Fuse and Attend: Generalized Embedding Learning for Art and Sketches

Ujjal Kr Dutta[0000−0002−0470−5521]

Myntra, Bengaluru, India
ukdacad@gmail.com

Abstract. While deep Embedding Learning approaches have witnessed widespread success in multiple computer vision tasks, the state-of-the-art methods for representing natural images need not necessarily perform well on images from other domains, such as paintings, cartoons, and sketch. This is because of the huge shift in the distribution of data from across these domains, as compared to natural images. Domains like sketch often contain sparse informative pixels. However, recognizing objects in such domains is crucial, given multiple relevant applications leveraging such data, for instance, sketch to image retrieval. Thus, achieving an Embedding Learning model that could perform well across multiple domains is not only challenging, but plays a pivotal role in computer vision. To this end, in this paper, we propose a novel Embedding Learning approach with the goal of generalizing across different domains. During training, given a query image from a domain, we employ gated fusion and attention to generate a positive example, which carries a broad notion of the semantics of the query object category (from across multiple domains). By virtue of Contrastive Learning, we pull the embeddings of the query and positive, in order to learn a representation which is robust across domains. At the same time, to teach the model to be discriminative against examples from different semantic categories (across domains), we also maintain a pool of negative embeddings (from different categories). We show the prowess of our method using the DomainBed framework, on the popular PACS (Photo, Art painting, Cartoon, and Sketch) dataset.

1 Introduction

Embedding Learning plays a crucial role in the success of a plethora of computer vision applications, such as object recognition, clustering, content-based retrieval, information extraction, question-answering, semantic understanding, to name a few. It essentially aims at learning a vector/ feature representation (*embedding*) of the raw data in question. The goal is to learn a discriminative embedding space, where similar examples are grouped together, while pushing away the dissimilar ones. The notion of similarity varies with an application, and usually pertains to the semantics of the object contained in an image.

While Embedding Learning has seen remarkable success in natural images, their success in image based applications from other domains like paintings,

cartoons, and sketch, is yet to reach the full potential. This is mainly because of the fact that the pixel level representation of such domains is very different from natural images. While the latter may benefit more from properties like color, texture, and shading, the former may benefit more from spatial and shape information of the objects. At the same time, they may have sparse pixel information.

In this paper, we try to address this question: "***Can we learn a single, common embedding which is good across multiple domains (be it natural photos, art paintings, cartoons, or sketch)***? " To this end, we pose this problem as a Domain Generalization (DG) approach, wherein, the idea is to leverage a number of labeled domains $\mathcal{D}_1, \cdots, \mathcal{D}_{tr}$ to train a model, which could perform well on any unseen domain $\mathcal{D}_{tr+1}$. Following are the motivations for formulating the problem as DG: i) One may have labeled data from across a few domains, and it would be expected that a model trained on those domains should be able to perform well on data from any unseen domain, ii) If the domains are related (say, containing similar semantics), then it could be beneficial to leverage the common information present in them, to arrive at a better representation.

For example, let us assume that we have labeled data from 3 domains: natural photos, art paintings, and cartoons, with a common set of semantic categories among them. If we could collectively represent the *global information* for a semantic category, say, dog, and force the embedding of a dog image from any domain to lie close to that *global information*, then we could perhaps make our embedding learning model robust even for an unseen domain, such as, sketch. By global information, we refer to those attributes, which are essential to identify a semantic category (eg, tail, nose, ears of a dog category). To achieve this, we make use of gated-fusion and attention mechanism to form a *positive example* which could capture such *global information*. Then, we make use of a Contrastive Learning loss to align the embeddings of a query and the positive corresponding to the semantic class of the query. At the same time, to ensure that examples of different semantic categories are separated far apart, we also make use of a pool of *negative examples*. We show the merits of our method for the classification task, using the DomainBed framework, on the popular PACS (Photo, Art painting, Cartoon, and Sketch) dataset.

2 Proposed Method

Our proposed method is illustrated in Figure 1. During training, given a query image, we maintain a pool of positive (same semantic class as query, but different domain) and negative examples (different semantic class, irrespective of domain). Given an encoder (eg, ResNet) with some initial model weights, the embedding/ feature of a positive from a domain is computed while being oblivious to the feature of the query. Thus, we call it as an *irresponsible representative*.

It is good if we can update this positive feature (into a *responsible representative*) so as to take into account the feature of the query as well, as both of them belong to the same semantic category, albeit from different domains (hence,

