Disentangled Embedding through Style and Mutual Information for Domain Generalization

Anonymous authors
Paper under double-blind review

Abstract

Deep neural networks often experience performance degradation when faced with distributional shifts between training and testing data, a challenge referred to as domain shift. Domain Generalization (DG) addresses this issue by training models on multiple source domains, enabling the development of invariant representations that generalize to unseen distributions. While existing DG methods have achieved success by minimizing variations across source domains within a shared feature space, recent advances inspired by representation disentanglement have demonstrated improved performance by separating latent features into domain-specific and domain-invariant components. We propose two novel frameworks: Disentangled Embedding through Mutual Information (DETMI) and Disentangled Embedding through Style Information (DETSI). DETMI enforces disentanglement by employing a mutual information estimator, minimizing the mutual dependence between domain-agnostic and domain-specific embeddings. DETSI, on the other hand, achieves disentanglement through style extraction and perturbation, facilitating the learning of domain-invariant and domain-specific representations. Extensive experiments on the PACS, Office-Home, and VLCS datasets show that both frameworks outperform several state-of-the-art DG techniques.

1 Introduction

Deep neural networks excel at learning discriminative features and achieve outstanding performance in computer vision tasks. However, their performance deteriorates significantly when tested on data with a distribution different from the training data Ben-David et al. (2010). For instance, a network trained on urban environment images performs poorly when applied to rural environments. This performance drop results from the domain shift Quiñonero-Candela et al. (2022), which refers to the difference between source and target data distributions. Domain Adaptation (DA) Farahani et al. (2021) addresses this challenge by leveraging labeled or unlabeled target domain data alongside source domain data to improve generalization. DA assumes the availability of some labeled or unlabeled data from a specific target domain, which, combined with source domain data, enhances the network's generalization ability. Nevertheless, the increasing complexity of real-world tasks makes collecting data from every potential target domain impractical.

Introduced in 2011 Blanchard et al. (2011), Domain Generalization (DG) proposes training a model on multiple relevant source domains to develop invariant representations that generalize to new distributions not present during training. Over the past 12 years, numerous approaches have been proposed to address the challenge of DG. Among these, adversarial training has been widely explored to encourage domain-invariant feature learning across different domains Li et al. (2018b;c); Sinha et al. (2017). Other strategies have employed meta-learning Li et al. (2018a); Balaji et al. (2018); Zhao et al. (2021); Finn et al. (2017), which enables models to learn to generalize by simulating domain shifts during training.

Data augmentation techniques have also emerged as promising solutions, where synthetic data is generated to expose the model to a broader range of variations, helping it generalize to unseen domains Mehmood & Barner (2024); Volpi et al. (2018); Zhou et al. (2020a;b). Additionally, some methods focus on learning domain-invariant features, such as shape representations, by minimizing the divergence between latent representations across domains, thereby improving generalization to new environments Zhou et al. (2022); Lu et al. (2022). These approaches represent key advancements in DG, each contributing unique mechanisms to tackle the distributional shifts that lead to domain shifts.

Most of these methods discard domain-specific information (*e.g,* background, style) to minimize the divergence between the embedding from different source domains. Recently, inspired by disentangled representation learning, the authors of POEM Jo & Yoon (2023) trained a model to learn domain invariant and domain-specific information separately. In addition, to remove redundant information between these two feature spaces, cosine similarity was minimized, further enhancing disentanglement.

Inspired by disentangled embedding learning for domain generalization (DG), we address the domain generalization problem by employing two encoders to separately capture domain-specific and domain-agnostic embeddings. To achieve this, we introduce two novel frameworks designed to effectively learn these representations.

In our first approach, inspired by POEM Jo & Yoon (2023), we recognize that while cosine similarity promotes linear independence between domain-specific and domain-invariant feature spaces by considering their spatial arrangement, it does not fully eliminate higher-order dependencies. To address this limitation, we propose the DETMI framework, which minimizes mutual information between the two feature spaces. By reducing higher-order correlations, DETMI ensures statistical independence, enabling the learning of richer disentangled embeddings and enhancing generalization to unseen domains.

In our second approach, we introduce the DETSI framework, which leverages style information to learn domain-specific and domain-agnostic embeddings. Inspired by Huang & Belongie (2017), which establishes a correlation between the domain of an image and its style, we apply style perturbation to encourage the extraction of domain-agnostic embeddings. These embeddings prioritize high-level features, such as object structure and semantic content, rather than domain-dependent attributes. Meanwhile, for domain-specific embeddings, DETSI captures style-specific characteristics, including texture and artistic style, instead of low-level features like edges or pixel intensity. By explicitly modeling style-based attributes, DETSI effectively disentangles domain-specific components from domain-invariant ones and produces robust embeddings, significantly enhancing generalization across diverse unseen domains.

We conducted extensive experiments on the PACS Li et al. (2017), Office-Home Venkateswara et al. (2017), and VLCS Torralba & Efros (2011) benchmarks. The results demonstrate that the proposed approaches outperform several state-of-the-art (SOTA) methods addressing Domain Generalization.

The remainder of this paper is organized as follows. Section 2 reviews related work, summarizing key methods developed to address the challenges of domain generalization. Section 3 details the proposed methodologies, including the techniques used to tackle domain generalization. Section 4 describes the experimental setup, covering datasets, implementation details, and the results of the proposed methods. Section 5 presents ablation studies, providing deeper insights into the contributions of individual components. Finally, Section 6 concludes the paper by summarizing the findings and discussing their implications.

2 Related Work

Domain Generalization (DG) Zhou et al. (2020a) is an approach that trains models using labeled data from multiple source domains to generalize to an unseen target domain effectively. This problem arises naturally in applications such as medical imaging, autonomous driving, and visual recognition, where the distribution of data in the real world may differ significantly from that in the training set, leading to poor generalization performance. Over the past decade, numerous techniques have been proposed to tackle domain generalization, including domain-invariant feature learning, which focuses on extracting representations that remain consistent across domains; meta-learning, which leverages learning-to-learn paradigms for better adaptability; data augmentation, which generates diverse training examples to improve model robustness; adversarial learning, which introduces adversarial objectives to align distributions across domains; and disentangled embedding learning, which separates domain-specific and domain-invariant factors to enhance generalization. Each of these approaches is discussed in detail in the following subsections.

2.1 Domain-Invariant Representation Learning

One of the most popular strategies in domain generalization is to learn domain-invariant representations, where features are shared across different domains and remain stable under distributional shifts. Early works in this area have leveraged feature alignment techniques to ensure that representations of different domains become indistinguishable in a learned latent space. For example, Muandet *et al.* Muandet et al. (2013) introduced Domain-Invariant Component Analysis (DICA), where they aimed to learn a feature transformation that removes domain-specific variations while retaining the information necessary for classification. Similarly, Ghifary *et al.* proposed Scatter Component Analysis (SCA) Ghifary et al. (2016), a method that projects data into a subspace where the variance between domains is minimized and the class separation is maximized.

