DYNAMICALLY DECODING SOURCE DOMAIN KNOWLEDGE FOR UNSEEN DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization is an important problem which has gain much attention recently. While most existing studies focus on learning domain-invariant feature representations, some researchers try ensemble learning of multi experts and demonstrate promising performance. However, in existing multi-expert learning frameworks, the source domain knowledge has not yet been much explored, resulting in sub-optimal performance. In this paper, we propose to adapt Transformers for the purpose of dynamically decoding source domain knowledge for domain generalization. Specifically, we build one domain-specific local expert per source domain, and one domain-agnostic feature branch as query. Then, all local-domain features will be encoded by Transformer encoders, as source domain knowledge in memory. While in the Transformer decoders, the domain-agnostic query will interact with the memory in the cross-attention module, where similar domains with the input will contribute more in the attention output. This way, the source domain knowledge will be dynamically decoded for the inference of the current input from unseen domain. Therefore, this mechanism makes the proposed method well generalizable to unseen domains. The proposed method is evaluated on three benchmarks in the domain generalization field. The comparison with the state-of-the-art methods shows that the proposed method achieves the best performance, outperforming the others with a clear gap.

1 INTRODUCTION

Due to increasing applications of artificial intelligent techniques to real life, the domain shift problem raises a big challenge to learned models in generalizing to unseen domains. In order to deal with the domain shift problem, Domain Generalization (DG) has become a popular research topic in recent years (Li et al., 2017). A lot of methods have been proposed and great improvements have been achieved in domain generalization.

Among them, some approaches try to maintain both domain-specific parameters and domain-agnostic parameters, or create domain experts, and then combine them during the inference of a new image. For example, D-SAM is proposed in (D'Innocente & Caputo, 2018) to build a domain-specific aggregation module for each source domain and try to combine both generic and specific information for domain generalization. Mancini et al. (2018) also proposed to learn domain-specific networks for each source domain, and further learn a classifier fusion module in a single end-to-end trainable architecture. Recently, in the application of person re-identification of DG, (Dai et al., 2021) proposed to build a domain expert for each source domain and integrate their features into an adaptive voting process.

However, most of the existing methods with domain-specific parameters lack of knowledge transfer or interactions between source domains. The domain relationships are usually not explored properly when generating features. Considering this, we propose a hybrid deep architecture of domain-specific local experts and Transformer-based query-memory decoding for domain generalization, as shown in Fig. 1. With domain-specific local experts and a domain-agnostic query feature branch, a cross-domain Transformer is designed to dynamically decode source domain knowledge for the inference of a new image from unseen domain. Intuitively, to infer a new image, though it is from an unseen domain, this domain may still share some similarity to existing source domains. For example, the photo domain is more similar to the art-painting domain than others in the PACS dataset

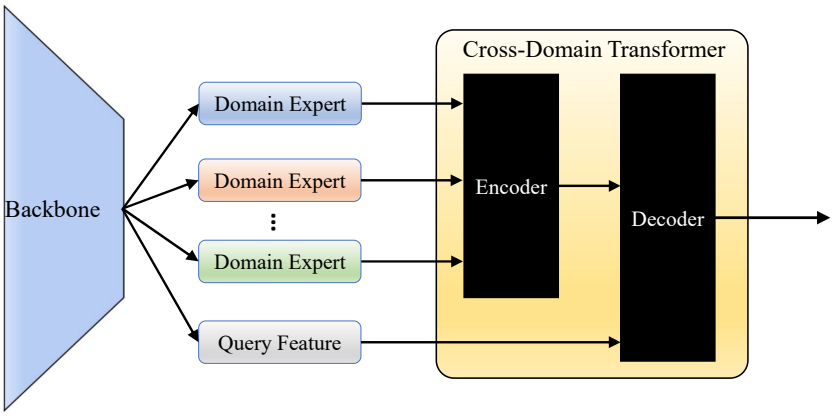

Figure 1: Concise illustration of the proposed approach.