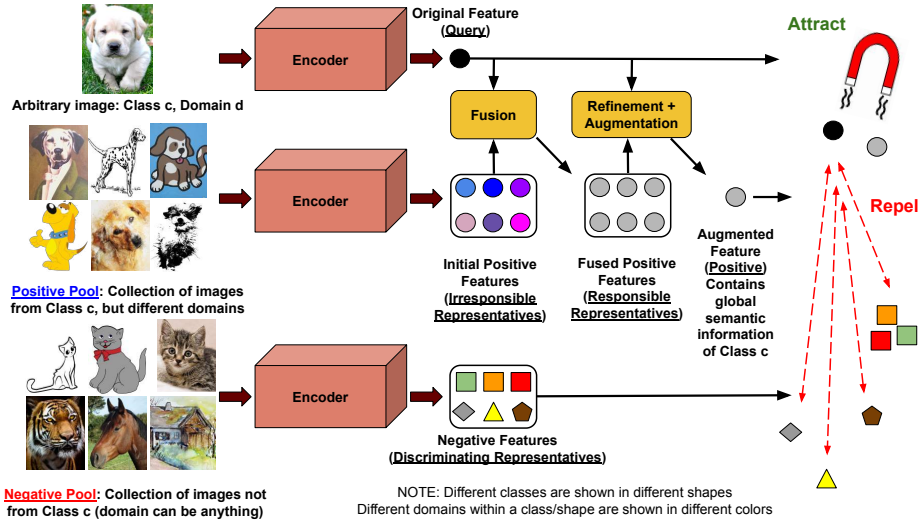

Fig. 1: An illustration of the proposed approach. The figure is best viewed in color.

these features must share some common semantic attribute as well, for instance, a natural image and a sketch image of a dog should contain nose, ear, tail, etc).

We propose a learnable (based on the end loss) gated-fusion mechanism to address this. Specifically, the ***gated-fusion learns what percentage of original domain information of an irresponsible positive to keep, and what percentage of new information from the domain of the query to be learnt, so as to update and obtain a responsible positive***.

Now that we have a set of *responsible representatives* for a query feature, we can use them to attend the query feature, and use the corresponding attention weights to fuse them together with a linear combination to obtain a final augmented positive feature. Intuitively, this augmented positive may be interpreted as containing global semantic information of the category/ class of the query, towards which the query embedding should be pulled closer, to make our encoder robust to domains. This is done with Contrastive Learning. We also push the embeddings for the negative examples away from the query embedding.

Hence, the focus of our work is the contrastive learning based training of a feature Encoder $f_\theta(.)$, that is discriminative enough, of semantically dissimilar content, while being domain robust, so that the generated features could directly be utilized by a classifier $g_\phi(.)$ to correctly predict the class label of an input image. In our method, we employ a Student-Teacher framework to learn such a feature Encoder. The overall architecture of our method is illustrated in Figure 2.

Let us denote an arbitrary example/ raw image from class c, and domain d as $x^d_{y=c}$, and its corresponding embedding/ feature vector obtained using our (Student/ Query) Encoder $f_\theta(.)$, as $f_\theta(x^d_{y=c})$ (Figure 2). Here, $y = c$ denotes the semantic class label for the example, and θ denotes the learnable parameters of the Query Encoder $f_\theta(.)$. Our objective is to learn θ in such a way that the embeddings $f_\theta(x^d_{y=c})$ and $f_\theta(x^{d'}_{y=c})$ for a pair $(x^d_{y=c}, x^{d'}_{y=c})$ of semantically

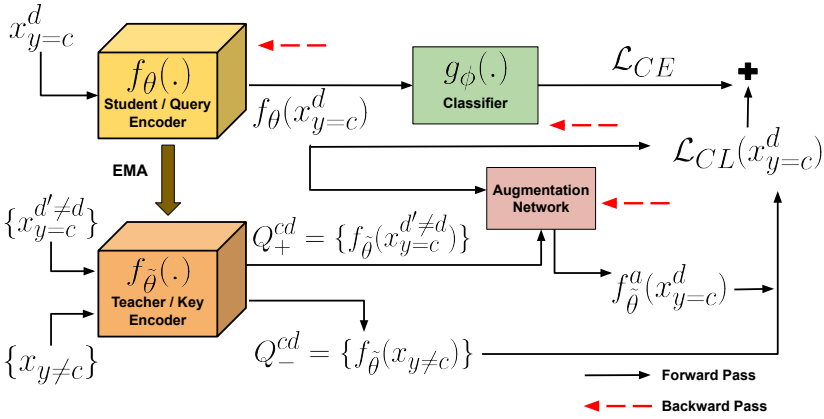

Fig. 2: Architecture of our method.

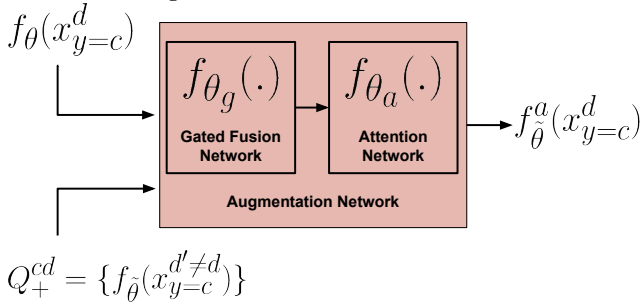

Fig. 3: Blocks of the Augmentation Network.

similar examples (i.e., $y = c$) from two different domains d and d' are grouped together, with the end goal of correctly predicting their class labels with a classifier $g_\phi(.)$, with parameters ϕ. In other words, $f_\theta(.)$ should be *domain robust* (or generalizable).