Another approach uses distribution matching techniques like Maximum Mean Discrepancy (MMD) to align the feature distributions across domains. For example, Li *et al.* Li et al. (2018b) proposed the Domain-Adversarial Neural Network (DANN) framework, where an adversarial loss is used to align feature distributions between domains by confusing a domain classifier that attempts to distinguish between source domains.

2.2 Meta-Learning

Meta-learning, often called "learning to learn," has recently gained traction as a promising framework for domain generalization. Meta-learning approaches Li et al. (2018a); Zhao et al. (2021); Finn et al. (2017) aim to train a model to quickly adapt to new, unseen domains. In this method, data from source domains is divided into meta-train and meta-test segments, allowing the model to be trained specifically to excel on the meta-test data using the meta-train data, mimicking real-world applications where the model must adapt to completely new data.

Li *et al.* Li et al. (2018a) introduced a meta-learning framework for domain generalization in which the model is trained on multiple source domains to simulate the process of generalizing to new, unseen domains. Specifically, the model is trained in a meta-learning loop, where each training iteration mimics the domain generalization process by exposing the model to different domain shifts, helping the model learn robust features that generalize well to unseen domains.

Another key work is by Balaji *et al.* Balaji et al. (2018), who proposed MetaReg, where a meta-regularization term is learned to guide the model's parameters to be domain-agnostic. These approaches focus on teaching the model to adapt quickly to new tasks by simulating domain shifts during training. Meta-learning approaches have shown great promise in domain generalization due to their ability to simulate and adapt to new domain distributions in the training phase.

2.3 Adversarial Learning

Adversarial learning has also been widely used in domain generalization to reduce the gap between source and unseen target domains. The core idea is to learn indistinguishable feature representations across domains using adversarial training techniques.

In one of the foundational works, Li *et al.*Li et al. (2018b) proposed an adversarial autoencoder (AAE), where the encoder is adversarially trained to produce features that are domain-invariant, while a domain discriminator is trained to distinguish between features from different domains. The feature extractor tries to fool the domain discriminator, thus learning domain-invariant representations.

In another work, Li *et al.* Li et al. (2018c) train a conditional invariant adversarial network to learn domain-invariant representations by making the learned representations on different domains indistinguishable through adversarial training.

Adversarial approaches have proven to be effective in learning representations that are robust to domain shifts, though they often require careful tuning of the adversarial loss function.

2.4 Data Augmentation

In addition to improving DG, data augmentation techniques Zhou et al. (2020a;b); Li et al. (2023) introduce more variety to training data by augmenting existing data pairs (x, y), where x represents the input and y the corresponding label. These techniques generate transformed pairs $(A(x), y)$, where $A(\cdot)$ is a transformation that preserves the original label. This process helps to prepare the model to handle the diverse conditions encountered in the source domains.

Volpi *et al.* Volpi et al. (2018) proposed an augmentation strategy based on adversarial perturbations, where synthetic examples are generated by perturbing the original data to mimic potential unseen domains. The model can better handle domain shifts at test time by training on these perturbed examples.

Shankar *et al.* Shankar et al. (2018) introduced CrossGrad, an approach that uses the gradient of the domain classifier to perturb input examples. This ensures that the generated examples lie closer to the decision boundary of the domain classifier, forcing the model to learn domain-invariant features.

Zhou *et al.* Zhou et al. (2020b) proposed a data augmentation approach that utilizes a data generator to synthesize samples from pseudo-novel domains, effectively expanding the source domain with artificially generated data. By creating these domain variations, the model is exposed to a wider range of potential domain shifts, enhancing its ability to generalize to unseen target domains.

In another work, Zhou *et al.* Zhou et al. (2020a) developed a Deep Domain-Adversarial Image Generation (DDAIG) network to generate more synthetic data using adversarial training, increasing domain diversity, and improving the model's overall generalization capabilities. Other techniques like Domain Randomization (DR) Tobin et al. (2017) and image transformations, which adjust visual features such as color, texture, and lighting, add further robustness.

Data augmentation methods have been particularly effective when the goal is to simulate diverse, unseen domains during training. However, a challenge with these methods is ensuring that the generated augmentations accurately reflect the domain shifts encountered in practice.

2.5 Disentangled Embeddings Learning

Inspired by representation learning, recent advances in disentangled embedding learning for domain generalization have increasingly focused on partitioning the latent space into domain-invariant and domain-specific embeddings. Researchers, including Bui et al. (2021) and Jo & Yoon (2023), minimize metrics such as covariance or cosine similarity between these spaces to enforce greater independence. Using a similar intuition, Yu et al. (2024), minimize KL divergence loss to learn domain-agnostic and domain-specific embeddings through a single encoder. This separation enhances the model's generalization ability across diverse and unseen domains.

Our proposed frameworks tackle DG by disentangling latent representations into domain-specific and domain-invariant components. We leverage domain-specific information to aid in the learning of domain-invariant semantic features, ensuring a more robust representation. We leverage mutual information and style information in DETMI and DETSI, respectively, to effectively capture both domain-related and domain-agnostic embeddings. Through this disentanglement, our proposed approaches improve the model's ability to learn well-separated representations, leading to improved generalization across diverse domains.

3 Proposed Method

The feature and label spaces are represented by $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{Y} \subset \mathbb{R}$, respectively. A domain is represented by a joint distribution $P_{xy} \in \mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$, where $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$ denotes the set of joint probability distributions on $\mathcal{X} \times \mathcal{Y}$. In DG, we have access to K similar but distinct source domains as $S^k = (x_i^{(k)}, y_i^{(k)})_{i=1}^{N^{(k)}}$, each associated with a joint distribution $P_{xy}^{(k)}$, where $(x_i^{(k)}, y_i^{(k)}) \sim P_{xy}^{(k)}$ and $N^{(k)}$ denotes the total number of data points in a particular domain k. The goal of DG is to learn a model $f : \mathcal{X} \to \mathcal{Y}$ using source domain data such that it can generalize well on an unseen target domain τ having joint distribution P_{xy}^{τ}.

The next two subsections introduce frameworks for disentangling embeddings to improve domain generalization. The first subsection, *Disentangled Embedding Through Mutual Information (DETMI)*, presents a framework that minimizes mutual information between domain-specific and domain-invariant components to achieve statistical independence and enhance the learning of generalizable features. This approach leverages neural network estimators to calculate and optimize mutual information, enabling the acquisition of semantically meaningful disentangled embeddings. The second subsection, *Disentangled Embedding Through Style Information (DETSI)*, proposes a complementary strategy that utilizes style information, such as feature statistics and Gram matrices, to isolate domain-specific and domain-agnostic features. This framework promotes the separation of domain-relevant components by focusing on content and style perturbations, further improving the model's ability to generalize across unseen domains.

