POEM: Polarization of Embeddings for Domain-Invariant Representations

Sang-Yeong Jo, Sung Whan Yoon*

Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
jsy7058@unist.ac.kr, shyoon8@unist.ac.kr

Abstract

Handling out-of-distribution samples is a long-lasting challenge for deep visual models. In particular, domain generalization (DG) is one of the most relevant tasks that aims to train a model with a generalization capability on novel domains. Most existing DG approaches share the same philosophy to minimize the discrepancy between domains by finding the domain-invariant representations. On the contrary, our proposed method called POEM acquires a strong DG capability by learning domain-invariant and domain-specific representations and polarizing them. Specifically, POEM co-trains category-classifying and domain-classifying embeddings while regularizing them to be orthogonal via minimizing the cosine-similarity between their features, i.e., the polarization of embeddings. The clear separation of embeddings suppresses domain-specific features in the domain-invariant embeddings. The concept of POEM shows a unique direction to enhance the domain robustness of representations that brings considerable and consistent performance gains when combined with existing DG methods. Extensive simulation results in popular DG benchmarks with the PACS, VLCS, OfficeHome, TerraIncognita, and DomainNet datasets show that POEM indeed facilitates the category-classifying embedding to be more domain-invariant.

Introduction

Despite the immense effort dedicated during the past decade, enhancing deep models to acquire a strong generalization capability on novel data distribution remains a daunting challenge. For computer vision, particularly, the distributional shift of the image domain between the train and test sets, known as domain shift, provokes significant performance degradation of deep visual models. Domain generalization (DG), the task of interest here, pursues developing algorithmic methods to overcome the domain shift. Specifically, the DG task assumes that an image classification model is trained on the data from source domains, such as photos, sketches, cartoons, etc., then the model is tested on the target domains which are not shown in the training phase.

To overcome the domain shift problem, most of the existing DG approaches are built upon the philosophy of minimizing the discrepancy across source domains, which aims to obtain domain-invariant knowledge. First of all, various algorithmic approaches have been proposed to minimize the divergence measurements across domains, such as the contrastive loss for alignment of in-class features from domains (Motiian et al. 2017; Dou et al. 2019), the Kullback-Leibler divergence (Kullback and Leibler 1951; Dou et al. 2019), and the maximum mean discrepancy between domains (Gretton et al. 2012; Li et al. 2018b). Another branch of approaches tries to utilize domain-specific information to learn domain-invariant representation via the employment of per-domain embedding network (Bousmalis et al. 2016) and domain classifiers (Ganin and Lempitsky 2015). Also, multi-task self-supervised learning (Albuquerque et al. 2020; Wang et al. 2020; Carlucci et al. 2019), optimization-based meta-learning (Li et al. 2018a; Dou et al. 2019), and ensemble learning (Arpit et al. 2022; Mancini et al. 2018; Zhou et al. 2021a) are shown to enhance the model robustness across domain shifts. On the other hand, another group of algorithms pursues to erase domain-related spurious factors in input space, such as the texture of images (Wang et al. 2019) or sensitive features in representation space (Huang et al. 2020) to obtain domain-invariant features.

In the surge of various DG approaches to suppress discrepancy between domains, a work of (Gulrajani and Lopez-Paz 2021) reveals that, when a model is carefully trained, Empirical Risk Minimization (ERM) of (Vapnik 1998), which is probably the simplest approach for training across multiple domains, outperforms the existing complicated DG methods. After the surprising findings, many researchers have turned attention to developing particular optimizers that make models robust, rather than employing explicit ways to find domain-invariant representation. For instance, recent approaches beat many prior works by combining ERM with model averaging methods for seeking flatter minima in loss landscape (Izmailov et al. 2018; Cha et al. 2021). In addition, a very recent work of (Cha et al. 2022) maximizes the mutual information between a DG model and a pretrained oracle representation, rather than adopting a particular way to make the DG model more domain-invariant.

To the best of our knowledge, most of the existing DG methods aim to discard domain-specific information to reduce the divergence of representations between different domains or indirectly utilize domain-specific information to facilitate the acquisition of domain-invariant representations.

*Sung Whan Yoon is the corresponding author.

Moreover, recently suggested methods of (Cha et al. 2021, 2022) overlook the effort for finding domain-invariant representation and focus on the robust-guaranteeing optimization methods of models. We want to emphasize a significant difference between the strategy of prior work and how humans identify image categories across different domains. For a given image, human recognizes the image category and domain together, and construct domain-invariant features based on the understanding of domain-specific features, i.e., human clearly acknowledges how a cartoon-based cat looks different from a photograph-based cat. In contrast, none of the existing DG methods can explicitly identify both the domain-specific and domain-invariant features, and distinctively learn them to build domain-robust knowledge.

With this motivation, we propose a DG method called POEM that aims to learn both domain-invariant and domain-specific features which are clearly separated from each other. Specifically, POEM employs two distinctive embeddings for the category and domain classification tasks, respectively, and zero-forces their cosine similarity to strengthen the clear discrimination between two embeddings. POEM eventually forces two representations of category and domain classification tasks to be orthogonal, where one contains domain-invariant features for category classification and another one bears domain-specific features for domain classification; here, we call the process as *polarization*.

We empirically show that POEM promotes the category-classifying embedding to be more domain-invariant. Also, we informally describe how POEM improves the generalization capability. The concept of POEM with the disentangled domain-specific and domain-invariant representations enlightens a unique direction to further improve the performance of the existing DG methods. Extensive simulations on the popular DG benchmarks including PACS (Li et al. 2017), VLCS (Fang, Xu, and Rockmore 2013), Office-Home (Venkateswara et al. 2017), TerraIncognita (Beery, Van Horn, and Perona 2018), and DomainNet (Peng et al. 2019) demonstrate that POEM yields a considerable gain when combined with the cutting-edge DG algorithms.

The main contributions of this paper are as follows:

- We propose a method called POEM that enhances the DG capability via polarization of domain-invariant and domain-specific features.
- We provide a brief explanation that informally describes the improvement of DG ability based on the separation of domain-invariant and domain-specific features.
- We demonstrate a consistent and considerable performance gain of POEM when combined with the cutting-edge DG methods.

Related Work

Beyond the brief summary of prior domain generalization (DG) methods in the Introduction, we herein focus on describing the highly-related works to POEM and the recent trend of DG algorithms.

Aligning Domains via Domain-Specific Knowledge

Most of the existing DG methods rely on the principle that minimization of the discrepancy across training domains improves the DG capability of models. A group of methods in (Bousmalis et al. 2016; Mancini et al. 2018) adopts per-domain embeddings that classify categories of images in each domain, and reduce the discrepancy between them. As another strategy that utilizes domain-specific knowledge to acquire domain-invariant representation, the method in (Ganin and Lempitsky 2015) employs a classifier of image domains and gradient-reversely co-trains it with the image category classifier. The process makes the model inept to recognize domains. In contrast to the prior methods, our method POEM explicitly co-trains category- and domain-classifying embeddings and disentangles them to achieve better generalization, which is never been proposed. The methods of (Bousmalis et al. 2016; Mancini et al. 2018) does not employ domain-classifying representations, and the algorithm of (Ganin and Lempitsky 2015) adopts just a domain classifier, not a domain-classifying embedding.

Erasing Domain-dependancy

On the other hand, DG approaches with the domain-erasing strategy pursue to discard domain-dependent features. A method called NGLCM of (Wang et al. 2019) regularizes domain-dependent texture features of images extracted by Gray-Level Co-occurrence Matrix (GLCM) of (Haralick, Shanmugam, and Dinstein 1973; Lam 1996). Representation Self-Challenging (RSC) of (Huang et al. 2020) learns to mask sensitive features in the representation space, which are believed to be domain-dependent. Common Specific Decomposition (CSD) of (Piratla, Netrapalli, and Sarawagi 2020) decomposes the model parameters into the common and domain-specific parts to identify the domain-invariant model parameters. When compared to the domain-erasing methods, POEM erases domain-dependent parts from domain-invariant representations by reducing the similarity between the category- and domain-classifying embeddings. However, POEM is fundamentally different from the method of (Wang et al. 2019) that relies on visual characteristics such as texture, and the methods of (Huang et al. 2020; Piratla, Netrapalli, and Sarawagi 2020) that are not able to recognize explicit domain-dependent representations.

