basic**, which enforces invariance across environments by penalizing the gradients of logits; (2) **IB-IRM**, which extends IRM Basic by incorporating a variance regularization term to stabilize feature representations; and (3) **PAIR (Pareto IRM)**, which balances empirical risk minimization, IRM penalties, and variance regularization using a weighted combination. Training involves computing losses that combine cross-entropy, IRM penalties, and variance terms depending on the chosen method. This methodology highlights the ability of IRM-based approaches to identify invariant features across domains and balance trade-offs between in-domain performance and robustness to domain shifts.

2.1. Results

The results in Table 3 illustrate the IID (in-domain) and OOD (out-of-domain) accuracy across epochs for the three methods: IRM Basic, IB-IRM, and PAIR. Among the methods, IRM Basic consistently demonstrates superior performance, achieving the highest IID accuracy of 0.9009 and OOD accuracy of 0.5722 by epoch 5. This highlights its effectiveness in modeling both in-domain and out-of-domain data. IB-IRM achieves competitive performance, with a peak IID accuracy of 0.8649 and OOD accuracy of 0.4630. This indicates that while IB-IRM improves OOD generalization over some epochs, it still falls short of IRM Basic's robustness. PAIR, on the other hand, balances IID and OOD accuracy but achieves lower overall performance compared to the other methods, with an IID accuracy of 0.7865 and OOD accuracy of 0.3148 at its best. While PAIR demonstrates stability, it does not reach the generalization capabilities of IRM Basic. Overall, IRM Basic emerges as the best-performing method in this comparison, excelling in both IID and OOD accuracy and showcasing robust performance across all evaluated epochs.

2.2. Discussion

Based on the results and visualizations, the implementation of IRM Basic, IB-IRM, and PAIR demonstrates significant differences in their ability to generalize across source and

Epoch	IRM Basic (IID, OOD)	IB-IRM (IID, OOD)	PAIR (IID, OOD)
1	(0.7620, 0.2995)	(0.7377, 0.2270)	(0.7068, 0.1753)
2	(0.8154, 0.3756)	(0.8482, 0.4305)	(0.7504, 0.2469)
3	(0.8608, 0.5050)	(0.8611, 0.4630)	(0.7704, 0.3137)
4	(0.8864, 0.5328)	(0.8649, 0.4422)	(0.7865, 0.3148)
5	(0.9009, 0.5722)	(0.8604, 0.4545)	(0.7836, 0.2917)

Table 3. IID and OOD Accuracy Across Epochs for IRM Basic, IB-IRM, and PAIR Methods.

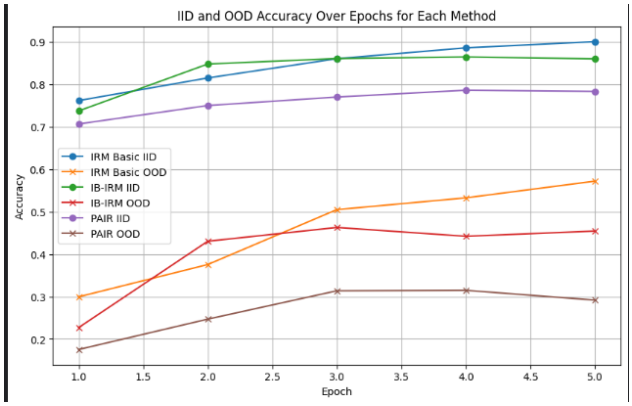

Figure 5. IID and OOD accuracy trends for IRM Basic, IB-IRM, and PAIR over epochs. IRM Basic shows superior performance, IB-IRM improves early but stabilizes, and PAIR offers a stable trade-off between IID and OOD accuracy.