3.1 Disentangled Embedding through Mutual Information (DETMI)

DG assumes the invariance between $\mathcal{X}$ and $\mathcal{Y}$. Existing methodologies primarily focus on acquiring invariant features from available source domains, anticipating that these invariant features can be extended to predict the target domains unseen during training. However, these approaches remain susceptible to erroneous predictions due to their inability to handle variations caused by data bias. Drawing inspiration from disentangled embedding learning Bengio et al. (2013), which delves into the underlying factors of variation in the data and fosters robust learning of task-specific features, we endeavor to disentangle domain-specific and domain-agnostic features.

Domain-specific features, including background, style variation, and location, contribute to data bias, impacting a model's prediction performance. By disentangling domain-specific features from domain-invariant ones, we augment our model's performance towards unseen targets by acquiring generalizable features.

To this end, we propose the Disentangled Embeddings Through Mutual Information (DETMI) framework illustrated in Fig. 1. In the framework, we train two encoders, E_c and E_d, to learn category-related and domain-specific features denoted as Z_c and Z_d, respectively. We also employ two classifiers, C and $\tilde{C}$, to predict class labels and domain labels, respectively, leveraging Cross-Entropy loss:

$$\mathcal{L}_c = -y_i^k \log\big(\sigma(C(E_c(x_i^k;\theta_c);\delta_c))\big), \tag{1}$$

and

$$\mathcal{L}_d = -d_i^k \log\big(\sigma(\tilde{C}(E_d(x_i^k;\theta_d);\delta_d))\big), \tag{2}$$

where y_i^k and d_i^k denote class and domain labels, respectively.

The optimization objective function for learning domain-agnostic feature space through the leverage of domain-specific information is:

$$\mathcal{L}_{dI} = \mathcal{L}_c + \lambda_1 \mathcal{L}_d. \tag{3}$$

where λ_1 is the weighting coefficient.

Although we leverage domain-specific features to learn only domain-agnostic features, this constraint is insufficient to remove all domain-relevant information. A recent study Jo & Yoon (2023) tackled this challenge by employing cosine similarity to encourage geometric independence between these feature spaces, thus improving disentanglement. Our approach minimizes mutual information between Z_c and Z_d to enforce independence, addressing higher-order correlations beyond the mere spatial arrangement.

3.1.1 Independence through Mutual Information

Let X and Y be two random variables. Mutual information $I(X;Y)$ measures the statistical dependence between two variables.

Theorem 1. If $I(X;Y) = 0$, then X and Y are statistically independent.

Proof. The mutual information between two random variables X and Y is given by:

$$I(X;Y) = \iint P(x,y) \log \frac{P(x,y)}{P(x)P(y)}\, dx\, dy. \tag{4}$$

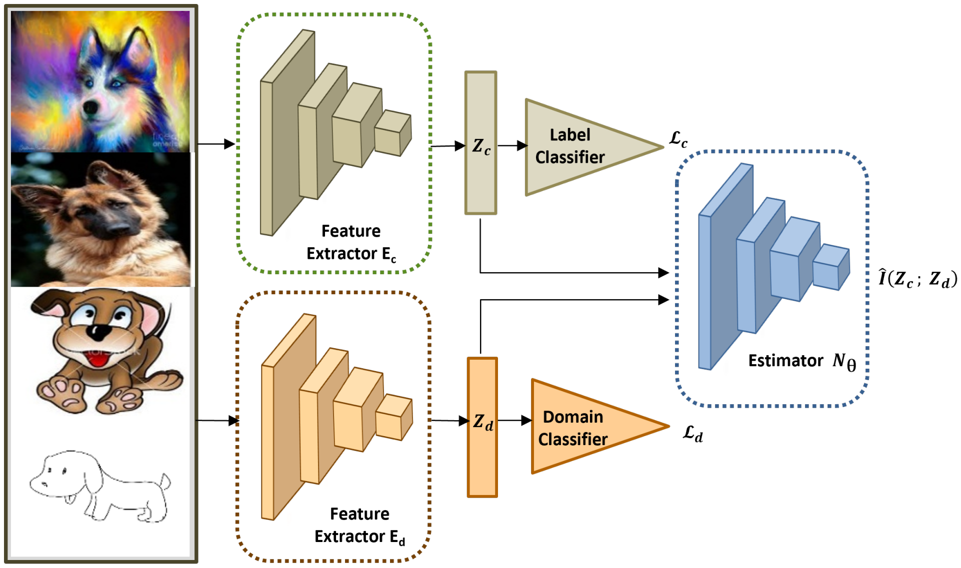

Figure 1: DETMI: Disentangled Embedding Through Mutual Information

If $I(X;Y) = 0$, then:

$$\log \frac{P(x,y)}{P(x)P(y)} = 0 \Rightarrow P(x,y) = P(x)P(y). \tag{5}$$

In light of Theorem 1, a well-known result in information theory, reducing $I(X;Y)$ brings $P(x,y)$ closer to $P(x)P(y)$, promoting statistical independence. This facilitates the learning of semantically meaningful disentangled embeddings, enhancing the extraction of generalizable features for improved domain adaptation.

The mutual information between Z_c and Z_d is given by:

$$I(Z_c; Z_d) = \iint p(\tilde{z}, \hat{z}) \log \left(\frac{p(\tilde{z}, \hat{z})}{p(\tilde{z})p(\hat{z})} \right) d\tilde{z}\, d\hat{z} \tag{6}$$

where $p(\tilde{z}, \hat{z})$ is the joint probability density function of Z_c and Z_d, and $p(\tilde{z})$ and $p(\hat{z})$ are the marginal probability density functions of Z_c and Z_d, respectively. For high-dimensional variables, the calculation of the double integrals is quite complex.

MINE Belghazi et al. (2018), utilizes the Donsker-Varadhan representation of the Kullback-Leibler (KL) divergence to provide a lower bound on mutual information and optimize it using a neural network.

Using Donsker-Varadhan representation, the Kullback-Leibler (KL) divergence between two probability distributions P and Q is given by:

$$D_{\mathrm{KL}}(P||Q) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_P[T] - \log \mathbb{E}_Q[e^T]. \tag{7}$$

where the supremum is taken over all functions T for which both expectations remain finite.

The mutual information $I(X;Y)$ between two random variables X and Y is defined using the Kullback-Leibler (KL) divergence as:

$$I(X;Y) = D_{\mathrm{KL}}(P(X,Y)||P(X)P(Y)), \tag{8}$$

where $P(X,Y)$ is the joint probability distribution of X and Y. $P(X)P(Y)$ represents the product of marginal distributions.