Optimizing Models for Generalization

After the authors of (Gulrajani and Lopez-Paz 2021) claim that Empirical Risk Minimization (ERM) of (Vapnik 1998) shows outperforming performance beyond the existing complicated DG methods, ensemble learning of moving average models (EoA) of (Arpit et al. 2022) shows the improved DG performance by just averaging model parameters during the ERM training steps. A group of approaches surpasses combines ERM and the model averaging methods that find flatter minima in loss space (Izmailov et al. 2018; Cha et al. 2021). POEM is also built upon ERM, which is the simplest way to handle the DG task and is easily plugged in with the flat-minima searching methods called Stochastic Weight Averaging Densely (SWAD) of (Cha et al.

2021) for cutting-edge DG performance. As the MIRO case, POEM can enhance the domain invariance of a model in conjunction with SWAD, which pays less attention to finding domain-invariant features.

Utilizing Pretrained Knowledge

Well-pretrained models from other datasets can be used for better DG performance. As a very recent work, Mutual Information Regularization with Oracle (MIRO) of (Cha et al. 2022) aims to maximize the mutual information between the pretrained oracle representation and the target model's representations for better generalization. MIRO does not adopt an explicit way to find domain-invariant features but just makes a model be similar to the oracle. Our method is essentially different from MIRO, so POEM can be in conjunction with MIRO to yield an additional performance gain via enhancing the domain invariance.

Proposed Method

In this section, the problem settings of domain generalization (DG) are presented and the details of the proposed algorithm POEM are described.

Problem Settings of Domain Generalization

Let us denote the set of training domains as $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^{K}$ where D_k is the k-th training domain. For a classification model $f(x;\theta)$ and the loss function $\mathcal{L}$, the objective of the DG task is to find the model parameter θ which is generalized well on the target domain $\mathcal{T}$, i.e.,

$$\theta^* = \arg\min_{\theta} \mathcal{L}\big(f(\mathbf{x};\theta), y\,;\mathcal{D}\big), \tag{1}$$

where $(\mathbf{x}, y)$ is a pair of input and class label from $\mathcal{T}$.

Model Description of POEM

POEM consists of a set of *elementary embeddings*. For the DG task, POEM contains two elementary embeddings, one is for image category classification, and the other one is for image domain classification. Here, we extend the concept to contain N number of elementary embeddings for a more general description. Based on the architecture, POEM adopts *disentangling loss* for spatially separating the elementary embeddings and discrimination loss for discriminating the features from different embeddings.

Set of elementary embeddings: Let us denote the set of elementary embedding as $\mathfrak{F} : \mathbb{R}^D \to \mathbb{R}^{N\times L}$ which is the set of elementary embeddings $\mathfrak{F} = \{f_i\}_{i=1}^{N}$ with model parameter $\Theta = \{\theta_i\}_{i=1}^{N}$:

$$\mathfrak{F}(\mathbf{x}\,;\Theta) \triangleq \big\{f_i(\mathbf{x}\,;\theta_i)\big\}_{i=1}^{N}, \tag{2}$$

where N is the number of elementary embeddings. Each elementary embedding f_i that is parameterized by θ_i maps an input $\mathbf{x}$ to the feature vector with the length of L. For the set of elementary embeddings, there exist N elementary tasks with different classifiers, i.e., category classifiers and domain classifiers for the DG task. The classifier $\mathbf{\Phi}$ is the set of N classifiers for elementary tasks. For a given input $\mathbf{x}$ and i-th elementary embedding, the classification loss $\mathcal{L}_c$

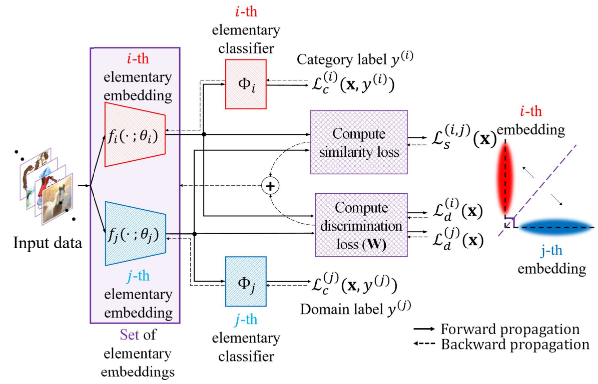


Figure 1. Proposed model architecture of POEM

is calculated with cross-entropy $\mathcal{H}$ with the probability from the Softmax computation and target label $y^{(i)}$:

$$\mathcal{L}_c^{(i)}(\mathbf{x}, y) = \mathcal{H}\Big(\text{Softmax}\big\{f_i(\mathbf{x}\,;\theta_i)\Phi_i\big\}, y^{(i)}\Big) \tag{3}$$

For the DG task, there exist $N = 2$ pairs of elementary embedding and classifier for category and domain classification, respectively. For instance, the PACS dataset contains seven categories, three train domains, and a single target domain. POEM then contains two elementary embeddings that classify seven categories and three domains for each.

Disentangling loss: POEM computes disentangling loss for separating elementary embeddings from each other. To be specific, the cosine-similarity loss between features from different embeddings is zero-forced. For a given input $\mathbf{x}$, the disentangling loss $\mathcal{L}_s^{(i,j)}(x)$ for a pair of i and j-th elementary embeddings is calculated as follows:

$$\mathcal{L}_s^{(i,j)}(\mathbf{x}) = \big|K\big(f_i(\mathbf{x}\,;\theta_i), f_j(\mathbf{x}\,;\theta_j)\big)\big|, \tag{4}$$

where $K(\cdot,\cdot)$ is the cosine similarity function of two vectors. The absolute operation $|\cdot|$ is for making the similarity be positive. We select cosine similarity for the disentangler to orthogonalize two embedded features.

Discrimination loss: POEM adopts discrimination loss which is to recognize the index of embeddings for a given feature. The discriminator $\mathbf{W}$ is a simple classifier with N classification weights: $\mathbf{W} = \{w_i\}_{i=1}^{N}$. For a given $\mathbf{x}$ and i-th elementary embedding, discrimination loss $\mathcal{L}_d^{(i)}(\mathbf{x})$ is computed with cross-entropy with the probability from Softmax calculation and target label i:

$$\mathcal{L}_d^{(i)}(\mathbf{x}) = \mathcal{H}\Big(\text{Softmax}\big\{f_i(\mathbf{x};\theta_i)\mathbf{W}\big\}, i\Big) \tag{5}$$

For the DG case, the discrimination is a binary classification to figure out the index of the embedding from the input feature vector.

In Fig. 1, the model architecture of POEM for the DG task is illustrated. The set of elementary embeddings contains two elementary embeddings f_i and f_j for the category-classification and the domain-classification tasks, respectively. Based on the two classfiers Φ_i and Φ_j for classifying image categories and domains as respectively, POEM calculates two classification loss terms denoted as $\mathcal{L}_c$. For orthogonalizing features from two elementary embeddings, POEM computes the disentangling loss $\mathcal{L}_s$. For the final loss term, a discriminator with parameter $\mathbf{W}$ calculates the discrimination loss $\mathcal{L}_d$.