We maintain a copy of $f_\theta(.)$, denoted as $f_{\tilde{\theta}}(.)$, which we call as the (Teacher/ Key) Encoder. Here, $\tilde{\theta}$ is obtained as an *Exponential Moving Average* (EMA) of θ, i.e., we first initialize $\tilde{\theta}$ as $\tilde{\theta} = \theta$, and then iteratively update it as: $\tilde{\theta} = \mu\tilde{\theta}+(1-\mu)\theta$, $\mu > 0$. In order to make $f_\theta(.)$ *domain robust*, the focus of our method is to leverage $f_{\tilde{\theta}}(.)$ to obtain an augmented feature (positive) $f^a_{\tilde{\theta}}(x^d_{y=c})$ corresponding to $f_\theta(x^d_{y=c})$, and pull the embeddings $f_\theta(x^d_{y=c})$ and $f^a_{\tilde{\theta}}(x^d_{y=c})$ closer to each other.

We obtain $f^a_{\tilde{\theta}}(x^d_{y=c})$ in such a way that it could capture a broad notion of the semantics of the class $y = c$, to which $x^d_{y=c}$ belongs. To do so, we maintain a pool/ set $Q^{cd}_+ = \bigcup_{d'\neq d} Q(x^{d'}_{y=c})$, such that $Q(x^{d'}_{y=c}) = \{f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})\}$ is a set of embeddings of examples from the same class as $x^d_{y=c}$, but from domain $d' \neq d$. The detailed blocks of the Augmentation Network from Figure 2 are shown in the Figure 3. The Augmentation Network takes as input the query embedding $f_\theta(x^d_{y=c})$ and the embeddings of Q^{cd}_+, to produce $f^a_{\tilde{\theta}}(x^d_{y=c})$. It consists of two major components: i) gated fusion, and ii) attention network, which we discuss next.

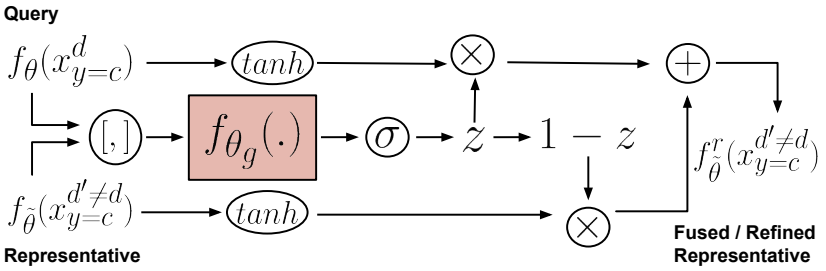

Fig. 4: The Gated Fusion Network takes one (irresponsible) positive representative $f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})$ at a time from Q^{cd}_+, and produces a fused/ refined (responsible) positive representative.

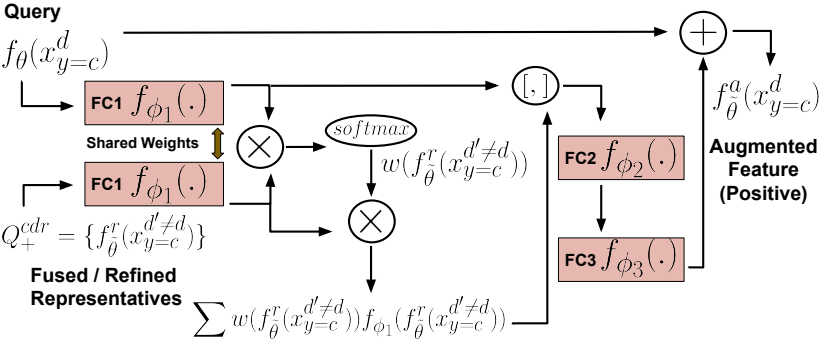

Fig. 5: The Attention Network takes the fused/ refined positive representatives to compute attention weights of the query embedding against them, and forms an augmented feature / positive, corresponding to the query.

2.1 Gated Fusion for Representative Refinement

Q^{cd}_+ is treated as a set of *positive representatives* which take the responsibility of refining their own initial representations (via the Key Encoder) $\{f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})\}$ into $Q^{cdr}_+ = \{f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c})\}$ (we consolidate/abuse the notation by using a single set of same class examples from other domains, to denote the union of sets of examples). This refinement is done by taking into account the representation $f_\theta(x^d_{y=c})$, using a gated fusion mechanism (Figure 4). By considering one representative $f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})$ from Q^{cd}_+ at a time, the refined/ fused representative $f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c})$ is obtained as:

$$f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c}) = z \odot tanh(f_\theta(x^d_{y=c})) + (1-z) \odot tanh(f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})) \quad (1)$$

Here, z is a learnable gate vector obtained as: $z = \sigma(f_{\theta_g}([f_\theta(x^d_{y=c}), f_{\tilde{\theta}}(x^{d'\neq d}_{y=c})]))$, such that θ_g represents parameters of the gated fusion network. Also, $\odot$, $\sigma(.)$, $tanh(.)$ and $[,]$ are the element-wise product, sigmoid, tanh and concatenation operations respectively.