Following MINE, we employ a neural network N_θ to estimate the lower bound of $I(Z_c; Z_d)$:

$$I(Z_c; Z_d) \geq \hat{I}(Z_c; Z_d) = \sup_\theta \mathbb{E}_{p(\tilde{z},\hat{z})}[N_\theta] - \log \mathbb{E}_{p(\tilde{z})\otimes p(\hat{z})}[e^{N_\theta}]. \tag{9}$$

The expectations in (9) are computed using the approach of MINE. Accordingly,

$$\hat{I}(Z_c; Z_d) = \frac{1}{n}\sum_{i=1}^{n} N(\tilde{z}_i, \hat{z}_i, \theta) - \log\left(\frac{1}{n}\sum_{i=1}^{n} \exp^{N\left(\tilde{z}_i, \tilde{\hat{z}}_i, \theta\right)}\right), \tag{10}$$

where $(\tilde{z}, \hat{z})$ is obtained through the joint probability density function $p(\tilde{z}, \hat{z})$ and $\tilde{\hat{z}}$ is obtained through the marginal distribution $p(\hat{z})$ by a random shuffle.

By minimizing $\hat{I}(Z_c; Z_d)$, we force the learned embeddings Z_c and Z_d to satisfy:

$$P(Z_c, Z_d) \approx P(Z_c)P(Z_d). \tag{11}$$

This prevents information leakage from Z_d into Z_c, ensuring proper disentanglement. The final optimization objective function for learning domain-agnostic feature space through the leverage of domain-related information is:

$$\mathcal{L}_{dI} = \mathcal{L}_c + \lambda_1 \mathcal{L}_d + \lambda_2 \hat{I}(Z_c; Z_d). \tag{12}$$

where λ_1 and λ_2 are the weighting coefficients.

Algorithm 1 Training procedure for DETMI

1: **Input:** K domains data samples; Encoders E_c, E_d; Classifiers C, $\tilde{C}$; Mutual information estimator N_θ; Batch size B
2: **Output:** Optimized Encoder E_c and Classifier C
3: Using MINE, Update Mutual information estimator N_θ by maximizing (5) until convergence.
4: **for** $i = 1$: epochs **do**
5: Sample a mini-batch $(x_i^{(k)}, y_i^{(k)})_{i=1}^{B} \in S^k$
6: Compute the objective function
7: $\mathcal{L}_{dI} = \mathcal{L}_c + \lambda_1 \mathcal{L}_d + \lambda_2 \hat{I}(Z_c; Z_d)$.
8: Update E_c, E_d, C, $\tilde{C}$
9: **end for**
10: **Return** Optimized Encoder E_c and Classifier C

3.2 Disentangled Embedding through Style Information (DETSI)

We propose DETSI, an alternative approach for learning disentangled embeddings, which leverages style information to learn domain-agnostic and domain-specific embeddings, denoted as Z_c and Z_d. Building on the work of Huang & Belongie (2017), which establishes the close relationship between style and instance-level feature statistics, and recognizing the strong correlation between image style and visual domains, we employ style perturbation to encourage content-focused learning for domain-agnostic embeddings. Additionally, we extract style features to enhance the learning of domain-specific information, enabling a more effective separation of domain-specific and domain-invariant components.

3.2.1 Preliminaries

Huang & Belongie (2017) demonstrated that CNN convolutional features maps statistics i.e. channel-wise mean and variance, effectively characterize image style. Building on this insight, Ulyanov et al. (2016) proposed Instance Normalization (IN) to normalize these style statistics and mitigate style variations in style transfer models.

For a given input image x, its feature maps are represented as $f_x \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of channels, and H and W correspond to the spatial dimensions. The formulation of Instance Normalization (IN) is expressed as:

$$\mathrm{IN}(f_x) = \gamma \frac{f_x - \mu_f}{\sigma_f} + \beta, \tag{13}$$

where $\gamma, \beta \in \mathbb{R}^C$ correspond to learnable affine transformation parameters, and $\mu_f, \sigma_f \in \mathbb{R}^C$ correspond to the channel-wise mean and standard deviation.

$$\mu_f = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} f_{c,h,w}, \tag{14}$$

and

$$\sigma_f = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (f_{c,h,w} - \mu_f)^2 + \epsilon}, \tag{15}$$

where a small constant ϵ is added to avoid numerical instability.

Furthermore, leveraging these style statistics, Huang & Belongie (2017) introduced Adaptive Instance Normalization (AdaIN), which transfers an image's style to a target style by replacing the affine parameters with corresponding style-specific statistics (μ_s, σ_s). AdaIN is defined as:

$$\mathrm{AdaIN}(f_x, s) = \sigma_s \frac{f_x - \mu_f}{\sigma_f} + \mu_s. \tag{16}$$

In this paper, we introduce perturbations to the channel-wise mean and standard deviation of each feature map to introduce style randomization. Furthermore, we utilize AdaIN to substitute the original style information with randomly generated style statistics, enhancing style invariance in our feature representations.

3.2.2 Content-focused Learning

We achieve content-focused learning by introducing style perturbation, which alters the feature statistics at the instance level of the training images. These feature statistics, closely tied to style, are perturbed to encourage the model to focus on content rather than stylistic variations. The process is formally defined by:

$$\mu_{new} = \lambda \mu(f_c^{(l)}) + (1-\lambda)\mu(\tilde{f}_c^{(l)}), \tag{17}$$

and

$$\sigma_{new} = \lambda \sigma(f_c^{(l)}) + (1-\lambda)\sigma(\tilde{f}_c^{(l)}), \tag{18}$$

where $\lambda \sim \mathcal{U}(0,1)$. Here, $f_c^{(l)}$ represents a specific batch of feature maps at a layer l of the encoder E_c, and $\tilde{f}_c^{(l)}$ is derived by shuffling $f_c^{(l)}$ randomly across the batch dimension.

Following the method outlined by Huang & Belongie (2017), we then reconstruct the new feature maps using the perturbed feature statistics:

$$f_{new}^{(l)} = \sigma_{new} \frac{f_c^{(l)} - \mu(f_c^{(l)})}{\sigma(f_c^{(l)})} + \mu_{new}. \tag{19}$$

The perturbed feature maps are used to drive content-focused learning, thereby aiding in learning domain-agnostic embeddings.

3.2.3 Domain-focused Learning

Recognizing the strong correlation between image style and its domain, we prioritize extracting style features over traditional low-level features in the initial layers of the domain-specific encoder E_d. This approach aims to enhance the learning of domain-specific embeddings. To achieve this, we employ two widely used methods for style feature extraction.