Learning Procedures of POEM

Training phase: The learning procedures of POEM are based on the most straightforward framework called Empirical Risk Minimization (ERM) (Vapnik 1998; Gulrajani and Lopez-Paz 2021) that minimizes the empirical risk, which is the average of category-classification losses $\mathcal{L}$ over the source domains. The empirical risk is formulated as follows:

$$\hat{\mathcal{E}}_{\mathcal{B}}(\theta) \triangleq \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x},y)\in\mathcal{B}} \mathcal{L}(f(\mathbf{x};\theta), y), \tag{6}$$

where $\mathcal{B} = \{\mathcal{B}_k\}_{k=1}^{K}$ is a mini-batch, and $\mathcal{B}_k$ is a sampled mini-batch from $\mathcal{D}_k$ of domain k. $f(\cdot\,;\theta)$ is an embedding parameterized by θ, and y is the image category label. Similarly, POEM trains learnable parameters including Θ, Φ and $\mathbf{W}$ to minimize the empirical risk as follows:

$$\hat{\mathcal{E}}_{\mathcal{B}}(\Theta, \Phi, \mathbf{W}) \triangleq \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x},y)\in\mathcal{B}} \mathcal{L}(\mathfrak{F}(\mathbf{x};\Theta), \Phi, \mathbf{W}, y). \tag{7}$$

The particular loss term $\mathcal{L}$ is computed by considering the classification loss of elementary tasks $\mathcal{L}_c$, the disentangling loss $\mathcal{L}_s$ between different embeddings, and the discrimination loss $\mathcal{L}_d$ for each embedding which are aforementioned:

$$\begin{aligned} &\mathcal{L}(f\big(\mathbf{x};\Theta), \Phi, \mathbf{W}, y) = \\ &\frac{1}{N}\sum_{i=1}^{N}\Big\{\mathcal{L}_c^{(i)}(\mathbf{x}, y) + \mathcal{L}_d^{(i)}(\mathbf{x}) + \sum_{j\neq i}^{N}\mathcal{L}_s^{(i,j)}(\mathbf{x})\Big\}. \end{aligned} \tag{8}$$

Then the set of parameters Θ, Φ and $\mathbf{W}$ are updated by computing the gradients of the empirical risk, i.e., $\hat{\mathcal{E}}_{\mathcal{B}}(\Theta, \Phi, \mathbf{W})$:

$$\nabla\hat{\mathcal{E}}_{\mathcal{B}}(\Theta, \Phi, \mathbf{W}) = \frac{1}{N|\mathcal{B}|} \sum_{(\mathbf{x},y)\in\mathcal{B}} \sum_{i=1}^{N} \nabla\mathcal{L}(\mathfrak{F}\big(\mathbf{x};\Theta), \Phi, \mathbf{W}, y) \tag{9}$$

Testing phase: In testing, POEM keeps the embedding and classifier for the category-classifying task but drops other embeddings and classifiers. With the retained embedding and classifier, i.e., $f_z(\cdot;\theta_z)$ and Φ_z, POEM is evaluated on the samples in the target domains $\mathcal{T}$, where z is the index of the elementary embedding for classifying categories of images. Algorithm 1 presents the pseudocode of POEM.

Understanding of POEM

Herein, we informally explain how POEM improves the domain generalization capability. Although the explanation is not a formal mathematical analysis, we conceptually understand how the elementary embeddings of POEM are constructed and how the well-trained POEM achieves an improved generalization capability beyond ERM.

Algorithm 1: Training procedures for POEM

Input: Training domain $\mathcal{D}$, Number of elementary embeddings N, learning rate η
Initialization: Initial weights Θ_0, $\mathbf{\Phi}_0$, and $\mathbf{W}_0$, set of elementary embeddings $\mathfrak{F}(\cdot\,;\Theta_0)$
Output: Parameterized model $f_z(\cdot\,;\theta_z)$ and classifier Φ_z

for $\tau = 1, \cdots, T$ **do**
  Sample a mini-batch $\mathcal{B} = \{\mathcal{B}_k\}_{k=1}^{N}$, where $\mathcal{B}_k \in \mathcal{D}_k$
  for $(\mathbf{x}, y) \in \mathcal{B}$ **do**
    Set $\mathcal{L}_c = 0$, $\mathcal{L}_s = 0$, and $\mathcal{L}_d = 0$
    for $i = 1, \cdots, N$ **do**
      $\mathcal{L}_c \leftarrow \mathcal{L}_c + \mathcal{L}_c^{(i)}(\mathbf{x}, y)$ ▷ Eq. (3)
      $\mathcal{L}_d \leftarrow \mathcal{L}_d + \mathcal{L}_d^{(i)}(\mathbf{x})$ ▷ Eq. (5)
      for $j = 1, \cdots, N$ **do**
        if $j \neq i$ **then**
          $\mathcal{L}_s \leftarrow \mathcal{L}_s + \mathcal{L}_s^{(i,j)}(\mathbf{x})$ ▷ Eq. (4)
        end if
      end for
    end for
  end for
  $\hat{\mathcal{E}}_{\mathcal{B}} \leftarrow \frac{1}{N|\mathcal{B}|}(\mathcal{L}_c + \mathcal{L}_s + \mathcal{L}_d)$
  $(\Theta, \mathbf{\Phi}, \mathbf{W}) \leftarrow (\Theta, \mathbf{\Phi}, \mathbf{W}) - \eta\nabla\hat{\mathcal{E}}_{\mathcal{B}}$
end for
Return $f_z(\cdot\,;\theta_z)$ and Φ_z, where z is the index of the category-classifying embedding

---

Before the explanation, let us introduce some useful notations. We denote the trained set of elementary embeddings of POEM as $\mathfrak{F}(\cdot;\Theta^*) = \{f_i(\cdot;\theta_i^*)\}_{i=1}^{N}$ where $f_i(\cdot;\theta_i^*)$ is i-th elementary embedding with the learned parameters θ_i^*, and N is the number of elementary embeddings. N_i is the number of labels for the classification task of i-th embeddings, e.g., when we have seven image categories and four domains, $N_1 = 7$ and $N_2 = 4$. $\mathcal{X}$ is the input distribution that contains input samples $\mathbf{x}$. Let us denote the distribution of feature vectors of i-th elementary embedding as $\mathcal{Z}_i^*$. Based on the notations, let us describe the following desirable properties of the trained POEM embeddings.

Property 1. *(from the discrimination loss $\mathcal{L}_d^{(i)}$)* When the feature $\mathbf{z}_i^*$ is extracted by i^{th} embedding, i.e., $\mathbf{z}_i^* \sim \mathcal{Z}_i^*$, then

$$\mathbf{z}_i^* \cdot \mathbf{w}_i \geq \max_{j\neq i}(\mathbf{z}_i^* \cdot \mathbf{w}_j). \tag{10}$$

Based on the discrimination loss, POEM is trained to identify the index of embedding where a given feature is extracted. Thus the property is desirable. POEM tries to separate the feature distribution of each embedding so that the distributions are not overlapped.

Property 2. *(from the disentangling loss $\mathcal{L}_s^{(i,j)}$)* When two feature vectors are extracted from different i^{th} and j^{th} embeddings for a single input $\mathbf{x}$, then

$$\big|K\big(f_i(\mathbf{x};\theta_i^*), f_j(\mathbf{x};\theta_j^*)\big)\big| \simeq 0. \tag{11}$$

Based on the disentangling loss for a given input, POEM is trained to minimize the cosine similarity between two fea-

Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Average
MMD (Li et al. 2018b)	84.7	77.5	66.4	42.2	23.4	58.8
Mixstyle (Zhou et al. 2021b)	85.2	77.9	60.4	44.0	34.0	60.3
GroupDRO (Sagawa et al. 2020)	84.4	76.7	66.0	43.2	33.3	60.7
IRM (Arjovsky et al. 2019)	83.5	78.6	64.3	47.6	33.9	61.6
ARM (Zhang et al. 2021)	85.1	77.6	64.8	45.5	35.5	61.7
VREx (Krueger et al. 2021)	84.9	78.3	66.4	46.4	33.6	61.9
CDANN (Li et al. 2018c)	82.6	77.5	65.7	45.8	38.3	62.0
DANN (Ganin et al. 2016)	83.7	78.6	65.9	46.7	38.3	62.6
RSC (Huang et al. 2020)	85.2	77.1	65.5	46.6	38.9	62.7
MTL (Blanchard et al. 2021)	84.6	77.2	66.4	45.6	40.6	62.9
I-Mixup (Xu et al. 2020)	84.6	77.4	68.1	47.9	39.2	63.4
MLDG (Li et al. 2018a)	84.9	77.2	66.8	47.8	41.2	63.6
SagNet (Nam et al. 2021)	86.3	77.8	68.1	48.6	40.3	64.2
CORAL (Sun and Saenko 2016)	86.2	78.8	68.7	47.7	41.5	64.5
SWAD (Cha et al. 2021)	88.1	79.1	70.6	50.0	46.5	66.9
MIRO (Cha et al. 2022)	85.4	79.0	70.5	50.4	44.3	65.9
ERM† (Vapnik 1998)	84.1 $\pm$ 0.7	77.9 $\pm$ 0.8	67.0 $\pm$ 0.3	46.8 $\pm$ 1.1	**44.1** $\pm$ 0.0	64.0
POEM (Ours)	**86.7** $\pm$ 0.2	**79.2** $\pm$ 0.6	**68.0** $\pm$ 0.2	**49.5** $\pm$ 0.6	44.0 $\pm$ 0.0	**65.5** ($\uparrow$ 1.5%)
SWAD† (Cha et al. 2021)	88.3 $\pm$ 0.3	77.7 $\pm$ 0.3	**70.7** $\pm$ 0.1	49.7 $\pm$ 0.6	46.2 $\pm$ 0.0	66.5
SWAD† + POEM (Ours)	**88.5** $\pm$ 0.2	**79.4** $\pm$ 0.3	70.5 $\pm$ 0.1	**51.5** $\pm$ 0.1	**47.1** $\pm$ 0.0	**67.4** ($\uparrow$ 0.9%)
MIRO† (Cha et al. 2022)	85.4 $\pm$ 0.3	79.1 $\pm$ 0.7	70.7 $\pm$ 0.0	**49.7** $\pm$ 0.2	44.3 $\pm$ 0.2	65.8
MIRO† + POEM (Ours)	**86.7** $\pm$ 0.4	**79.1** $\pm$ 0.2	**71.4** $\pm$ 0.0	**49.3** $\pm$ 0.8	**44.3** $\pm$ 0.2	**66.1** ($\uparrow$ 0.3%)
MIRO + SWAD†	87.7 $\pm$ 0.3	78.5 $\pm$ 0.3	71.3 $\pm$ 0.1	51.0 $\pm$ 0.2	46.9 $\pm$ 0.0	67.1
MIRO + SWAD† + POEM (Ours)	**88.5** $\pm$ 0.1	**79.5** $\pm$ 0.3	**71.7** $\pm$ 0.1	**51.6** $\pm$ 0.0	**47.1** $\pm$ 0.0	**67.7** ($\uparrow$ 0.6%)

† indicates our reproduced experiments based on the DomainBed settings. $\uparrow$ indicates the performance gains obtained by POEM.

Table 1. Domain generalization accuracies on the five benchmarks

tures that are extracted from different embeddings. Thus the property is also desirable.

With Property 1, the distributions of embeddings are separated but not orthogonalized. On the other hand, with Property 2, the sample-wise orthogonalization is guaranteed but the distributions can be overlapped. When POEM tries to achieve these both properties, the feature distributions of different embeddings should be separated and orthogonalized, i.e., the polarization of embeddings. In the following section, we visually show the separation of feature distributions of different embeddings, and empirically confirm the zero-forced cosine similarity values between randomly-sampled pair of features from different embeddings.

Based on the understanding of POEM, we informally provide the following claim to explain how POEM achieves the improved generalization capability. First, let us process the singular value decomposition (SVD) of the matrix $\mathbf{M}_j$ formed by the collected feature vectors from j^{th} embedding, i.e., $\mathbf{M}_j = \mathbf{U}_j \mathbf{\Sigma}_j \mathbf{V}_j^T$. Then let us project a feature vector $\mathbf{z}_i^*$ from different i^{th} embedding to the vector space $\mathbf{U}_j \mathbf{\Sigma}_j$. Then the power of the projected feature vector will be zero-forced because the dominant components of $\mathbf{U}_j$ would be orthogonal to $\mathbf{z}_i^*$ due to the polarization of embeddings.

Claim 1. *(Information separation of embeddings)* When feature vector $\mathbf{z}_i^* \sim \mathcal{Z}_i^*$ is projected to the space formed by the features from different j^{th} embedding, then the power of the projected feature is minimized to zero:

$$||\mathbf{z}_i^* \mathbf{U}_j \mathbf{\Sigma}_j||^2 \simeq 0. \tag{12}$$

It implies the information separation between embeddings, i.e., for the DG task, the features for the domain-classifying embedding are zero-forced in the category-classifying embedding space. In other words, features from the category-classifying embedding are domain-invariant, or do not contain the information for domain-classification. Otherwise, the domain-specific features contained in the category-classifying features will remain non-zero when projected to the domain-classifying embedding. The formal analysis of POEM remains as a future work.

Experimental Results

Experiment Settings

Benchmarks: We have conducted extensive experiments to evaluate POEM on the five popular domain generalization (DG) benchmarks based on PACS (Li et al. 2017) (containing 9,991 images, 7 classes and 4 domains), VLCS (Fang, Xu, and Rockmore 2013) (containing 10,729 images, 5 classes, and 4 domains), OfficeHome (Venkateswara et al. 2017) (containing 15,588 images, 65 classes, and 4 domains), TerraIncognita (Beery, Van Horn, and Perona 2018) (containing 24,788 images, 10 classes, and 4 domains), and DomainNet (Peng et al. 2019) (containing 586,575 images, 345 classes, and 6 domains). For each benchmark, if a domain is selected as the target domain, then the remaining domains are designated to be the training source domains. We test all cases for each target domain and take the average of accuracies. Our experiments are run on the DomainBed framework of (Gulrajani and Lopez-Paz 2021), which is publicly released under the MIT license to evaluate the existing DG methods[1]. We follow the training and evaluation protocols of DomainBed of (Gulrajani and Lopez-Paz 2021). Also, we follow the data splitting introduced by the work of SWAD (Cha et al. 2021).

Experiments Details: We set the number of training iterations of POEM to be the same as the experiments done in (Cha et al. 2021), i.e., PACS: 5,000, VLCS: 5,000, OfficeHome: 5,000, TerraIncognita: 5,000, DomainNet: 15,000 iterations. When POEM is combined with MIRO of (Cha et al. 2022), twice number of iterations are used, i.e., PACS: 10,000, VLCS: 10,000, OfficeHome: 10,000, TerraIncognita: 10,000, DomainNet: 30,000. For every elementary embedding, we adopt the ResNet50 architecture of (He et al. 2016) which is pretrained on the ImageNet dataset (Russakovsky et al. 2015) with freezing batch normalization parameters. A mini-batch contains 32 images from each source domain in benchmark datasets. Due to the lack of memory in our simulation, a mini-batch for the DomainNet case contains 20 images for each source domain. For all benchmarks, we have searched proper hyperparameters that include learning rates, dropout ratios, and weight decay rates for both elementary embeddings. Details of the hyperparameter values and the optimizers are described in Supplementary.

Methods to be considered: Similar to other cutting-edge algorithms, POEM is built upon the ERM framework of (Vapnik 1998). We denote the vanilla version of our method based on ERM as **POEM**. Also, the concept of POEM can be plugged in with other approaches. We evaluate **SWAD + POEM**, **MIRO + POEM**, and **MIRO + SWAD + POEM**, by combining POEM with the most promising DG approaches. POEM contains two elementary embeddings where one is for category-classifying, and the other is for domain-classifying. SWAD + POEM adopts the optimization process for finding flat minima only for category-classifying embedding of POEM. MIRO + POEM employs the pretrained oracle network to maximize the mutual information between the features from the oracle and both elementary embeddings of POEM. MIRO + SWAD + POEM combines all three methods. We described details of hyperparameters for SWAD and MIRO in Supplementary.