2.2 Attention-based Query Embedding Refinement and Augmentation for Positive Generation

The refined features from Q_+^{cdr} are then attended by $f_\theta(x^d_{y=c})$, one representative at a time (Figure 5), to obtain attention weights: $\{w(f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c})) = softmax(f_{\phi_1}(f_\theta(x^d_{y=c}))^\top f_{\phi_1}(f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c})))\}$. Here, $softmax(.)$ is used to normalize the attention weights across the representatives. Using these weights, a linear combination of the (refined) representatives in Q_+^{cdr} is performed to obtain a single *positive representative* $p_{y=c} = \sum w(f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c}))f_{\phi_1}(f^r_{\tilde{\theta}}(x^{d'\neq d}_{y=c}))$, which is then used to obtain the final augmented feature as:

$$\begin{aligned} f^a_{\tilde{\theta}}(x^d_{y=c}) &= relu(f_\theta(x^d_{y=c}) + \\ & f_{\phi_3}(relu(f_{\phi_2}([f_{\phi_1}(f_\theta(x^d_{y=c})), p_{y=c}])))). \end{aligned} \tag{2}$$

$f^a_{\tilde{\theta}}(x^d_{y=c})$ now captures the accumulated semantic information of the class $y = c$ from across all the domains, and serves as a positive for the query embedding. Here, $\theta_a = \{\phi_1, \phi_2, \phi_3\}$ represents parameters of the attention network.

2.3 Student-Teacher based Contrastive Learning

The Student-Teacher framework arises in our method due to the fact that the Key Encoder serves as a *Teacher* to provide a *positive* embedding (as a *target*/ guidance), with respect to which the query/ *anchor* embedding $f_\theta(x^d_{y=c})$ obtained by the (Student/ Query) Encoder needs to be pulled closer. The notions of (anchor/ query, positive/ key) are quite popular in the *embedding learning* literature, where, in order to group similar examples, triplets of examples in the form of (anchor, positive, negative) are sampled. Usually, the anchor and positive are semantically similar (and thus, need to be pulled closer), while the negative is dissimilar to the other two (and thus, needs to be pushed away).

Now, although we have obtained an augmented feature to serve as the positive for the query encoding $f_\theta(x^d_{y=c})$, we also need to push away embeddings of dissimilar examples, in order to learn a robust, yet semantically discriminative embedding. For this purpose, corresponding to each $x^d_{y=c}$, we also maintain a pool/ set of negatives denoted as $Q^{cd}_- = \bigcup_{\forall d':d'\neq d, d'=d}\{\bigcup_{c'\neq c} Q(x^{d'}_{y=c'})\} = \{f_{\tilde{\theta}}(x_{y\neq c})\}$ (abusing the notation to concisely represent the entire pool of negatives as a single set). Essentially, this pool contains embeddings of example images from other classes, across all the domains (including the same domain as $x^d_{y=c}$). It should be carefully noted that the embeddings in the pools Q^{cd}_+ and Q^{cd}_- are obtained using the Key Encoder $f_{\tilde{\theta}}(.)$.

Keeping in mind the computational aspects, we maintain a dynamically updated queue $Q^{c'd'} \approx Q(x^{d'}_{y=c'})$ of fixed size *queue_sz*, for each class c', from across all domains d'. While training, for each considered example $x^d_{y=c}$, the pools Q^{cd}_+ and Q^{cd}_- are obtained using union from the collection of queues $Q^{c'd'}$. During the network updates, the examples present in a mini-batch are used to replace old examples from the collection $Q^{c'd'}$.

Now, for a given $x^d_{y=c}$, in order to pull its embedding closer to the augmented feature $f^a_{\tilde{\theta}}(x^d_{y=c})$ while moving away the negatives from Q^{cd}_-, we compute an adapted Normalized Temperature-scaled cross entropy (NT-Xent) loss [3] as follows:

$$\mathcal{L}_{CL}(x^d_{y=c}) = \\ -\log \frac{\exp(f_\theta(x^d_{y=c})^\top f^a_{\tilde{\theta}}(x^d_{y=c})/\tau)}{\sum_{f_{\tilde{\theta}}(x_{y\neq c})\in Q^{cd}_-} \exp(f_\theta(x^d_{y=c})^\top f_{\tilde{\theta}}(x_{y\neq c})/\tau)}. \tag{3}$$

Algorithm 1 Pseudocode of RCERM

```
def enQ_deQ(queue,data_emb):# enqueue+dequeue
    queue=torch.cat((queue, data_emb), 0)
    if queue.size(0) > queue_sz:
        queue = queue[-queue_sz:]

''' In algorithms.py '''
class RCERM(Algorithm):
    def __init__(self):# initialize networks
    # featurizer, classifier, f_gated, f_attn
        key_encoder=copy.deepcopy(featurizer)
        # optim.Adam(featurizer, classifier, f_gated, f_attn)
    def update(self,minibatch):
    # minibatch: domainbed styled minibatch
        loss_cl,all_x,all_y=0,None,None
        for id_c in range(nclass):
            for id_d in range(ndomains):
                #data_tensor: minibatch(id_c,id_d)
                q = featurizer(data_tensor)
                all_x = torch.cat((all_x, q), 0)
                all_y = torch.cat((all_y, labels), 0)
                pos_Q,neg_Q=get_posneg_queues(id_c,id_d)
                # Using f_gated+f_attn obtain:
                k=self.get_augmented_batch(q,pos_Q)
                # .detach() k and l2 normalize q,k
                loss_cl += loss_func(q, k, neg_Q)
                data_emb=key_encoder(data_tensor)
                #.detach()+l2 normalize data_emb
                enQ_deQ(queues[id_c][id_d],data_emb)
        all_pred=classifier(all_x)
        loss_ce=F.cross_entropy(all_pred, all_y)
        loss = loss_ce+lambda*loss_cl
        optim.zero_grad()
        loss.backward()
        optim.step()
        for thtild, tht in (key_encoder.params(),featurizer.params()):
            thtild=mu*thtild + (1 - mu)*tht
```