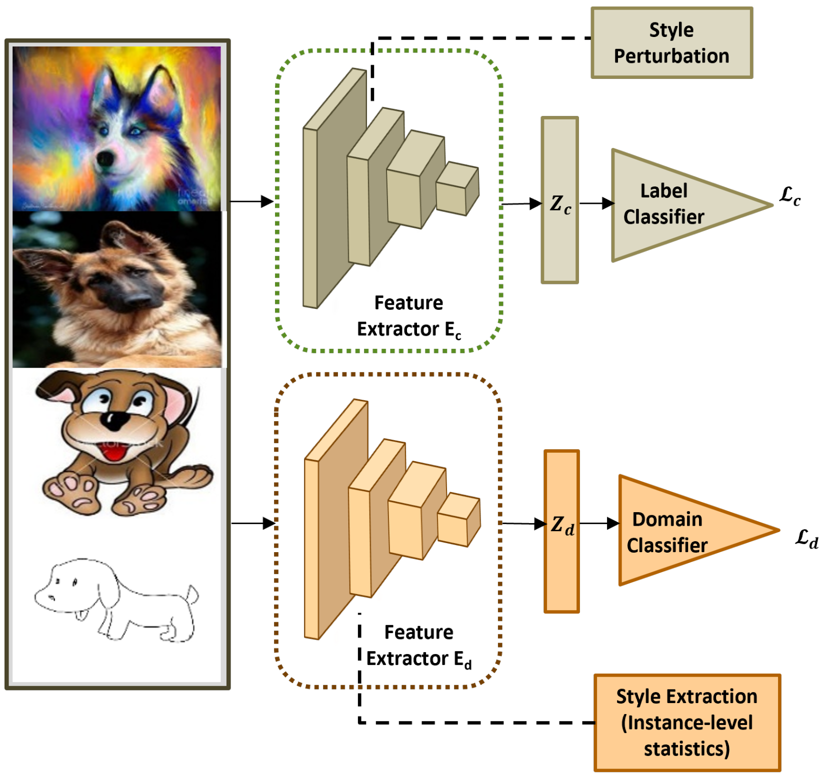

Figure 2: DETSI: Disentangled Embedding Through Style Information, leveraging instance-level feature statistics for style feature extraction to enable domain-focused learning.

Instance-Level Feature Statistics (Mean and Variance) We compute instance-level feature statistics, such as the mean and variance of feature activations, to represent an image's style. To capture style information, we utilize these instance-level statistics and modify the current feature maps following the methodology introduced by Huang & Belongie (2017):

$$f_{sty}^{(l)} = \mu_{sty} + \sigma_{sty} \cdot f_d^{(l)}, \tag{20}$$

where $f_d^{(l)}$ represents a specific batch of feature maps at a layer l of the encoder E_d, and μ_{sty} and σ_{sty} denote the style characteristics of the feature maps $f_d^{(l)}$. Instead of directly using $f_d^{(l)}$, the domain-specific embedding Z_d is computed using $f_{sty}^{(l)}$.

This approach enhances domain-specific learning and strengthens the disentanglement of domain-specific and domain-invariant components. By improving this separation, we enhance the model's ability to generalize to unseen domains. Fig. 2 illustrates the DETSI framework, which leverages instance-level feature statistics for style extraction to enable domain-focused learning.

Gram Matrix-Based Approach This work also considers another popular method introduced by Gatys et al. (2016) to capture the style using Gram matrices, which compute the correlations between feature channels in a CNN. The Gram matrix encodes second-order statistics (feature correlations), effectively capturing texture and style information. The Gram matrix is formally defined as:

$$G^{(l)} = f_d^{(l)} \cdot f_d^{(l)T}, \tag{21}$$

where $f_d^{(l)}$ represents a specific batch of feature maps at a layer l of the encoder E_d. To extract style information, Gram matrices are computed from the early layers of the domain-specific encoder E_d. Adaptive average pooling is applied to the Gram matrices to reduce complexity while retaining critical information, producing resized feature vectors denoted $f_{sty}^{(l)}$. These vectors are combined and passed through fully connected layers to learn the domain-specific embedding Z_d.

The DETSI framework leverages Gram matrices for style feature extraction to enable domain-focused learning, as illustrated in Fig. 3.

The performance analysis of both methods for style extraction is detailed in Section 5. Details about the network structure are provided in the following section.

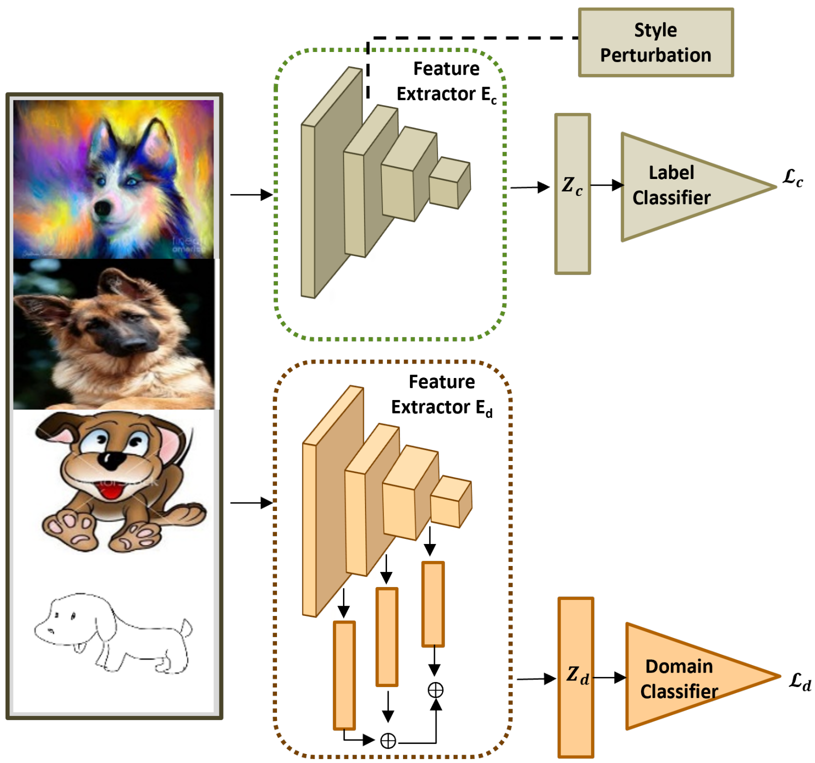

Figure 3: DETSI: Disentangled Embedding Through Style Information, utilizing Gram Matrix for style feature extraction to enable domain-focused learning.

4 Experimental Setup and Evaluation

This section details the experimental setup, including the datasets, implementation specifics, and network architectures, to evaluate the proposed methods for DG. We utilize three widely recognized benchmarks, PACS, VLCS, and Office-Home, each designed to test the robustness of DG models across diverse domains and styles. The implementation details describe the training protocols, hyperparameter configurations, and data preprocessing techniques, ensuring reproducibility.