(Li et al., 2017). Therefore, if the domain relationships are properly discovered, the closely related domain experts will be able to infer the new image well. Accordingly, in our design we apply Transformer to model domain relationships and decode useful information from the encoded source domain knowledge for the inference of unknown sample.

Specifically, the proposed method contains a shared Convolutional Neural Network (CNN) backbone, domain-specific local experts, and a global expert based on a cross-domain Transformer. A shared CNN backbone is utilized for feature extraction to ensure computational efficiency, and also enable learning of general features at low level. Then, we build a domain-specific expert for each source domain, as well as a domain-agnostic query feature branch. Then a cross-domain Transformer is designed for deep feature learning and domain relationships exploring. The source domain features are further encoded as memory, with self-attention to interact among different domains. Then the domain-agnostic feature branch is used as query, and a Transformer decoder is applied with both memory and query as inputs to explore the discriminant knowledge with source domains and dynamically decode the source domain knowledge for the inference of the input image with unseen domain. The final feature output by the cross-domain Transformer is used for classification.

The contribution of this work can be summarized as follows.

- An architecture is designed to encode source domain knowledge, by designing domain-specific local experts, and applying Transformer encoders, where the self-attention mechanism enables finding domain similarity and sharing generic and specific domain knowledge.
- A Transformer decoding scheme is designed, by interacting domain-agnostic query with the encoded memory in the cross-attention module, where similar domains with the input will contribute more in the attention output. This way, the source domain knowledge will be dynamically decoded for the inference of new images from unseen domains.
- This mechanism makes the proposed method well generalizable to unseen domains. Experimental results prove that the proposed method clearly outperforms the state-of-the-art methods.

2 Related Work

The domain generalization is a generic problem that makes a great challenge when applying learned models to work on unseen scenarios. Though it is a challenging problem, many efforts have been made to deal with the domain shift problem thanks to the fast development of deep neural networks. In the very recent years, many methods have been proposed and contributed great improvements to DG (Wang et al., 2021).

Existing methods can be divided into different categories based on their approaches. One kind of methods tries to improve the domain generalization ability by generating more and diverse data

samples for training, which can be categorized into data augmentation. It is proved that data augmentation strategies really help to increase the domain diversity of the source domains and hence improve the domain generalization ability. One of the famous methods is the JiGen method proposed by Carlucci et al. (2019), which used Jigsaw puzzle sample images along with the Jigsaw classifier to capture more informative features. Besides, Shankar et al. (2018) perturbed the input data with Bayesian Net and utilized adversarial strategy, Volpi et al. (2018) synthesized the "hard" data in training, Mancini et al. (2020) mixed up multiple source domains and categories to produce unseen categories in unseen domain, and Zhou et al. (2021) mixed styles of training instances to create new style samples.

Recently, the success of Learning to Learn or Meta Learning attracted many researchers' attention, especially for researchers in the DG field. Inspired by Meta Learning, some studies utilized the core idea of Meta Learning optimization strategy to improve the domain generalization ability. For example, Li et al. (2018) proposed the Meta-Learning for Domain Generalizaion (MLDG), Balaji et al. (2018) used a regularization function to improve domain generalization ability with Meta learning, named as MetaReg, Dou et al. (2019) introduced model-agnostic learning of semantic features (MASF), and so on (Santoro et al., 2016)(Finn et al., 2017). The main idea of methods in this category is to divide the given source domains to the meta-train and meta-test subsets and simulate the domain shift problem during training, which can be summarized to utilizing the optimization strategies.

Beyond the above approaches, another kind of methods can be concluded to the category of focusing on the feature levels across multiple source domains. For example, Muandet et al. (2013) first tried to learn an invariant transformation for DG by minimizing the differences in the marginal distributions across source domains and a kernel-based method was proposed. Then, Ghifary et al. (2015) introduced Multi-Task Auto-Encoder (MTAE) to learn unbiased object features with the data reconstruction. In order to minimize the distance between images from the same category but different domains, Motiian et al. (2017) imported the maximum mean discrepancy. Recently, inspired by the adversarial training strategy, Ganin et al. (2016), Li et al. (2018), Zhao et al. (2020) and Matsuura & Harada (2020) introduced this strategy in learning domain-invariant features. Some methods employ model based approaches by combining domain-agnostic and domain-specific parameters. These methods also belong to the feature level category. For example, Khosla et al. (2012) tried to represent parameters of each domain with domain-shared parameters and domain-specific parameters with shallow model. With a similar objective, Li et al. (2017) developed a low-rank parameterized deep model for end-to-end domain generalization learning.