Performance on Target Domain

In Table 1, the DG performance of POEM, SWAD + POEM, MIRO + POEM, MIRO + SWAD + POEM are compared with the existing methods. The accuracies are obtained by taking the averages over three trials. We emphasize that POEM yields consistent performance gains when combined with ERM, SWAD, and MIRO. Specifically, POEM obtains the averaged gains by +1.5% beyond ERM and by +0.9% beyond SWAD. Also, POEM yields an extra gain by +0.6% beyond MIRO + SWAD. The results confirm that POEM enlightens a unique way to enhance the domain-invariance of representations beyond cutting-edge algorithms. Performance on source domains are presented in Supplementary.

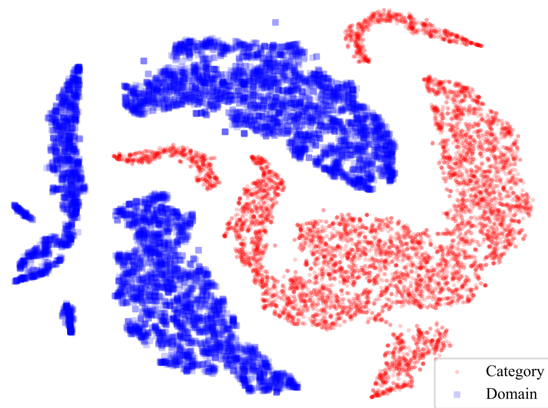


Figure 2. Visualization of features with embedding labels

t-SNE Visualization of Embeddings

To visualize the orthogonality between elementary embeddings, the t-SNE analysis of (Van der Maaten and Hinton 2008) is conducted. We consider the experiment case of the VLCS benchmark where the target domain is the 'SUN09' domain. Fig. 2 is the t-SNE plot of features from the category-classifying embeddings and the domain-classifying embedding, which are colored by red and blue, respectively. This visualization clearly shows that POEM separates elementary embeddings without any overlaps.

Entropy Analysis of Embeddings

For quantifying the domain-invariance of category-classifying features, we calculate the cross-entropy values when category-classifying features are used to classify domains. For the category embedding of POEM, the classifiers for domains are not prepared so we compute the domain-wise centroids $\{\mathbf{c}_k\}_{k=1}^N$ of features and utilize them as the classifiers for domains. After obtaining the domain centroids, the cross-entropy loss is calculated by measuring the probability based on the Euclidean distance between feature vectors and centroids, i.e.,

$$P(y=k \mid \mathbf{x}) = \frac{\exp\big(-d(f_z(\mathbf{x}\,;\theta_z),\mathbf{c}_k)\big)}{\sum_{l=1}^N \exp\big(-d(f_z(\mathbf{x}\,;\theta_z),\mathbf{c}_l)\big)}, \quad (13)$$

where N is the number of source domains, $d(\cdot,\cdot)$ means the Euclidean distance, and z is the index of the category embedding. In addition, we train the ERM-based model on the same source domains and compute the cross-entropy loss with the same way. As shown in Table 4, the category features from POEM show higher cross-entropy values when compared to the values of ERM. It indicates that POEM discards the domain-related information from the category embedding. OfficeHome, TerraIncognita, DomainNet is denoted as OH, Terra, DN, due to the space limit.

[1]Code is available at github.com/JoSangYoung/Official-POEM

Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Average
ERM†	84.09 $\pm$ 0.7	77.88 $\pm$ 0.8	67.00 $\pm$ 0.3	46.78 $\pm$ 1.1	44.13 $\pm$ 0.0	64.0
POEM only with $\mathcal{L}_s$	84.86 $\pm$ 0.3	78.29 $\pm$ 0.5	67.12 $\pm$ 0.3	47.21 $\pm$ 2.0	43.78 $\pm$ 0.2	64.3
POEM only with $\mathcal{L}_d$	84.96 $\pm$ 0.3	78.28 $\pm$ 0.4	67.45 $\pm$ 0.4	47.82 $\pm$ 0.8	**44.04** $\pm$ 0.1	64.5
POEM	**86.73** $\pm$ 0.3	**79.24** $\pm$ 0.6	**67.96** $\pm$ 0.2	**49.48** $\pm$ 0.6	**44.03** $\pm$ 0.0	**65.5**

† indicates our implementation

Table 2. Effect of loss functions in our method based on ERM over three trials

Method	PACS	VLCS	OH	Terra	DN	Avg
ERM	0.22	0.27	0.09	0.14	0.06	0.16
POEM	3.8e-05	1.0e-04	1.5e-04	2.9e-04	1.4e-03	3.94e-04

Table 3. Averaged cosine similarity between category-classifying features and domain-classifying features

Method	PACS	VLCS	OH	Terra	DN	Avg
ERM	2.65	2.80	1.13	1.85	0.81	1.85
POEM	2.98	4.01	1.41	2.12	0.68	2.24

Table 4. Averaged cross-entropy for classifying domains with category-classifying features in 5 benchmark datasets

Orthogonality Analysis of Embeddings

To confirm the orthogonality of different elementary embeddings of POEM, we compute the averaged cosine similarity values by randomly sampling two features from category- and domain-classifying embeddings. Table 3 shows averaged cosine similarities in 5 benchmark datasets, by considering more than 1,000 samples for each domain. As a counterpart, we prepare the ERM model for classifying image categories, and also prepare a separate ERM model that classifies image domains across different categories. Then the averaged cosine similarity values are computed in the same way as POEM cases. OfficeHome, TerraIncognita, DomainNet are denoted as OH, Terra, DN, respectively. The result shows that POEM makes elementary embeddings more orthogonal when compared to ERM for all benchmarks. Note that ERM shows larger cosine similarities on the PACS and VLCS cases. By zero-forcing the cosine similarities, POEM indeed shows more considerable gains in that benchmarks when compared to others, as reported in Table 1.

Ablation Analysis

We conduct ablation studies of the loss terms of POEM. Table 2 shows the performance gain in the addition of the proposed loss functions. POEM only with $\mathcal{L}_s$ makes the cosine similarity between two paired features from a single image be zero. The performance gain for POEM only with $\mathcal{L}_s$ is +0.3% when compared to ERM. The gain is quite small because the loss term $\mathcal{L}_s$ cannot separate the clusters of features from two embeddings. Only with the discrimination loss $\mathcal{L}_d$, a moderated performance gain by +0.5% is obtained beyond ERM. However, the gain is not yet considerable because the loss cannot make two elementary embeddings orthogonal. Finally, POEM with both loss terms eventually separates two embeddings in two orthogonal directions so that the considerable performance gain is achieved, i.e., +1.5% beyond ERM.

Complexity Analysis

POEM prepares two elementary embeddings, but once training is over, POEM drops the domain-classifying embedding and utilizes only the category-classifying embedding for inference. It means that POEM shows the same level of memory and computational costs during testing when compared to ERM. When we compare POEM with SWAD of (Cha et al. 2021), which is a promising DG method, SWAD is required to store an additional moving average model during iterations. It means that SWAD requires twice the number of parameters during the training phase, i.e., the same as the costs of POEM. MIRO of (Cha et al. 2022) shows the same level of costs as ERM during training, but MIRO requires additional costs for the pretraining of the oracles.

Conclusion

For achieving the robustness of the deep visual models on the out-of-distribution problem, we propose a method called POEM with a set of elementary embeddings where the elementary embeddings are trained to be disentangled with each other. We show that considerable performance gains can be achieved by combining POEM with other cutting-edge DG methods, including ERM, SWAD, and MIRO.

Discussion

POEM is possibly extended to the more complicated generalization scenarios. For example, the medical image classification task may include a variety of dimensions such as diseases, organs, patients, and types of imaging equipment. Then POEM with an embedding for each dimension possibly handles the generalization tasks across multiple dimensions. We leave it as a future work. Specifically, we expect that POEM enables training the disease-related embedding invariant to the other factors, i.e., patients or medical imaging equipment. When considering the detection task for road objects, images would be diverse during daytime and night-time. By employing the day/night-classifying embedding, the concept of POEM can be used to train the encoder to extract the day/night-invariant features by utilizing the day/night-classifying features.

Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)), (No. 2021-0-02201, Federated Learning for Privacy Preserving Video Caching Networks), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1C1C1012797), and the Settlement Research Fund (1.200101.01) of UNIST (Ulsan National Institute of Science & Technology).