Finally, we outline the architecture of the feature extractors, predictors, and auxiliary networks used in our experiments, emphasizing their roles in learning disentangled embeddings and leveraging style and mutual

Algorithm 2 Training procedure for DETSI

Input: K domains data samples; Encoders E_c, E_d; Classifiers C, $\tilde{C}$
Output: Optimized Encoder E_c and Classifier C
for $i = 1$: epochs **do**
  Sample a mini-batch $(x_i^{(k)}, y_i^{(k)})_{i=1}^{B} \in S^k$
  $f_c^{(l)} = E_c^l(x_i^k)$
  $f_d^{(l)} = E_d^l(x_i^k)$
  $\tilde{f}_c^{(l)}$= Shuffle $(f_c^{(l)})$
  $f_{new}^{(l)}$= StylePerturbation $(f_c^{(l)}, \tilde{f}_c^{(l)})$
  $f_{sty}^{(l)}$= StyleExtraction $(f_d^{(l)})$
  Compute the objective function.
  $\mathcal{L}_{dI} = \mathcal{L}_c + \lambda_1 \mathcal{L}_d$
  Update E_c, E_d, C, $\tilde{C}$
end for
Return Optimized Encoder E_c and Classifier C

information for improved generalization. These components collectively validate the effectiveness of our framework in addressing the challenges posed by DG.

4.1 Datasets

PACS PACS is a benchmark for object recognition in DG tasks. It includes four domains named Photo, Art-painting, Cartoon, and Sketch, each characterized by significant variations in image styles. The dataset contains 9,991 images across seven classes: dog, elephant, giraffe, guitar, horse, house, and person. This study uses the official training-validation split.

VLCS VLCS is another object recognition dataset, comprising 10,729 images across five categories. It includes four domains: VOC 2007 (Pascal), LabelMe, Caltech, and Sun. The training and validation split follows the methodology described in Li et al. (2018b).

Office-Home Office-Home is designed for DG and features 15,500 images spanning 65 categories. It includes four domains: Art, Clipart, Product, and Real-world with variations in viewpoints and image styles. The training and validation split adheres to the approach outlined in Xu et al. (2021). Fig. 4 provides an illustration of these three benchmarks.

4.2 Implementation Details

We follow the standard approach for domain generalization (DG) illustrated in Fig. 5. In this setup, one domain is designated as the target domain for testing, while the training and validation splits of the remaining domains are used to train the model and select the best-performing configuration, respectively.

The model is trained using mini-batch stochastic gradient descent (SGD) with a batch size of 32 for the PACS, Office-Home, and VLCS datasets. The training process spans 50 epochs, with a weight decay of 5e^{-4} and an initial learning rate of 0.001. The learning rate is reduced by a factor of 0.1 after every 40 epochs. The weighting coefficients λ_1 and λ_2 are set to 1. Standard data augmentation techniques are applied, including color jittering, horizontal flipping, and random resized cropping. All input images are resized to 224×224 to ensure consistency during training.

4.3 Structure of Networks

Feature Extractors We utilize pre-trained ResNet50 He et al. (2016) as the feature encoders E_c and E_d for all datasets. The encoder E_c extracts domain-agnostic features, denoted as Z_c, while E_d extracts domain-specific features, denoted as Z_d.

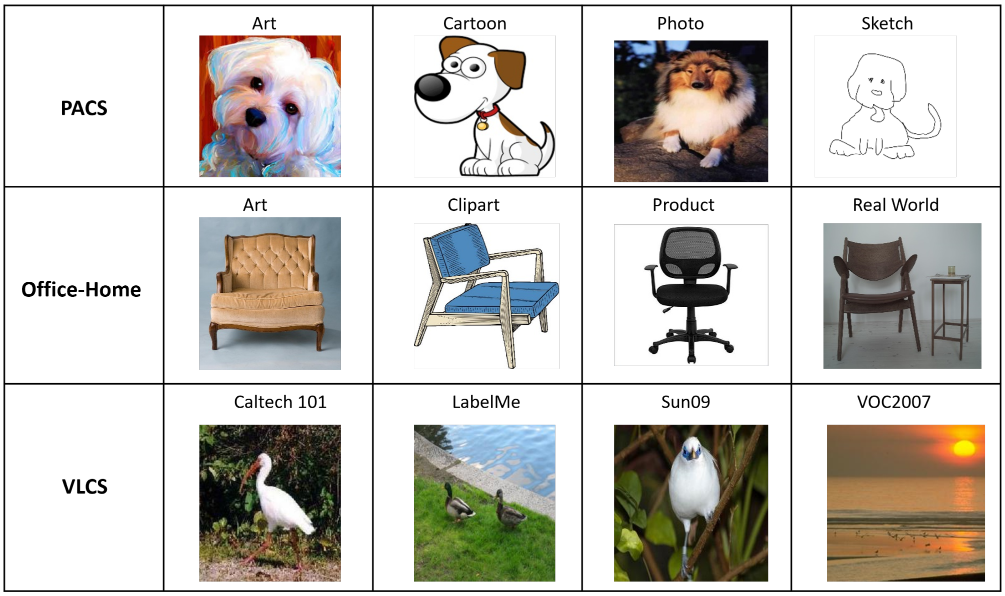

Figure 4: Domain Generalization Benchmarks

Figure 5: Training Vs. Inference Samples

Predictors The class label predictor consists of a single fully connected linear classifier C with an input dimension of 2048. We use a classifier $\tilde{C}$ to predict domain labels comprising an input layer of size 2048, two hidden layers, and an output layer. The dimensions of the hidden layers are identical to the input layer, and a ReLU activation function follows each hidden layer. The output dimensions of both classifiers correspond to the number of object classes and domains in the training data.

Network for Mutual Information Estimation The mutual information estimator, N_θ, is a neural network with two fully connected layers. The hidden layer outputs a feature vector of size 512, and the network produces a single scalar value representing the estimated mutual information between Z_c and Z_d.

Network for Style Extraction in DETSI In the DETSI framework, style information is extracted using Gram matrices computed from three layers of the encoder E_d. Experimental results indicate that the first layer captures the most significant style information compared to the subsequent layers. To optimize computational efficiency while preserving critical information, the Gram matrices initially sized 256×256 (layer 1), 512×512 (layer 2), and 1024×1024 (layer 3) are resized to a uniform dimension of 64×64 using adaptive average pooling. The resized Gram matrices are flattened and passed through a fully connected network with ReLU activation. The resulting features are fed into the domain classifier $\tilde{C}$, ensuring a balance between computational efficiency and effective style information utilization for disentanglement.

4.4 Results and Discussion

We employ ResNet-50 as the embedding encoder for the PACS, OfficeHome, and VLCS datasets. The corresponding results are presented in Table 1, Table 2, and Table 3, respectively. The results demonstrate that the proposed frameworks, DETMI and DETSI, outperform several state-of-the-art (SOTA) approaches in terms of average accuracy across these standard benchmarks.

For the PACS benchmark, both approaches achieve comparable performance compared to several SOTA methods, with average accuracy improvements of +0.74%. Notably, in the sketch domain—characterized by its reliance on domain-invariant features—DETSI significantly outperforms DETMI and other SOTA methods, demonstrating its effectiveness in learning disentangled embeddings for improved performance.