Besides, Mancini et al. (2018) proposed to learn different domain-specific networks and classifiers for source domains and fuse the classification predictions with learnable weights for the target sample. Furthermore, the D-SAM method by D'Innocente & Caputo (2018) also proposed to set separate aggregation modules for each source domain with a shared backbone and combine probabilities of the outputs of domain-specific aggregation modules. Similar to D-SAM, Seo et al. (2020) proposed to maintain normalization parameters for each source domain and the final prediction is linearly combined. Zhou et al. (2020) also suggested the domain adaptive ensemble learning method with multiple domain-specific classifiers. However, these methods do not have a universal branch for generalization. The test samples from unknown domains just go pass the known domain specific networks for a fusion. Besides, they cannot model the relationships among source domains properly. Considering this, we would like to keep a domain-agnostic feature branch for the test image from unknown domain. Furthermore, in the proposed architecture the domain-agnostic feature acts as query and interacts with the local domain experts to discover useful knowledge across the source domains by a cross-domain Transformer, which helps a lot for generalizing to unseen domains.

3 THE PROPOSED METHOD

The overall architecture of the proposed method is shown in Figure 2. From the figure, it can be seen that multiple source domains share a CNN backbone at the beginning, and the feature output is then taken to different domain-specific local experts for specialist feature learning. Each domain expert is connected to a separate domain classifier, where the loss belonging to this domain only is computed. Beyond domain experts, it also keeps a domain-agnostic feature branch as query. Then, the learned domain expert features and the query feature are fed into a cross-domain Transformer for

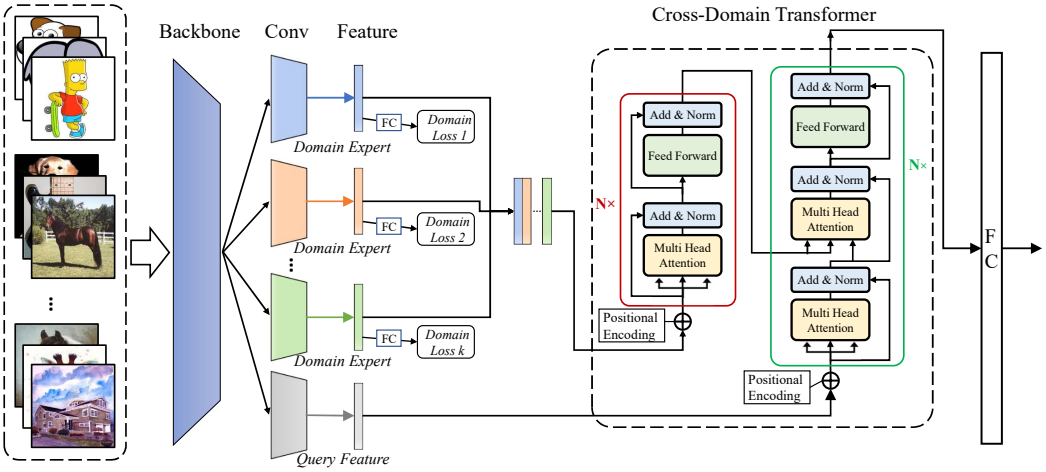

Figure 2: The architecture of the proposed method D^{2}SDK.

further learning. To better present the proposed method, we introduce Transformer in the following subsection.