Here, $\tau > 0$ denotes the temperature parameter. $\mathcal{L}_{CL}(x^d_{y=c})$ in (3) is summed over all the examples from across all classes c and all domains d to obtain the following aggregated *contrastive loss*:

$$\mathcal{L}_{CL} = \sum_c \sum_d \sum_{x^d_{y=c}} \mathcal{L}_{CL}(x^d_{y=c}). \tag{4}$$

Now, assuming that we have a classifier $g_\phi(.)$ that predicts the class label of an example $x^d_{y=c}$, using the embedding obtained by the domain robust Query

Encoder $f_\theta(.)$, the *Empirical Risk Minimization* (ERM) can be approximated by minimizing the following loss:

$$\mathcal{L}_{CE} = \sum_d \sum_c l_{CE}(g_\phi(f_\theta(x^d_{y=c})), y = c). \tag{5}$$

Here, $l_{CE}(.)$ is the standard cross-entropy loss for classification. Then, the total loss for our method can be expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{CL}. \tag{6}$$

Here, $\lambda > 0$ is a hyperparameter. Thus, the overall optimization problem of our method can be expressed as:

$$\min_{\theta,\phi,\theta_g,\theta_a} \mathcal{L}_{total}. \tag{7}$$

Here, $\theta, \phi, \theta_g, \theta_a$ respectively denote the parameters of the (Query) Encoder $f_\theta(.)$, classifier $g_\phi(.)$, gated-fusion network $f_{\theta_g}(.)$ and the attention network $f_{\theta_a}(.)$. It should be noted that as the Key Encoder $f_{\tilde{\theta}}(.)$ is obtained as an EMA of the Query Encoder, we do not backpropagate gradients through it. In Algorithm 1, we provide a pseudo-code, roughly outlining the integration of our method into the PyTorch-based DomainBed framework. We call our method as the Refined Contrastive ERM (RCERM), owing to its refinement based feature augmentation, for positive generation in Contrastive learning.

3 Related Work and Experiments

Dataset: To evaluate the prowess of our proposed method, we make use of the **Photos, Art, Cartoons, and Sketches (PACS) dataset**. It consists of images from 4 different domains: i) Photos (1,670 images), ii) Art Paintings (2,048 images), iii) Cartoons (2,344 images) and iv) Sketches (3,929 images). There are seven semantic categories shared across the domains, namely, 'dog', 'elephant', 'giraffe', 'guitar', 'horse', 'house', 'person'. It is a widely popular dataset to test the robustness of Domain Generalization (DG) models. In particular, ***in this work, we are specifically interested in figuring out how our proposed method performs in the domains containing drawings and abstract imagery, such as, Art, Cartoons, and Sketches domains***. The images contained in these domains are significantly different from that of the Photos domain containing natural scene images.

Baseline/ Related Methods: Contrary to the traditional supervised learning assumption of the training and test data belonging to an identical distribution, Domain Generalization (DG) methods assume that training data is divided into a number of different, but semantically related domains, with an underlying causal mechanism, and test data could be from a different distribution. To generalize well in unseen data from a different distribution, and account for real-world situations, they try to learn invariance criteria among these domains.

Following are some of the related DG methods that have been used as baselines in this paper:

Table 1: Performance of SOTA methods on PACS using the training-domain validation model selection criterion

Method	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC	85.4 ± 0.8	79.7 ± 1.8	97.6 ± 0.3	78.2 ± 1.2	85.2
AND-mask	85.3 ± 1.4	79.2 ± 2.0	96.9 ± 0.4	76.2 ± 1.4	84.4
SAND-mask	85.8 ± 1.7	79.2 ± 0.8	96.3 ± 0.2	76.9 ± 2.0	84.6
Fishr	88.4 ± 0.2	78.7 ± 0.7	97.0 ± 0.1	77.8 ± 2.0	85.5

1. ERM [17]: Naive approach of minimizing consolidated domain errors.
2. IRM [1]: Attempts at learning correlations that are invariant across domains.
3. GroupDRO [14]: Minimizes the worst-case training loss over the domains.
4. Mixup [18]: Pairs of examples from random domains along with their labels are interpolated to perform ERM.
5. MLDG [8]: MAML based meta-learning for DG.
6. CORAL [16]: Aligning second-order statistics of a pair of distributions.
7. MMD [9]: MMD alignment of a pair of distributions.
8. DANN [4]: Adversarial approach to learn features to be domain agnostic.
9. CDANN [10]: Variant of DANN conditioned on class labels.
10. MTL [2]: Mean embedding of a domain is used to train a classifier.
11. SagNet [11]: Preserves the image content while randomizing the style.
12. ARM [19]: Meta-learning based adaptation of test time batches.
13. VREx [6]: Variance penalty based IRM approximation.
14. RSC [7]: Iteratively discarding challenging features to improve generalizability.
15. AND-mask [12]: Hessian matching based DG approach.
16. SAND-mask [15]: An enhanced Gradient Masking strategy is employed to perform DG.
17. Fishr [13]: Matching the variances of domain-level gradients.