For the VLCS dataset, DETMI and DETSI also demonstrate notable improvements over SOTA methods, achieving average accuracy gains of +0.67% and +1.01%, respectively.

For the OfficeHome dataset, known for its complexity due to the large number of classes, our methods outperform existing SOTA approaches, achieving improvements of +0.99% and +0.64% in average accuracy for DETMI and DETSI, respectively.

These results underscore the effectiveness of our methods in learning domain-agnostic embeddings, significantly enhancing the generalization capabilities of our framework to previously unseen target domains.

5 Ablation Studies

This section presents ablation studies conducted to evaluate the DETSI framework using the PACS dataset. The studies analyze key components and design choices within the framework to assess their impact on performance and generalization.

The first study examines the residual interaction between domain-specific (Z_d) and domain-invariant (Z_c) feature spaces by incorporating a mutual information estimator. This estimator quantifies any remaining mutual information between the two embedding spaces, providing insights into the effectiveness of the disentanglement process.

The second study evaluates the impact of different style extraction techniques for learning domain-specific embeddings (Z_d). It compares the performance of Gram matrices and instance-level feature statistics, with style perturbation applied solely through instance-level feature statistics across various layers.

Table 1: Leave-one-domain-out results on PACS. The best and second-best results are bolded and underlined respectively.

Methods	Art	Cartoon	Photo	Sketch	Avg.
ResNet-50					
ERMVapnik (2013)	84.70	80.80	97.20	79.30	85.50
EISNet (ECCV 20)	86.64	81.53	97.11	78.07	85.84
DSON (ECCV 20)	87.00	80.60	96.00	82.09	86.60
SagNet (CVRR 21)	87.40	80.70	97.10	80.00	86.30
MatchDG (ICML 21)	85.61	82.12	97.94	78.76	86.11
mDSDI (NeurIPS 21)	88.10	81.10	**98.40**	79.60	86.80
MIRO (ECCV 22)	-	-	-	-	85.04
POEM (AAAI 23)	-	-	-	-	86.90
CCFP (ICCV 23)	-	-	-	-	86.60
INSURE (TIP 24)	**90.20**	**85.30**	97.90	<u>83.80</u>	<u>89.30</u>
CMCL (TNNLS 25)	87.57	83.60	96.03	83.73	87.73
DETMI (Ours)	88.76	82.55	98.08	82.08	87.87
DETSI (Ours)	<u>90.18</u>	<u>84.85</u>	<u>98.20</u>	**86.94**	**90.04**

Table 2: Leave-one-domain-out results on VLCS. The best and second-best results are bolded and underlined respectively.

Methods	Caltech	Labelme	Pascal	Sun	Avg.
ResNet-50					
SagNet (CVRR 21)	-	-	-	-	77.80
mDSDI (NeurIPS 21)	-	-	-	-	79.30
MIRO (ECCV 22)	-	-	-	-	79.00
POEM (AAAI 23)	-	-	-	-	79.80
CCFP (ICCV 23)	-	-	-	-	79.20
INSURE (TIP 24)	<u>98.80</u>	63.80	**81.04**	72.20	79.01
DETMI (Ours)	98.65	<u>68.22</u>	<u>78.70</u>	<u>76.32</u>	<u>80.47</u>
DETSI (Ours)	**99.15**	**68.67**	78.34	**77.08**	**80.81**

Table 3: Leave-one-domain-out results on OfficeHome. The best and second-best results are bolded and underlined respectively.

Methods	Art	Clipart	Product	Real	Avg.
ResNet-50					
ERMVapnik (2013)	61.30	52.40	75.80	76.60	66.50
DDAIG (AAAI 20)	59.20	52.30	74.60	76.00	65.50
L2A-OT (ECCV 20)	60.60	50.10	74.80	77.00	65.60
SagNet (CVRR 21)	63.40	54.80	75.80	78.30	68.10
mDSDI (NeurIPS 21)	<u>68.40</u>	52.50	76.20	<u>80.60</u>	69.42
IIB (AAAI 22)	-	-	-	-	68.60
POEM (AAAI 23)	-	-	-	-	68.20
CCFP (ICCV 23)	-	-	-	-	68.90
CMCL (TNNLS 25)	67.22	57.88	<u>78.47</u>	79.79	70.84
DETMI (Ours)	**68.58**	<u>58.28</u>	**79.22**	**81.27**	**71.83**
DETSI (Ours)	67.30	**60.29**	78.30	80.03	<u>71.48</u>

The third study investigates the effect of style perturbation in the task-specific encoder and style extraction in the domain-specific encoder across different layers. The findings highlight the critical role of early encoder layers in achieving effective disentanglement.

Finally, the fourth study explores the independent contributions of style perturbation for task-specific embeddings and style extraction for domain-specific embeddings. This analysis underscores their complementary roles in facilitating robust disentanglement and enhancing generalization across domains.

5.1 Mutual Information Analysis in DETSI

We integrated a mutual information estimator into the DETSI framework, as shown in Fig. 6, to evaluate the residual interaction between domain-specific (Z_d) and domain-invariant (Z_c) feature spaces. This integration enabled us to quantify any remaining mutual information between the two embedding spaces. The results, summarized in Table 4, indicate no significant performance improvement with the addition of mutual information estimation. These findings confirm that DETSI's inherent design effectively achieves robust disentanglement without the need for additional mutual information measures.