3.1 Multi-Head Attention

Transformer is a global attention method proposed by Vaswani et al. (2017) to learn dependencies for sentences with an encoder and decoder architecture. The core module in Transformer is the Multi-Head Attention (MHA). Specifically, The scaled dot-product attention is the basic attention function used in MHA, and the formulation is

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{1}$$

where $Q \in \mathbb{R}^{T \times d_k}$ is the query feature matrix, $K \in \mathbb{R}^{M \times d_k}$ is the key feature matrix, and $V \in \mathbb{R}^{M \times d_v}$ is the value feature matrix. d_k is the feature dimension of the keys and queries, and d_v is the feature dimension of the values. Besides, M is the sequence length of the keys and values, and T is the sequence length of queries. Then, the multi-head attention concatenates h results of scaled dot-product attention for the given data, denoted as:

$$\text{Multi-Head}(Q, K, V) = \text{Concat}(H_1, H_2, ..., H_h)W^O, \tag{2}$$

where H_i stands for the attention result for the i-th head, and $W_O \in \mathbb{R}^{hd_v \times d}$ is a projection matrix that is finally multiplied to the concatenation of h heads attention results. For H_i, the definition is

$$H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{3}$$

where $W_i^Q \in \mathbb{R}^{d \times d_k}$, $W_i^K \in \mathbb{R}^{d \times d_k}$ and $W_i^V \in \mathbb{R}^{d \times d_v}$ are learned projection matrices for head H_i.

3.2 Cross-Domain Transformer

The Transformer is built with two parts. One is the Encoder part which is stacked with the Transformer encoder modular, and the other one is Decoder part which is also stacked with the Transformer decoder modular. The Transformer encoder contains two main blocks, which are MHA block and Multi Layer Perceptron (MLP) block, as shown in Fig. 2. It should be noted that the encoder uses multi-head self attention (MHSA) in the MHA block, which means $Q = K = V$. The MLP is a position-wise fully connected feed-forward network that contains two linear layers with a RELU non-linearity between them. In each block, both layer normalization and residual connection are utilized to stablize the learning of the deep architecture. As for Transformer decoder, it contains two MHA blocks followed by one MLP block, as also shown in Fig. 2. The first MHA block is also a self-attention block as in the encoder, while the following MHA block is a cross-attention block,

where keys and values are from the encoded memory different from the input queries. The layer normalization and residual connection are also utilized at the end of each block. Both Transformer Encoder and Decoder could be self-stacked for N times to build deep encoders and deep decoders for high-level feature learning.

The success of Transformer in natural language processing has attracted many researchers in the computer vision field. For example, Dosovitskiy et al. (2021) proposed Visual Transformer (ViT) by importing Transformer encoder to raw image patches for classification, and Srinivas et al. (2021) proposed Bottleneck Transformer with MHSA in deep CNN. Both of them only utilized self attention. In the work of Carion et al. (2020), they proposed Detection Transformer (DETR) for object detection using both Transformer encoder and decoder where the query input in decoder is learnable parameters. In contrast, we use both self attention and cross attention in the proposed method, which takes general CNN features as the query input of Transformer decoder to find its close and discriminant features from the learned memory according to the existing source domains.

Specifically, the domain-specific features from domain experts are concatenated and fed into the Transformer encoders so that the Transformer is able to learn further dependency relationships and build the knowledge of existing source domains. The domain-agnostic feature branch is directly imported to the Transformer decoders as the query input. It will go through the self-attention block firstly and then is fed into the cross-attention block as query features. At the same time, the outputs of Transformer encoder are taken as memory, containing keys and values, to the cross-attention block in the Transformer decoder. The relationships among different domains will be explored in the Transformer decoder, and the newly learned feature is an ensemble feature across the seen domains according to the dynamically learned domain relationships.

3.3 THE PROPOSED D^{2}SDK

As shown in Fig. 2, the backbone is shared to all domain experts and the query feature branch. This enables common CNN feature learning across different domains at low layers and avoids excessive computational and memory consuming in building domain-specific backbone network. In order to learn discriminant features for each domain, we further append a small CNN sub-network at the beginning of each domain expert, as well as the query feature branch. Then, the learned domain expert features are added with positional embeddings before they are fed into the Transformer encoders for source domain knowledge learning. The query feature branch is also added with positional embeddings, and then is taken as query input to the Transformer decoders. The query feature interacts with the domain expert features in the multi-head cross-attention block of the Transformer decoders, where the source domain knowledge in the memory is decoded dynamically with respect to the query input. The aggregated feature is finally go through a fully connected layer for classification.