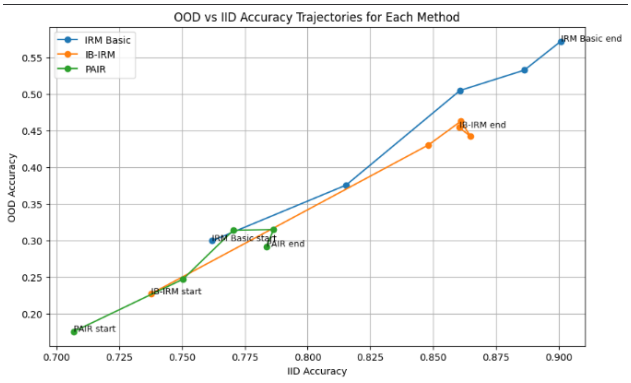

Figure 6. Trajectories of OOD versus IID accuracy for IRM Basic, IB-IRM, and PAIR methods. The plot highlights IRM Basic's strong linear improvement, IB-IRM's stabilization after early gains, and PAIR's consistent trade-off between IID and OOD performance.

target domains in an out-of-domain (OOD) setup. As shown in Figure 5, IRM Basic consistently achieves the highest OOD accuracy, culminating at 0.5722 by the fifth epoch, as detailed in Table 3. This highlights its robust capacity to simultaneously improve both in-domain (IID) and OOD performance. The steady upward trajectory in both IID and OOD accuracy for IRM Basic, depicted in Figure 6, emphasizes its capability to generalize while maintaining in-domain fidelity. This method outperforms IB-IRM and PAIR across all metrics, showcasing its effectiveness in handling domain shifts without sacrificing performance on the source domain.

IB-IRM, while competitive in earlier epochs, demonstrates diminishing returns in later stages. Specifically, its OOD accuracy peaks at 0.4630 by epoch 3, as shown in Table 3, but declines slightly thereafter. This is likely attributed to the regularization introduced by the variance penalty, which, while beneficial for disentangling invariant features, may over-constrain the model and hinder its adaptability to unseen domains. As depicted in Figure 6, the trajectory of IB-IRM exhibits a less consistent improvement in OOD accuracy as IID accuracy increases, highlighting the trade-offs introduced by its regularization mechanism. Although IB-IRM achieves higher initial OOD accuracy compared to PAIR, its performance stabilizes or even slightly deteriorates, indicating challenges in maintaining a balance between the two objectives.

PAIR, on the other hand, exhibits a more stable and gradual improvement in both IID and OOD accuracy. As shown in Table 3, its OOD accuracy reaches 0.3137 by epoch 3, but does not match the peak performance of IRM Basic or IB-IRM. However, as evidenced by Figure 6, PAIR demonstrates a smoother trajectory, reflecting a more consistent balance between IID and OOD accuracy over epochs. This stability can be attributed to its weighted combination of penalties, which provides a trade-off between empirical risk minimization (ERM), invariant risk minimization (IRM), and variance regularization. Despite its lower peak performance, the stability of PAIR makes it a suitable choice in scenarios where consistent performance across epochs is more desirable than achieving the highest OOD accuracy.

Analyzing the trade-offs between IID and OOD accuracy reveals distinct patterns for each method. As shown in Figure 6, IRM Basic exhibits a near-linear relationship between IID and OOD accuracy, indicating that improvements in in-domain performance directly translate to better generalization on the target domain. This contrasts with IB-IRM, which shows a less predictable trend, with OOD accuracy stabilizing or declining after initial gains. The results under-

score the challenges posed by strong regularization, which, while designed to improve invariance, can inadvertently limit the model's capacity to generalize effectively. PAIR, with its more gradual trajectory, offers a balanced trade-off but does not reach the same level of generalization as IRM Basic.