Evaluation Protocol: The presence of multiple domains, the numerous available choices to construct a fair evaluation protocol, due to the multiple ways of forming the training data from across multiple domains makes model selection in DG a non-trivial problem. Inconsistencies in evaluation protocol, network architectures, etc, also makes a fair comparison among the plethora of available approaches difficult. To address this, the recently proposed framework DomainBed

Table 2: Performance of SOTA methods on PACS using the leave-one-domain-out cross-validation model selection criterion

Method	A	C	P	S	Avg
ERM	83.2 ± 1.3	76.8 ± 1.7	97.2 ± 0.3	74.8 ± 1.3	83.0
IRM	81.7 ± 2.4	77.0 ± 1.3	96.3 ± 0.2	71.1 ± 2.2	81.5
GroupDRO	84.4 ± 0.7	77.3 ± 0.8	96.8 ± 0.8	75.6 ± 1.4	83.5
Mixup	85.2 ± 1.9	77.0 ± 1.7	96.8 ± 0.8	73.9 ± 1.6	83.2
MLDG	81.4 ± 3.6	77.9 ± 2.3	96.2 ± 0.3	76.1 ± 2.1	82.9
CORAL	80.5 ± 2.8	74.5 ± 0.4	96.8 ± 0.3	78.6 ± 1.4	82.6
MMD	84.9 ± 1.7	75.1 ± 2.0	96.1 ± 0.9	76.5 ± 1.5	83.2
DANN	84.3 ± 2.8	72.4 ± 2.8	96.5 ± 0.8	70.8 ± 1.3	81.0
CDANN	78.3 ± 2.8	73.8 ± 1.6	96.4 ± 0.5	66.8 ± 5.5	78.8
MTL	85.6 ± 1.5	78.9 ± 0.6	97.1 ± 0.3	73.1 ± 2.7	83.7
SagNet	81.1 ± 1.9	75.4 ± 1.3	95.7 ± 0.9	77.2 ± 0.6	82.3
ARM	85.9 ± 0.3	73.3 ± 1.9	95.6 ± 0.4	72.1 ± 2.4	81.7
VREx	81.6 ± 4.0	74.1 ± 0.3	96.9 ± 0.4	72.8 ± 2.1	81.3
RSC	83.7 ± 1.7	82.9 ± 1.1	95.6 ± 0.7	68.1 ± 1.5	82.6

Table 3: Performance of SOTA methods on PACS using the test-domain validation model selection criterion

Method	A	C	P	S	Avg
ERM	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7
IRM	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5
GroupDRO	87.5 ± 0.5	82.9 ± 0.6	97.1 ± 0.3	81.1 ± 1.2	87.1
Mixup	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8
MLDG	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8
CORAL	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	82.7 ± 0.6	87.1
MMD	88.1 ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	87.2
DANN	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2
CDANN	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8
MTL	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7
SagNet	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4
ARM	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8
VREx	87.8 ± 1.2	81.8 ± 0.7	97.4 ± 0.2	82.1 ± 0.7	87.2
RSC	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2
AND-mask	86.4 ± 1.1	80.8 ± 0.9	97.1 ± 0.2	81.3 ± 1.1	86.4
SAND-mask	86.1 ± 0.6	80.3 ± 1.0	97.1 ± 0.3	80.0 ± 1.3	85.9
Fishr	87.9 ± 0.6	80.8 ± 0.5	97.9 ± 0.4	81.1 ± 0.8	86.9

[5] was introduced, that ensures an uniform evaluation protocol to inspect and compare DG approaches, by running a large number of hyperparameter and model combinations (called as a *sweep*) automatically.

It also introduces 3 model selection criteria:

1. **Training-domain validation**: The idea is to partition each training domain into a big split and a small split. Sizes of respective big and small splits would be the same for all training domains. The union of the big splits is used to train a model configuration, and the one performing the best in the union of the small

splits is chosen as the best model for evaluating on the test data.
2. **Leave-one-out validation**: As the name indicates, it requires to train a model on all train domains except one, and evaluate on the left out domain. This process is repeated by leaving out one domain at a time, and then choosing the final model with the best average performance.
3. **Test-domain validation**: A model is trained on the union of the big splits of the train domains (similar to Training-domain validation), and the final epoch performance is evaluated on a small split of the test data itself. The third criterion though not suited for real-world applications, is often chosen only for evaluating methods.

Performance of State-Of-The-Art (SOTA) approaches on PACS: Following the standard sweep of DomainBed, in Table 1, Table 2 and Table 3, we respectively report the performance of the SOTA DG methods on PACS dataset (in terms of classification accuracy, meaning that a higher value is better). The domains Art painting, Cartoons, Photos (natural images), and Sketches are shown as columns A, C, P and S respectively, along with the average performance of a method across these as column Avg. As also claimed by the DomainBed paper, we observed that ***no single method outperforms the classical ERM by more than one point, thus making ERM a reasonable baseline for DG***. Very recently, the Fishr method [13] has been proposed which performs competitive to ERM, on average. The most important observation is the fact that none of the methods performs the best across all the domains, showcasing the difficult nature of the dataset, and the problem of generalization in particular.