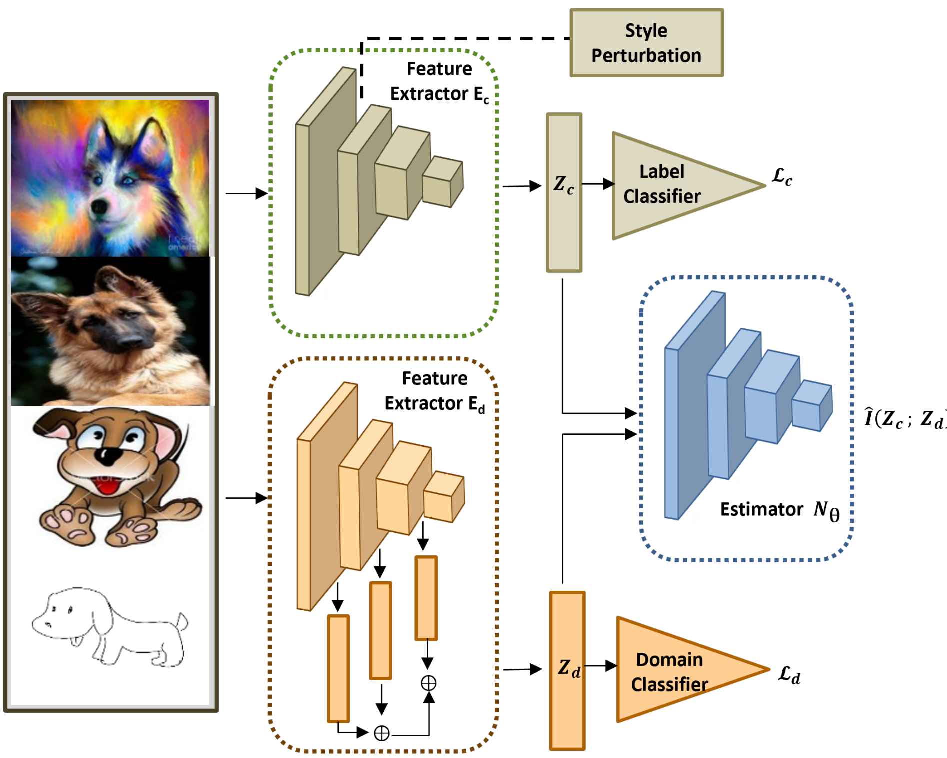

Figure 6: DETSI with Mutal Information Estimation

Table 4: Mutual Information Analysis in DETSI using the PACS Dataset.

Methods	Art	Cartoon	Photo	Sketch	Avg.
		ResNet-50			
DETSI with MI	89.79	84.72	98.32	86.43	89.82

5.2 Impact of Style Extraction Techniques on Learning Domain-Specific Embeddings

This subsection presents the experimental results evaluating the impact of different style extraction techniques on learning domain-specific embeddings (Z_d). As shown in Table 5, Gram matrix-based style extraction consistently outperforms methods based on instance-level feature statistics.

The superior performance of Gram matrices stems from their ability to encode second-order statistics, which capture correlations between feature maps across the entire image. These correlations effectively represent texture patterns, color distributions, and structural styles, all of which are critical components of an image's overall style. In contrast, instance-level feature statistics primarily focus on low-level details, such as brightness and contrast, but fail to capture global style attributes, including texture patterns, artistic brushstrokes, and structural relationships. These limitations make instance-level feature statistics less effective for representing domain-specific style information.

Table 5: Impact of Style Extraction Techniques on Learning Domain-Specific Embeddings Z_d on the PACS Dataset.

Methods	Art	Cartoon	Photo	Sketch	Avg.
		ResNet-50			
Instance-level Statistics	86.69	84.25	98.08	85.72	89.43
Gram Matrix	90.18	84.85	98.20	86.94	90.04

5.3 Effect of Style Perturbation and Extraction Across Layers in DETSI

This study examines the impact of style extraction and perturbation across the early layers of encoders in the DETSI framework, using instance-level feature statistics to learn domain-specific embeddings.

The results, summarized in Table 6, reveal that applying style extraction and perturbation within the first three layers of the domain-specific encoders achieves superior performance compared to other layer combinations. These findings align with the established understanding that early encoder layers primarily capture low-level features, such as textures and patterns, which are closely associated with style.

Interestingly, the fourth layer, which primarily encodes high-level semantic features rather than style attributes, was excluded from this analysis. This exclusion underscores the critical role of early encoder layers in style manipulation for enhancing the disentanglement of domain-specific and domain-agnostic embeddings in the DETSI framework.

Table 6: Evaluation of Style Feature Extraction and Perturbation Across Different Layers of DETSI on the PACS Dataset

Methods	Art	Cartoon	Photo	Sketch	Avg.
		ResNet-50			
$E_c(L1), E_d(L1)$	89.79	84.51	98.08	82.92	88.82
$E_c(L12), E_d(L12)$	89.59	84.00	98.32	84.72	89.15
$E_c(L123), E_d(L123)$	89.69	84.25	98.08	85.72	89.43

5.4 Effect of Style Perturbation and Style Feature Extraction in the DETSI Framework

This ablation study investigates the individual and combined effects of style perturbation for task-specific embeddings and style extraction for domain-specific embeddings within the DETSI framework. Initially, we evaluated each component independently. The results, presented in Table 7, indicate that both style perturbation and style extraction independently contribute to improved performance.

Notably, the combined application of these components yielded significantly better results than either method used in isolation. This finding highlights the complementary nature of style perturbation and style extraction, as their integration facilitates enhanced disentanglement and richer feature representations. The combined approach ultimately improves the model's generalization performance across diverse domains.

Table 7: Evaluation of DETSI Performance on the PACS Dataset for Each Component

Methods	Art	Cartoon	Photo	Sketch	Avg.
		ResNet-50			
ERMVapnik (2013)	84.70	80.80	97.20	79.30	85.50
Style Perturbation (E_c)	90.18	84.59	98.02	85.54	89.58
Style Extraction (E_d)	88.67	83.19	98.08	81.62	87.89
DETSI	90.18	84.85	98.20	86.94	90.04

6 Conclusion

This paper addresses the challenge of domain shift in Domain Generalization (DG) by introducing two novel frameworks: Disentangled Embedding through Mutual Information (DETMI) and Disentangled Embedding through Style Information (DETSI). These frameworks effectively disentangle the latent feature space into domain-specific and domain-invariant components, enabling the extraction of class-relevant features and enhancing generalization to unseen distributions.

DETMI utilizes a mutual information estimator to enforce feature disentanglement, while DETSI achieves disentanglement through style extraction and perturbation. Both frameworks demonstrate superior performance compared to state-of-the-art DG techniques, promoting domain invariance and improving generalization. Notably, DETSI achieves comparable results with reduced complexity, making it a practical and efficient solution for scenarios with limited computational resources.

The proposed frameworks advance the state-of-the-art in DG and highlight the critical role of leveraging domain-specific information alongside domain-invariant features. These findings underscore the potential of disentanglement-based approaches to effectively address domain shift and provide a foundation for developing more efficient and robust DG methods in the future.

References

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201, 2021.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Sang-Yeong Jo and Sung Whan Yoon. Poem: polarization of embeddings for domain-invariant representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8150–8158, 2023.

Chenming Li, Daoan Zhang, Wenjian Huang, and Jianguo Zhang. Cross contrasting feature perturbation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1327–1337, 2023.

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*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639, 2018c.

Wang Lu, Jindong Wang, Haoliang Li, Yiqiang Chen, and Xing Xie. Domain-invariant feature exploration for domain generalization. *Transactions on Machine Learning Research (TMLR)*, 2022.

Noaman Mehmood and Kenneth Barner. Augmentation, mixing, and consistency regularization for domain generalization. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*, pp. 1–6. IEEE, 2024.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2022.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

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*, pp. 5018–5027, 2017.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14383–14392, 2021.

Xi Yu, Huan-Hsin Tseng, Shinjae Yoo, Haibin Ling, and Yuewei Lin. Insure: an information theory inspired disentanglement and purification model for domain generalization. *IEEE Transactions on Image Processing*, 2024.

Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6277–6286, 2021.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13025–13032, 2020a.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 561–578. Springer, 2020b.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.