In conclusion, IRM Basic emerges as the best-performing method, excelling in both IID and OOD accuracy, as demonstrated by its consistent improvement across epochs in Figure 6 and its superior performance metrics in Table 3. IB-IRM, while effective in early epochs, is limited by the constraints imposed by its regularization strategy. PAIR provides a stable alternative, striking a balance between IID and OOD accuracy but falling short in overall performance. These findings emphasize the importance of method selection based on the specific requirements of the application, such as prioritizing peak performance (IRM Basic) versus stability (PAIR) or regularization effects (IB-IRM). The results provide valuable insights into the trade-offs inherent in domain generalization methods, highlighting the need for a nuanced approach to selecting and tuning these techniques for optimal performance.

3. Disentanglement Representation Learning

3.1. Methodology

Disentangled Representation Learning (DRL) focuses on extracting interpretable and independent latent factors from data. This task employs β-VAE as the primary method for learning disentangled representations by modifying the standard VAE objective to encourage factor separation. The methodology involves preparing datasets, defining the model architecture, training the model, and evaluating its performance. We used the *dSprites* dataset, which is composed of simple geometric shapes with generative factors such as size, position, and orientation, for evaluating disentanglement.

The β-VAE introduces a hyperparameter β in the loss function to promote disentanglement. The loss function is defined as:

$$L_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta \cdot KL(q_\phi(z|x)\|p(z)),$$

where the encoder maps input data x to a latent space z, and the decoder reconstructs x from the latent variables z. The KL divergence term regularizes $q_\phi(z|x)$ to align with the prior $p(z)$, ensuring disentanglement when $\beta > 1$. The β-VAE is trained on the selected dataset using different values of β (e.g., $\beta = 1, 5, 10, \ldots$) with an encoder-decoder architecture and standard hyperparameters. During training, reconstruction loss and disentanglement metrics are monitored to ensure convergence and stability.

To evaluate the disentanglement performance, latent factor visualization is performed by generating samples where one latent dimension is varied while keeping others fixed. This approach helps in assessing the correspondence between latent dimensions and generative factors, such as size, position, and style. Quantitative metrics, such as Z-diff or metrics from libraries like *DisentanglementLib*, are used to measure disentanglement and compare results across different β values and the baseline VAE. Additionally, qualitative analysis is conducted by inspecting the quality of generated images and assessing the model's ability to capture independent generative factors. Reconstructions are visualized to identify any loss of fine-grained details or mode collapse.

The insights derived from this methodology include analyzing the impact of β on disentanglement and reconstruction quality. Trade-offs between interpretability and fidelity are highlighted, and failure cases where β-VAE does not effectively disentangle factors or leads to degraded reconstructions are discussed. This comprehensive methodology ensures a systematic evaluation of disentanglement methods, providing valuable insights into achieving a balance between interpretability and fidelity in latent representations.

3.2. Discussion

3.3.1. Visualizing disentangled latent factors

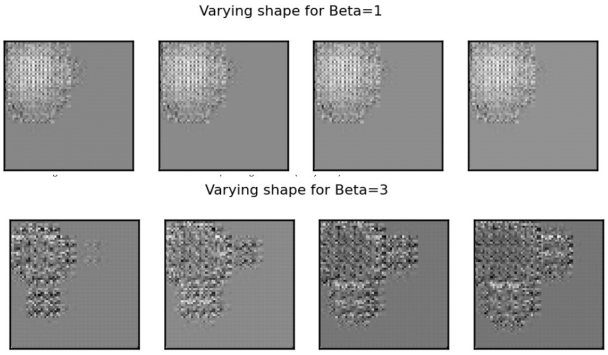

Figure 7. Plot showing the impact of increasing the generating factor / latent variable 'shape' for different Beta values in Beta-VAE architecture

The figure demonstrates how the disentanglement of the latent variable 'shape' evolves with increasing β in the Beta-VAE architecture. For $\beta = 1$, the shape remains consistent, indicating that the latent representation does not effectively disentangle this factor. However, for $\beta = 3$, there is a notable increase in the variability of the shape representation, suggesting that a higher β value encourages the model to allocate more capacity in the latent space for disentangling and explicitly representing the 'shape' factor. This highlights the role of the β parameter in balancing reconstruction fidelity and disentanglement, with larger values emphasizing the latter.