Beyond the final classifier, we also add separate domain classifiers for each domain expert. This includes an independent FC layer and a classification loss function for each domain. Therefore, the final loss function is built with both domain expert loss and final classification loss as follows.

$$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{\theta}) = (1-\lambda)\mathcal{L}^G(\boldsymbol{x}, y; \boldsymbol{\theta}^G \cup \boldsymbol{\theta}^B) + \lambda \sum_{i=1}^{k} d_i \mathcal{L}_i^D(\boldsymbol{x}, y; \boldsymbol{\theta}_i^D \cup \boldsymbol{\theta}^B), \tag{4}$$

where $\boldsymbol{\theta}$ stands for the whole network parameters, and $\boldsymbol{\theta}^G$ stands for the query branch, Transformer, and the final classifier parameters, $\boldsymbol{\theta}_i^D$ stands for the i-th domain expert network parameters, and $\boldsymbol{\theta}^B$ is the shared CNN backbone parameters. As for the loss functions, $\mathcal{L}^G$ is the final classification loss function, and $\mathcal{L}_i^D$ is the local domain expert loss function for domain i. They are both cross-entropy loss for general classification. Besides, $d_i = 1$ if the sample $\boldsymbol{x}$ belongs to the domain i, otherwise $d_i = 0$. k is the number of source domains, and λ is the weight of the local domain experts.

4 EXPERIMENTS

To evaluate the performance of the proposed method, we test it on three popularly used datasets, compared to existing state-of-the-art methods.

Table 1: Experimental results on PACS with ResNet-18 as backbone.

Method	Photo	Art-paiting	Cartoon	Sketch	Average
DSAM (D'Innocente & Caputo, 2018)	95.30	77.33	75.89	69.27	80.72
MLDG (Li et al., 2018)	94.00	78.70	73.30	65.10	80.70
Metareg (Balaji et al., 2018)	95.50	83.70	77.20	70.30	81.70
JiGen (Carlucci et al., 2019)	96.03	79.42	75.25	71.35	80.51
MASF (Dou et al., 2019)	94.99	80.29	71.17	71.69	81.03
AGG (Li et al., 2019)	94.4	77.6	73.9	70.3	79.1
Epifcr(Li et al., 2019)	93.9	82.1	77.0	73.0	81.5
Cumix (Mancini et al., 2020)	95.10	82.30	76.50	72.60	81.60
MMLD (Matsuura & Harada, 2020)	96.09	81.28	77.16	72.29	81.83
ER (Zhao et al., 2020)	96.65	80.70	76.40	71.77	81.38
DADG (Chen et al., 2020)	94.86	79.89	76.25	70.51	80.38
MixStyle (Zhou et al., 2021)	96.1	84.1	78.8	75.9	83.7
D^2SDK	95.85	83.60	81.24	76.47	84.29

4.1 Implementation Details

In the experiments, the ResNet proposed by He et al. (2016) is utilized, until layer3, as the shared CNN backbone. The instance normalization is used in the CNN backbone. Then, the layer4 of ResNet is deeply copied several times to be domain-specific expert and domain-agnostic query feature branch. Afterwards, we add 1-D learnable positional embeddings to the CNN features before they are fed into the Transformers. In the cross-domain Transformer, two layers of the Transformer encoders and two layers of the Transformer decoders are stacked. For the combined loss function we set $\lambda = 0.1$ by default for the weight of domain expert loss on all the evaluated datasets. In the optimization, an SGD solver with a learning rate of 0.001 and a batch size of 32 is used. The training takes 80 epochs. The learning rate is decayed by a factor of 0.1 after reaching 80% of the training epochs, according to Carlucci et al. (2019). During training, we follow the data augmentation strategy as in Carlucci et al. (2019) on all evaluated datasets. Besides, all of the results reported are averaged among ten rounds of run to avoid random bias.