References

Albuquerque, I.; Naik, N.; Li, J.; Keskar, N.; and Socher, R. 2020. Improving out-of-distribution generalization via multi-task self-supervised pretraining. *arXiv preprint arXiv:2003.13525*.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant Risk Minimization. *arXiv preprint arXiv:1907.02893*.

Arpit, D.; Wang, H.; Zhou, Y.; and Xiong, C. 2022. Ensemble of Averages: Improving Model Selection and Boosting Performance in Domain Generalization. *Advances in Neural Information Processing Systems (NeurIPS)*.

Beery, S.; Van Horn, G.; and Perona, P. 2018. Recognition in Terra Incognita. *Proceedings of the European conference on computer vision (ECCV)*.

Blanchard, G.; Deshmukh, A. A.; Dogan, U.; Lee, G.; and Scott, C. 2021. Domain Generalization by Marginal Transfer Learning. *Journal of Machine Learning Research (JMLR)*.

Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; and Erhan, D. 2016. Domain Separation Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 29.

Carlucci, F. M.; D'Innocente, A.; Bucci, S.; Caputo, B.; and Tommasi, T. 2019. Domain Generalization by Solving Jigsaw Puzzles. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cha, J.; Chun, S.; Lee, K.; Cho, H.-C.; Park, S.; Lee, Y.; and Park, S. 2021. SWAD: Domain Generalization by Seeking Flat Minima. *Advances in Neural Information Processing Systems (NeurIPS)*.

Cha, J.; Lee, K.; Park, S.; and Chun, S. 2022. Domain Generalization by Mutual-Information Regularization with Pretrained Models. *European Conference on Computer Vision (ECCV)*.

Dou, Q.; Coelho de Castro, D.; Kamnitsas, K.; and Glocker, B. 2019. Domain Generalization via Model-Agnostic Learning of Semantic Features. *Advances in Neural Information Processing Systems (NeurIPS)*.

Fang, C.; Xu, Y.; and Rockmore, D. N. 2013. Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. *International Conference on Computer Vision (ICCV)*.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised Domain Adaptation by Backpropagation. *International Conference on Machine Learning (ICML)*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research (JMLR)*.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research (JMLR)*.

Gulrajani, I.; and Lopez-Paz, D. 2021. In Search of Lost Domain Generalization. *International Conference on Learning Representations (ICLR)*.

Haralick, R. M.; Shanmugam, K.; and Dinstein, I. H. 1973. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Z.; Wang, H.; Xing, E. P.; and Huang, D. 2020. Self-Challenging Improves Cross-Domain Generalization. *European Conference on Computer Vision (ECCV)*.

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging Weights Leads to Wider Optima and Better Generalization. *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-Distribution Generalization via Risk Extrapolation (REx). *International Conference on Machine Learning (ICML)*.

Kullback, S.; and Leibler, R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*.

Lam, S. W.-C. 1996. Texture feature extraction using gray level gradient based co-occurence matrices. *IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929)*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, Broader and Artier Domain Generalization. *International Conference on Computer Vision (ICCV)*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2018a. Learning to Generalize: Meta-Learning for Domain Generalization. *AAAI Conference on Artificial Intelligence (AAAI)*.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain Generalization with Adversarial Feature Learning. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y.; Gong, M.; Tian, X.; Liu, T.; and Tao, D. 2018c. Domain Generalization via Conditional Invariant Representations. *AAAI Conference on Artificial Intelligence (AAAI)*.

Mancini, M.; Bulo, S. R.; Caputo, B.; and Ricci, E. 2018. Best sources forward: domain generalization through source-specific nets. *International Conference on Image Processing (ICIP)*.

Motiian, S.; Piccirilli, M.; Adjeroh, D. A.; and Doretto, G. 2017. Unified Deep Supervised Domain Adaptation and Generalization. *International Conference on Computer Vision (ICCV)*.

Nam, H.; Lee, H.; Park, J.; Yoon, W.; and Yoo, D. 2021. Reducing Domain Gap by Reducing Style Bias. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment Matching for Multi-Source Domain Adaptation. *International Conference on Computer Vision (ICCV)*.

Piratla, V.; Netrapalli, P.; and Sarawagi, S. 2020. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. *International Conference on Machine Learning (ICML)*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. *International Conference on Learning Representations (ICLR)*.

Sun, B.; and Saenko, K. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. *European Conference on Computer Vision (ECCV)*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: wiley.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, H.; He, Z.; Lipton, Z. C.; and Xing, E. P. 2019. Learning Robust Representations by Projecting Superficial Statistics Out. *International Conference on Learning Representations (ICLR)*.

Wang, S.; Yu, L.; Li, C.; Fu, C.-W.; and Heng, P.-A. 2020. Learning from Extrinsic and Intrinsic Supervisions for Domain Generalization. *European Conference on Computer Vision (ECCV)*.

Xu, M.; Zhang, J.; Ni, B.; Li, T.; Wang, C.; Tian, Q.; and Zhang, W. 2020. Adversarial Domain Adaptation with Domain Mixup. *AAAI Conference on Artificial Intelligence (AAAI)*.

Zhang, M.; Marklund, H.; Dhawan, N.; Gupta, A.; Levine, S.; and Finn, C. 2021. Adaptive Risk Minimization: Learning to Adapt to Domain Shift. *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021a. Domain Adaptive Ensemble Learning. *IEEE Transactions on Image Processing*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021b. Domain Generalization with MixStyle. *International Conference on Learning Representations (ICLR)*.

Supplementary Material for POEM: Polarization of Embeddings for Domain-Invariant Representations

In this supplementary material, we provide the details of the experiment setting and other empirical results, such as source domain accuracies and the learning trend of the introduced loss terms.

Details of Experiment Setting

Experimental Environment

All experiments are conducted by utilizing NVIDIA Quadro RTX 8000, 400GB RAM, and Xeon(R) Gold 5218R CPU @ 2.10GHz with ubuntu 20.04 with python 3.8.12, PyTorch 1.7.1, Torchvision 0.8.2, and CUDA 11.0. Our source code is partially based on the codes of DomainBed (Gulrajani and Lopez-Paz 2021), SWAD (Cha et al. 2021) and MIRO (Cha et al. 2022).

Method	**Hyperparameter**	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet
POEM	Learning rate	5e-5	1e-5	1e-5	3e-5	3e-5
	Dropout ratio	0.1	0.1	0.5	0.5	0.5
	Weight decay	1e-6	1e-4	1e-4	1e-4	1e-6
SWAD† + POEM	Learning rate	5e-5	5e-5	3e-5	5e-5	3e-5
	Dropout ratio	0.1	0.0	0.5	0.0	0.5
	Weight decay	1e-6	0	1e-4	0	1e-6
	N_s	3	3	3	3	3
	N_e	6	6	6	6	6
	r	1.3	1.2	1.3	1.3	1.3
MIRO† + POEM	Learning rate	3e-5	1e-5	1e-5	3e-5	3e-5
	Dropout ratio	0.1	0.1	0.5	0	0.1
	Weight decay	1e-4	1e-4	1e-4	1e-4	1e-6
	λ for category embedding	0.01	0.01	0.1	0.1	0.1
	λ for domain embedding	0.01	0	0.01	0	0
MIRO + SWAD† + POEM	Learning rate	3e-5	1e-5	1e-5	3e-5	3e-5
	Dropout ratio	0.1	0.1	0.5	0	0.1
	Weight decay	1e-4	1e-4	1e-4	1e-4	1e-6
	N_s	3	3	3	3	3
	N_e	6	6	6	6	6
	r	1.3	1.2	1.3	1.3	1.3
	λ for category embedding	0.01	0.01	0.1	0.1	0.1
	λ for domain embedding	0.01	0	0.01	0	0

† indicates our implementation

Table 5. Hyperparameter setting for each algorithm and dataset

Details of Hyperparameter Settings

For all benchmarks, we search hyperparameters of the learning rate, the dropout ratio, and the weight decay for both domain-classifying and category-classifying embeddings. They are grid searched in [1e-5, 3e-5, 5e-5], [0.0, 0.1, 0.5], [1e-4, 1e-6], respectively. Also, when combined with SWAD, the patient parameter N_s, the overfitting patient parameter N_e, and the tolerance rate r are set to be 3, 6, and 1.3, respectively. In the case of VLCS, the tolerance rate r is particularly set to be 1.2, following the original setting of SWAD. When combining MIRO, the regularization coefficient λ is set to be 0.1 for the PACS and VLCS experiments and 0.01 for other cases by following the original setting of MIRO. The regularization coefficient for domain-classifying embedding in conjunction with MIRO has been grid searched in [0, 0.1, 0.01] for each dataset. When λ is zero, it indicates that domain-classifying embedding is set to be ERM. Table 5 shows the chosen hyperparameter setting for all benchmarks and methods.