3.3.2. Evaluating Disentanglement

Mutual Information Gap (MIG) is a quantitative metric used to evaluate disentangled representations, particularly in generative models such as Variational Autoencoders (VAEs). It was introduced in the **Beta-TCVAE** paper to measure the degree of disentanglement achieved by these models.

The primary goal of disentanglement is to learn latent representations where each dimension corresponds to a distinct and identifiable factor of variation, such as color, shape, or position. MIG quantifies how well these dimensions capture independent factors of variation by computing the mutual information between the latent representations and the true factors. A higher MIG score indicates that each dimension of the latent space corresponds to a specific factor of variation, while a lower score suggests entanglement, where multiple factors contribute to a single latent dimension. As such, MIG provides an interpretable and direct measure of how well a model has learned disentangled representations.

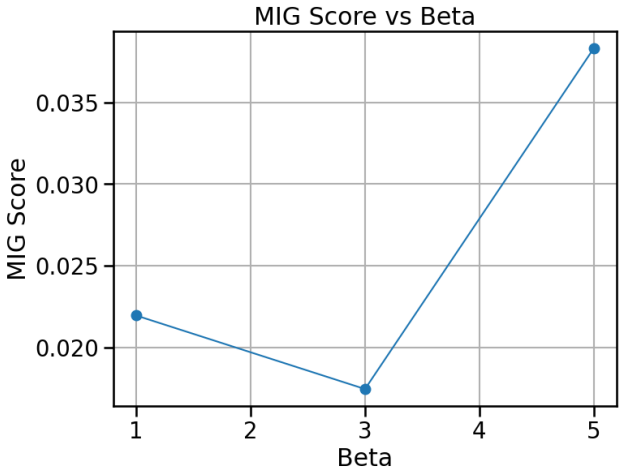

Figure 8. Plot showing how MIG score varies with Beta

The plot above shows that MIG score initially decreasing for a minimal amount, then increases considerably, showing quantitatively that disentanglement is occuring as we increase the weight of KL divergence term. This trend indicates that as the weight of the KL divergence term increases, the model initially struggles to balance reconstruction quality and latent space regularization. However, with further increase, the model achieves better disentanglement by effectively aligning the latent representations with the underlying generative factors, thereby improving the Mutual Information Gap. This demonstrates the trade-off between disentanglement and reconstruction during training and highlights the importance of tuning the beta parameter for optimal results.

3.3.3. Effect of varing beta on disentanglement and reconstruction quality

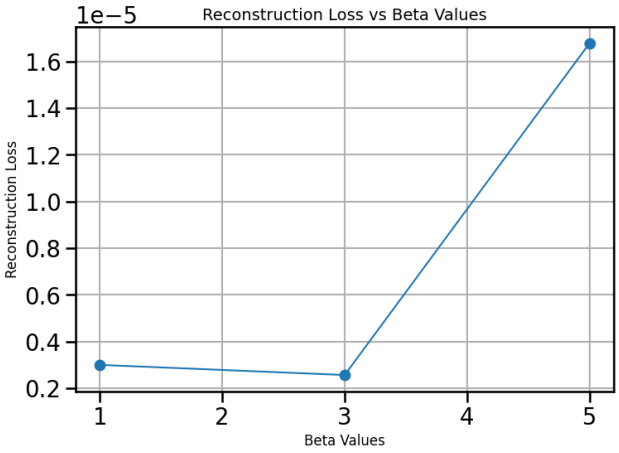

Figure 9. Plot showing how Reconstruction Loss varies with Beta

The plot above shows that Reconstruction Loss score initially decreasing for a minimal amount, then increases considerably, showing quantitatively that the model starts focusing more on disentanglement than reconstruction as we increase the weight of KL divergence term. This behavior reflects the inherent trade-off between reconstruction quality and disentanglement in variational models. As the weight of the KL divergence term increases, the model prioritizes enforcing structure in the latent space over accurately reconstructing the input data. This leads to a shift in focus, where the model sacrifices some reconstruction fidelity to achieve better disentanglement, aligning the latent representations more closely with the underlying generative factors.