3.1 Comparison of our method RCERM against the SOTA ERM and Fishr methods:

Due to the SOTA performances of the ERM and the Fishr methods, we now compare our proposed method against them. In Table 4, we report the comparison of our method against the ERM method (best method for a column, within a selection criterion, is shown in bold). ***For the first two model selection criteria (train domain validation and leave-one-out) our proposed method outperforms the ERM method on all the domains, except on Cartoons***. While the performance gain on natural scene Photos domain is not large, we found that ***on Arts and Sketches, our method performs better than ERM by a large margin (upto 2% point). In the sketch domain, our method performs better by 3.7% point over ERM using the leave-one-out criterion***. Our method also performs better on Cartoons than ERM, when feedback from the test set is provided. However, in all other cases, ERM in itself performs quite competitive, in the first place.

We also compare our method with the most recently proposed Fishr approach in Table 5, and observe that ***our method outperforms Fishr on the Sketch domain by 3.6% points using train-domain validation criterion, and the Cartoons domain by 2.4% points using test-domain validation criterion*** (NOTE: Results of Fishr on leave-one-out have not been evaluated by

Table 4: Comparison of our proposed method against the state-of-the-art ERM method.

Model Selection train-domain validation					
Method	A	C	P	S	Avg
ERM	84.7 ± 0.4	**80.8** ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
RCERM	**86.6** ± 1.5	78.1 ± 0.4	**97.3** ± 0.1	**81.4** ± 0.6	**85.9**
Model Selection leave-one-domain out					
Method	A	C	P	S	Avg
ERM	83.2 ± 1.3	**76.8** ± 1.7	97.2 ± 0.3	74.8 ± 1.3	**83.0**
RCERM	**84.9** ± 1.7	70.6 ± 0.1	**97.6** ± 0.8	**78.5** ± 0.2	82.9
Model Selection test-domain validation set					
Method	A	C	P	S	Avg
ERM	**86.5**± 1.0	81.3 ± 0.6	96.2 ± 0.3	**82.7**± 1.1	**86.7**
RCERM	85.0 ± 0.7	**83.2** ± 1.4	**96.6** ± 0.2	80.8 ± 2.0	86.4

Table 5: Comparison of our proposed method against the state-of-the-art Fishr method.

Model Selection train-domain validation					
Method	A	C	P	S	Avg
Fishr	**88.4**± 0.2	**78.7**± 0.7	97± 0.1	77.8 ± 2.0	85.5
RCERM	86.6 ± 1.5	78.1 ± 0.4	**97.3** ± 0.1	**81.4** ± 0.6	**85.9**
Model Selection test-domain validation set (oracle)					
Method	A	C	P	S	Avg
Fishr	**87.9**± 0.6	80.8± 0.5	**97.9**± 0.4	**81.1**± 0.8	**86.9**
RCERM	85.0 ± 0.7	**83.2** ± 1.4	96.6 ± 0.2	80.8 ± 2.0	86.4

the authors). In fact, ***on average, we outperform Fishr using the train-validation criterion by 0.4% points***, which despite being a *low looking value*, is actually quite significant in the DG problem for the PACS dataset.

3.2 Ablation/ Analysis of our method:

While DomainBed frees the user of the hyperparameter searches automatically, we perform an ablation analysis of our method by removing the gated-fusion component of our method, and call it as RCERM with No Gated fusion (RCERMNG). The results are reported in Table 6. We found that ***RCERM by virtue of its gated fusion component indeed performs better than RCERMNG without the gated fusion***. This is because the positives update their representation by taking into account a query. This helps in contributing to the alignment of the embeddings of semantically similar objects from across domains.

3.3 Key takeaways:

1. The DG methods when compared in a fair, uniform setting using the DomainBed framework, perform more or less similar to the classical ERM

Table 6: Ablation studies showing role of the gated-fusion component

Model Selection train-domain validation					
Method	A	C	P	S	Avg
RCERMNG	86.5 ± 0.8	**80.4** ± 0.2	96.3 ± 0.4	77.1 ± 1.4	85.1
RCERM	**86.6** ± 1.5	78.1 ± 0.4	**97.3** ± 0.1	**81.4** ± 0.6	**85.9**
Model Selection leave-one-domain out					
Method	A	C	P	S	Avg
RCERMNG	82.9 ± 3.4	**75.4** ± 1.5	96.5 ± 1.3	76± 0.7	82.7
RCERM	**84.9** ± 1.7	70.6 ± 0.1	**97.6** ± 0.8	**78.5** ± 0.2	**82.9**

Fig. 6: Illustration of PACS type images.

Fig. 7: Illustration of a few Cartoon images with abnormal semantics.

method on the PACS dataset. The recently proposed Fishr method performs competitive to ERM.