Details of Experimental Results

Target Domain Accuracies

The performance on the target domains provided in the main paper shows the averaged accuracies across target domains for each benchmark over three trials. The target accuracies on all test domains are shown in the tables from Table 6 to Table 10.

Method	Art painting	Cartoon	Photo	Sketch	Average
POEM	85.34 ± 2.28	82.16 ± 0.77	97.01 ± 0.16	82.41 ± 1.14	86.73
SWAD† + POEM	90.12 ± 0.51	83.62 ± 0.33	97.78 ± 0.05	82.60 ± 0.81	88.53
MIRO† + POEM	87.29 ± 1.22	81.50 ± 0.74	97.65 ± 0.24	80.30 ± 0.03	86.68
MIRO + SWAD† + POEM	89.35 ± 0.70	83.02 ± 0.40	98.20 ± 0.07	83.21 ± 0.48	88.45

† indicates our implementation

Table 6. Domain generalization accuracies on target domains in the PACS benchmark

Method	Caltech101	LabelMe	SUN09	VOC2007	Average
POEM	97.91 ± 0.26	66.71 ± 0.77	76.12 ± 0.48	76.23 ± 2.02	79.24
SWAD† + POEM	98.35 ± 0.24	64.22 ± 0.26	76.12 ± 0.48	79.01 ± 0.44	79.43
MIRO† + POEM	98.44 ± 0.33	66.44 ± 0.87	73.86 ± 0.83	77.55 ± 0.56	79.12
MIRO + SWAD† + POEM	98.91 ± 0.08	64.63 ± 0.17	75.63 ± 0.36	78.96 ± 0.46	79.53

† indicates our implementation

Table 7. Domain generalization accuracies on target domains in the VLCS benchmark

Method	Art	Clipart	Product	Realworld	Average
POEM	64.06 ± 0.21	53.92 ± 0.73	76.21 ± 0.41	77.66 ± 0.40	67.96
SWAD† + POEM	67.08 ± 0.31	57.01 ± 0.36	78.21 ± 0.45	79.59 ± 0.46	70.47
MIRO† + POEM	69.60 ± 0.38	54.67 ± 0.67	79.39 ± 0.49	81.78 ± 0.25	71.36
MIRO + SWAD† + POEM	69.46 ± 0.23	55.41 ± 0.13	79.98 ± 0.14	82.03 ± 0.16	71.73

† indicates our implementation

Table 8. Domain generalization accuracies on target domains in the OfficeHome benchmark

Method	location100	location38	location43	location46	Average
POEM	59.28 ± 1.14	38.83 ± 1.43	58.93 ± 0.97	40.86 ± 1.36	49.48
SWAD† + POEM	57.80 ± 0.20	47.23 ± 0.89	58.99 ± 0.33	41.97 ± 0.62	51.50
MIRO† + POEM	57.10 ± 4.22	44.13 ± 0.71	57.16 ± 1.56	38.68 ± 2.05	49.27
MIRO + SWAD† + POEM	59.89 ± 0.72	45.47 ± 0.22	60.37 ± 0.23	41.10 ± 0.47	51.71

† indicates our implementation

Table 9. Domain generalization accuracies on target domains in the TerraIncognita benchmark

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
POEM	64.41 ± 0.15	21.43 ± 0.30	49.92 ± 0.28	13.22 ± 0.22	62.22 ± 0.09	52.97 ± 0.09	44.03
SWAD† + POEM	66.67 ± 0.08	23.49 ± 0.02	54.26 ± 0.06	15.84 ± 0.13	65.69 ± 0.09	56.47 ± 0.08	47.07
MIRO† + POEM	66.73 ± 0.20	23.44 ± 0.06	53.96 ± 0.07	15.29 ± 0.06	67.19 ± 0.72	55.44 ± 0.20	47.01
MIRO + SWAD† + POEM	66.72 ± 0.19	23.46 ± 0.05	54.01 ± 0.12	15.27 ± 0.08	67.88 ± 0.04	55.10 ± 0.14	47.07

† indicates our implementation

Table 10. Domain generalization accuracies on target domains in the DomainNet benchmark

Source Domain Accuracies

We also measure the performance of POEM on the source domains. For each source domain, we split the dataset into 80% of the samples for the training set and 20% for the validation set. We evaluate two performance metrics, i.e., image-category classification accuracy and image-domain classification accuracy. When evaluating the category classification task, the elementary embedding to classify image categories is used. Otherwise, when evaluating the domain classification task, the secondary embedding to classify image domains is used. The averaged accuracies of POEM for classifying image categories and domains over three trials are shown in Table 11. We confirm the slight performance gain for all experiment cases. It implies that the joint training of the multiple elementary embeddings via POEM does not harm the training of the individual embedding. To provide the details of the results, the validation accuracies for category-classification and domain-classification for each source domain are shown in the tables from Table 12 to 16 and Tables 17 to 21, respectively.

Accuracies for classifying image categories

Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Average
ERM†	96.99 ± 0.1	86.21 ± 0.1	80.38 ± 0.1	91.63 ± 0.1	60.00 ± 0.1	83.31
POEM	97.14 ± 0.1	86.91 ± 0.1	80.85 ± 0.04	92.16 ± 0.1	60.82 ± 0.1	83.58

Accuracies for classifying image domains

Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Average
ERM†	99.02 ± 0.1	94.10 ± 0.1	85.13 ± 0.1	99.95 ± 0.02	89.26 ± 0.1	93.49
POEM	98.98 ± 0.1	93.99 ± 0.1	85.53 ± 0.1	99.96 ± 0.02	89.32 ± 0.1	93.56

† indicates our implementation

Table 11. Averaged validation accuracies on source domains for each benchmark

Method	Art painting	Cartoon	Photo	Sketch	Average
ERM†	97.61 ± 0.1	96.82 ± 0.1	96.29 ± 0.1	97.24 ± 0.2	96.99 ± 0.1
POEM	97.67 ± 0.1	96.73 ± 0.1	96.29 ± 0.1	97.86 ± 0.2	97.14 ± 0.1

† indicates our implementation

Table 12. Validation accuracies for image-category classification on source domains of PACS benchmark

Method	Caltech101	LabelMe	SUN09	VOC2007	Average
ERM†	81.31 ± 0.1	90.79 ± 0.1	87.33 ± 0.1	85.43 ± 0.1	86.21 ± 0.1
POEM	81.66 ± 0.2	91.30 ± 0.2	88.03 ± 0.1	86.69 ± 0.1	86.92 ± 0.1

† indicates our implementation

Table 13. Validation accuracies for image-category classification on source domains of VLCS benchmark

Method	Art	Clipart	Product	Realworld	Average
ERM†	83.68 ± 0.2	81.18 ± 0.3	77.48 ± 0.1	79.17 ± 0.2	80.38 ± 0.1
POEM	84.23 ± 0.1	80.83 ± 0.2	77.56 ± 0.1	80.77 ± 0.3	80.85 ± 0.1

† indicates our implementation

Table 14. Validation accuracies for image-category classification on source domains of OfficeHome benchmark

Method	location100	location38	location43	location46	Average
ERM†	90.40 ± 0.1	91.98 ± 0.1	91.06 ± 0.1	93.08 ± 0.1	91.63 ± 0.1
POEM	90.79 ± 0.1	92.20 ± 0.1	92.00 ± 0.1	93.66 ± 0.1	92.16 ± 0.1