3.3.4. Limitations of Beta VAE

The β-VAE introduces a weighting factor β to the standard VAE objective, encouraging disentanglement by penalizing the Kullback-Leibler (KL) divergence term more heavily. However, this adjustment comes with significant limitations.

One of the primary limitations is the loss of reconstruction quality. By increasing β, the model prioritizes disentanglement of latent representations at the expense of accurately reconstructing input data. This is particularly evident when β becomes too large, where the latent space is overly constrained, causing the decoder to struggle in generating high-fidelity outputs.

Another limitation lies in its inability to disentangle specific factors in certain scenarios. For datasets with highly correlated or complex generative factors, β-VAE may fail to separate these factors into independent latent dimensions, leading to entangled representations. This is often observed in datasets where factors such as shape, position, and texture are highly intertwined.

Furthermore, β-VAE often suffers from instability during

training, particularly as β increases. The optimization process becomes sensitive to hyperparameter selection, and achieving a balance between disentanglement and reconstruction requires careful tuning, which is computationally expensive.

Lastly, β-VAE is limited in scalability to datasets with complex, high-dimensional generative factors. While it performs well on simpler datasets like dSprites, its effectiveness diminishes on real-world datasets with intricate variations, such as CelebA, where factors like lighting, pose, and facial features interact non-linearly.

Conclusion

This report highlights the distinctions among the methods employed for Unsupervised Domain Adaptation (UDA) and Domain Generalization (DG). For UDA, Domain Adaptation Network (DAN) and Domain Adversarial Neural Network (DANN) were effective in reducing the domain gap. DAN utilized Maximum Mean Discrepancy (MMD) for domain alignment, while DANN relied on adversarial learning to enforce domain invariance. DANN demonstrated greater adaptability to complex shifts but required careful tuning to avoid training instabilities. Both methods performed well, though their effectiveness depended heavily on dataset characteristics and hyperparameter configurations.

In DG tasks, IRM Basic emerged as the most robust method, achieving consistently high out-of-domain (OOD) accuracy while maintaining in-domain (IID) performance. IB-IRM showed early improvements in OOD accuracy but suffered diminishing returns in later epochs due to over-regularization. PAIR, while not achieving the highest peak performance, offered a stable balance between IID and OOD accuracy. Overall, IRM Basic and DAN excel in scenarios requiring high accuracy under significant domain shifts, while PAIR and DANN provide more stable alternatives when consistent performance across domains is prioritized. These findings emphasize the importance of aligning method selection with specific task requirements and domain complexities.

Contributions

Task	Done by
task 1	Adeen and Huraira
task 2	Adeen
task 3	Huraira

References

1. Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

2. Isabela Albuquerque, João Monteiro, Tiago H Falk, and Ioannis Mitliagkas. Adversarial target-invariant representation learning for domain generalization. *CoRR*, 2019.

3. Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision*, pages 213–226. Springer, 2010.

4. Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

5. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

6. Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

7. Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, volume 2011, page 4. Granada, 2011.

8. Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015.

9. Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015.

10. Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2551–2559, 2015.

11. Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.

12. Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

13. Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

14. Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021.

15. Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Bingzhe Wu, Yonggang Zhang, Kaili Ma, Han Yang, Peilin Zhao, Bo Han, et al. Pareto invariant risk minimization: Towards mitigating the optimization dilemma in out-of-distribution generalization. *arXiv preprint arXiv:2206.07766*, 2022.

16. Xin Wang, Hong Chen, Zihao Wu, Wenwu Zhu, et al. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

17. Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15(2018):11, 2018.

18. Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3769, 2014.

19. Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

20. Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.