2. On the PACS dataset, for all methods in general, the classification performance is the best for the Photos domain containing natural scene images ($\sim 97\%$ using training-domain validation), because, the underlying backbone (eg, ResNet50) has been pretrained on the ImageNet dataset which contains natural scenes. The models perform relatively well on the Art paintings domain ($\sim 84-88\%$ using training-domain validation), as the distribution of art images is relatively closer to that of natural images (in terms of texture, shades, etc). The performance is poorer on both Sketches and Cartoons ($\sim 73-80\%$ using training-domain validation), which have greater distribution shift compared to that of natural images, and where attributes like shape play a prominent role than texture, color etc. Figure 6 illustrates PACS type of images.
3. ***Despite the challenging nature of the Art and Sketches domains, our proposed method outperforms/ performs competitive against SOTA methods like ERM and Fishr***. This validates the merit of our work, to be a suitable alternative to address images with drawings and abstract imagery, while being robust across different domains. The promise of our method lies on the fact that **we are able to learn a single, common embedding which performs well on domains like art and sketches**.
4. Regarding the Cartoons domain, by inspecting certain images qualitatively, we suspect the very nature of the images to be the hurdle in obtaining a better performance. This is because of the fact that Cartoon images very often violate the semantics of objects as observed in real life, by virtue of abnormal attributes (Figure 7). For instance, a house having eyes, an animal

Table 7: Ranking of all the SOTA methods (including our RCERM) using two model selection criteria, on the Art (A) and Sketch (S) domains. Avg (A, S) denotes the average ranks across these two domains. A lower value indicates a better rank.

Training-domain validation				Leave-one-domain-out cross-validation			
Method	**A**	**S**	**Avg (A, S)**	**Method**	**A**	**S**	**Avg (A, S)**
SagNet	4	2	3				
CORAL	2	5	3.5				
RCERM	**6**	**1**	**3.5**	**RCERM**	**5**	**2**	**3.5**
ARM	5	3	4	MMD	4	4	4
Fishr	1	8	4.5	Mixup	3	8	5.5
MTL	3	10	6.5	MTL	2	9	5.5
VREx	10	9	9.5	GroupDRO	6	6	6
ERM	16	4	10	ARM	1	11	6
RSC	13	7	10	CORAL	14	1	7.5
MMD	8	13	10.5	ERM	9	7	8
SAND-mask	11	11	11	SagNet	13	3	8
GroupDRO	18	6	12	MLDG	12	5	8.5
MLDG	12	12	12	DANN	7	13	10
DANN	7	17	12	VREx	11	10	10.5
Mixup	9	16	12.5	IRM	10	12	11
AND-mask	14	14	14	RSC	8	14	11
IRM	15	15	15	CDANN	15	15	15
CDANN	17	18	17.5				

having extremely large/small eyes relative to the size of the entire head of the animal, or, even unusually large head. We believe, that a separate study could be dedicated to such cartoon images.

5. Based on the results from Table 1, Table 2 and Table 3, in Table 7 we additionally report the rankings of the performances of all the SOTA methods, including our RCERM, using two model selection criteria, on the Art (A) and Sketch (S) domains. It can be concluded that **across all DG methods, for Art and Sketches, RCERM obtains the second best average rank using train-domain validation, and best average rank using leave-one-out criterion**.

4 Conclusion

In this paper, we propose a novel Embedding Learning approach that seeks to generalize well across domains such as Drawings and Abstract images. During training, for a given query image, we obtain an augmented positive example for Contrastive Learning by leveraging gated fusion and attention. At the same time, to make the model discriminative, we push away examples from different semantic categories (across domains). We showcase the prowess of our method using the DomainBed framework, on the popular PACS (Photo, Art painting, Cartoon, and Sketch) dataset.

Acknowledgment

I would like to thank Professor Haris, MBZUAI, for the insightful conversations.

References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019) 9
2. Blanchard, G., Lee, G., Scott, C.: Generalizing from several related classification tasks to a new unlabeled sample. Advances in neural information processing systems **24** (2011) 9
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proc. of International Conference on Machine Learning (ICML). pp. 1597–1607. PMLR (2020) 7
4. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The journal of machine learning research **17**(1), 2096–2030 (2016) 9
5. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: Proc. of International Conference on Learning Representations (ICLR) (2020) 10
6. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). In: International Conference on Machine Learning. pp. 5815–5826. PMLR (2021) 9
7. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). In: International Conference on Machine Learning. pp. 5815–5826. PMLR (2021) 9
8. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 9
9. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5400–5409 (2018) 9
10. Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D.: Deep domain generalization via conditional invariant adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 624–639 (2018) 9
11. Nam, H., Lee, H., Park, J., Yoon, W., Yoo, D.: Reducing domain gap via style-agnostic networks. arXiv preprint arXiv:1910.11645 **2**(7), 8 (2019) 9
12. Parascandolo, G., Neitz, A., ORVIETO, A., Gresele, L., Schölkopf, B.: Learning explanations that are hard to vary. In: International Conference on Learning Representations (2020) 9
13. Rame, A., Dancette, C., Cord, M.: Fishr: Invariant gradient variances for out-of-distribution generalization. In: International Conference on Machine Learning. pp. 18347–18377. PMLR (2022) 9, 11
14. Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731 (2019) 9
15. Shahtalebi, S., Gagnon-Audet, J.C., Laleh, T., Faramarzi, M., Ahuja, K., Rish, I.: Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. arXiv preprint arXiv:2106.02266 (2021) 9
16. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European conference on computer vision. pp. 443–450. Springer (2016) 9
17. Vapnik, V.N.: An overview of statistical learning theory. IEEE transactions on neural networks **10**(5), 988–999 (1999) 9
18. Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., Zhang, W.: Adversarial domain adaptation with domain mixup. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6502–6509 (2020) 9

19. Zhang, M., Marklund, H., Gupta, A., Levine, S., Finn, C.: Adaptive risk minimization: A meta-learning approach for tackling group shift. arXiv preprint arXiv:2007.02931 **8** (2020) 9