† indicates our implementation

Table 15. Validation accuracies for image-category classification on source domains of TerraIncognita benchmark

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
ERM†	57.89 ± 0.1	65.35 ± 0.1	59.90 ± 0.1	61.27 ± 0.1	56.84 ± 0.1	58.77 ± 0.1	60.00 ± 0.1
POEM	58.04 ± 0.1	66.22 ± 0.1	60.79 ± 0.1	61.40 ± 0.1	58.52 ± 0.1	59.94 ± 0.1	60.82 ± 0.1

† indicates our implementation

Table 16. Validation accuracies for image-category classification on source domains of DomainNet benchmark

Method	Art painting	Cartoon	Photo	Sketch	Average
ERM†	99.58 ± 0.1	99.07 ± 0.1	99.63 ± 0.2	97.81 ± 0.1	99.02 ± 0.1
POEM	99.52 ± 0.1	98.91 ± 0.1	99.27 ± 0.1	98.21 ± 0.1	98.98 ± 0.1

† indicates our implementation

Table 17. Validation accuracies for image-domain classification on source domains of PACS benchmark

Method	Caltech101	LabelMe	SUN09	VOC2007	Average
ERM†	89.56 ± 0.2	93.69 ± 0.1	97.16 ± 0.1	96.00 ± 0.1	94.10 ± 0.1
POEM	89.19 ± 0.1	93.52 ± 0.1	97.04 ± 0.1	96.19 ± 0.1	93.99 ± 0.1

† indicates our implementation

Table 18. Validation accuracies for image-domain classification on source domains of VLCS benchmark

Method	Art	Clipart	Product	Realworld	Average
ERM†	87.80 ± 0.2	76.49 ± 0.1	82.27 ± 0.1	93.97 ± 0.1	85.13 ± 0.1
POEM	88.34 ± 0.1	76.29 ± 0.1	83.17 ± 0.1	94.31 ± 0.1	85.53 ± 0.1

† indicates our implementation

Table 19. Validation accuracies for image-domain classification on source domains of OfficeHome benchmark

Method	location100	location38	location43	location46	Average
ERM†	99.93 ± 0.0	100.00 ± 0.0	99.95 ± 0.0	99.93 ± 0.0	99.95 ± 0.0
POEM	100.00 ± 0.0	99.93 ± 0.0	99.93 ± 0.0	99.98 ± 0.0	99.96 ± 0.0

† indicates our implementation

Table 20. Validation accuracies for image-domain classification on source domains of TerraIncognita benchmark

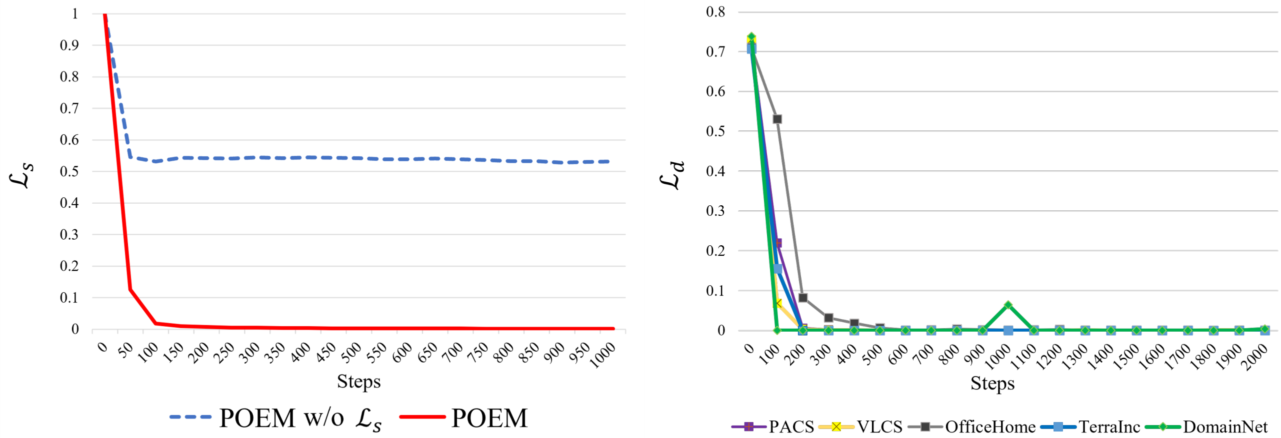
Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
ERM†	90.85 ± 0.1	87.95 ± 0.1	90.79 ± 0.1	84.21 ± 0.3	91.83 ± 0.2	89.93 ± 0.3	89.26 ± 0.1
POEM	90.77 ± 0.3	87.92 ± 0.1	90.76 ± 0.2	84.53 ± 0.2	91.51 ± 0.3	90.43 ± 0.3	89.32 ± 0.1

† indicates our implementation

Table 21. Validation accuracies for image-domain classification on source domains of DomainNet benchmark

Learning Trend of Disentangling and Discrimination Loss Terms

The learning trend of the key loss terms, which are the disentangling loss (or similarity loss) $\mathcal{L}_s$ and the discrimination loss $\mathcal{L}_d$ are illustrated in Fig. 3. As presented in Fig. 3a, the averaged cosine similarity between the category embedding and the domain embedding over VLCS domains decreases rapidly (See the solid line colored by red). It means that the feature vectors from the two elementary embeddings become orthogonal. In contrast, when we drop the similarity loss term in the training of POEM, the averaged cosine similarity between elementary embeddings over VLCS domains is not zero-forced (See the dotted line colored



(a) Trainig similarity loss $\mathcal{L}_s$

(b) Training discriminator loss L_d

Figure 3. Learning trend of the component losses of POEM

by blue). It shows that the orthogonality between the features from image- and domain-classifying elementary embeddings are not trivially achieved without POEM's similarity loss. Fig. 3b shows the learning trend of the discrimination loss for the five benchmarks. The suppressed loss values mean that the features from different embeddings become distinctive so that POEM can discriminate the features from different embeddings.

Cross-Entropy between Category Features and Domain Labels

In the main paper, we show the cross-entropy values when the category-classifying features are used to estimate the source domain. The result emphasizes the POEM's category-classification feature is not effective in classifying domains, i.e., it implies the domain-invariance of features. Here, we attach all calculated cross-entropy values. Every experiment with ERM and POEM uses the same random seed so that the validation set is equal. From Tables 22 to 26, we show the cross-entropy values. In most cases, it appears that POEM shows larger cross-entropy values than ERM which indicates that POEM learns a more domain-invariant category-classifying elementary embedding than ERM.

Method	Art painting	Cartoon	Photo	Sketch	Average
ERM†	2.58	2.31	1.77	3.19	2.65
POEM	3.26	2.88	1.81	3.98	2.98

† indicates our implementation

Table 22. Cross-Entropy for source domain data samples on PACS benchmark

Method	Caltech101	LabelMe	SUN09	VOC2007	Average
ERM†	2.53	1.77	2.94	3.97	2.80
POEM	4.46	2.51	4.06	5.01	4.01

† indicates our implementation

Table 23. Cross-Entropy for source domain data samples on VLCS benchmark

Method	Art	Clipart	Product	Realworld	Average
ERM†	1.02	1.21	1.15	1.13	1.13
POEM	1.18	1.36	1.35	1.76	1.41

† indicates our implementation

Table 24. Cross-Entropy for source domain data samples on OfficeHome benchmark

Method	location100	location38	location43	location46	Average
ERM†	1.27	1.98	2.29	1.86	1.85
POEM	1.42	3.31	2.32	1.44	2.12

† indicates our implementation

Table 25. Cross-Entropy for source domain data samples on TerraIncognita benchmark

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
ERM†	0.71	0.91	0.72	1.16	0.59	0.78	0.81
POEM	0.59	0.71	0.58	1.08	0.48	0.63	0.68

† indicates our implementation

Table 26. Cross-Entropy for source domain data samples on DomainNet benchmark