4.2 Datasets

In the experiment, the proposed architecture is evaluated on PACS, Office-Home and VLCS datasets, which are the most popular datasets in domain generalization. The **PACS** dataset is proposed by Li et al. (2017) for DG. It contains 9,991 images coming from seven categories. It is built of four domains, namely Photo (P), Art Painting (A), Cartoon (C) and Sketches (S). Till now, it is the most popular DG dataset, which can be downloaded freely for research purpose[1]. The experimental protocol on this benchmark (proposed in Li et al. (2017)) is followed to ensure fair comparison.

Besides of PACS, Office-Home (Venkateswara et al., 2017) is another widely used DG dataset, which contains 65 categories coming from daily used objects. It also has four domains, which are Art, Clipart, Product and Real World. More information can be found in the website[2] and the data is also free for research usage. In the experiment, the protocol of leaving one domain out for test as introduced in PACS is followed.

At last, the proposed method is also evaluated on **VLCS**, which is also a classic domain generalization benchmark (Torralba & Efros, 2011). It contains 10,729 images. The dataset contains only 5 categories that are shared by PASCAL VOC 2007 (V), Labelme (L), CALTECH (C) and SUN (S) databases. We take the experimental protocol which is proposed by Ghifary et al. (2015) for a fair comparison.

[1] https://domaingeneralization.github.io/#data

[2] https://www.hemanthdv.org/officeHomeDataset.html

Table 2: Experimental results on Office-Home with ResNet-18 as backbone.

Methods	Art	Clipart	Product	RealWorld	Average
DSAM (D'Innocente & Caputo, 2018)	58.03	44.37	69.22	71.45	60.77
MLDG (Li et al., 2018)	52.88	45.72	69.90	72.68	60.30
JiGen (Carlucci et al., 2019)	53.04	47.51	71.47	72.79	61.20
DADG (Chen et al., 2020)	55.57	48.71	70.90	73.70	62.22
D^2SDK	60.34	52.23	75.05	77.64	66.32

Table 3: Experimental results on VLCS.

Method	Caltech	Labelme	Pascal	Sun	Average
MMD-AEE (Li et al., 2018)	94.40	62.60	67.70	64.40	72.30
D-SAM (D'Innocente & Caputo, 2018)	91.75	56.95	58.59	60.84	67.03
MLDG (Li et al., 2018)	94.4	61.3	67.7	65.9	72.3
JiGen (Carlucci et al., 2019)	96.93	60.90	70.62	64.30	73.19
AGG (Li et al., 2019)	93.1	60.6	65.4	65.8	71.2
Epi-FCR (Li et al., 2019)	94.1	64.3	67.1	65.9	72.9
MASF (Dou et al., 2019)	94.78	64.90	69.14	67.64	74.11
DADG (Chen et al., 2020)	96.80	66.81	70.77	63.64	74.46
D^2SDK	97.41	62.63	75.48	69.04	76.14

4.3 State-Of-The-Art Methods

To evaluate the proposed method, the following state-of-the-art methods are compared in the experiments. MMD-AAE Li et al. (2018) imports the adversarial auto-encoders to learn an invariant feature representation by aligning the data distributions with MMD. D-SAM D'Innocente & Caputo (2018) wants to merge the generic and specific information with the domain-specific aggregation modules to improve the model generalization ability. Recently, based on meta-learning, MLDG Li et al. (2018) is proposed with a model-agnostic training procedure that trains any given model to be more robust to domain shift. After MLDG, the owner studied the episodic training strategy and proposed Epi-fcr and AGG methods Li et al. (2019). Similar to MLDG, MetaReg Balaji et al. (2018) is proposed, which uses a regularizer to gain a general representation across domains with episodic training procedure. Furthermore, MASF Dou et al. (2019) uses model-agnostic episodic learning procedure with a triplet loss, and DADG Chen et al. (2020) utilizes both discriminant adversarial learning and the train/test domain-shift meta-learning techniques to gain generalized features. Beyond episodic strategy, Carlucci et al. (2019) suggested to improve the deep learning generalization ability by solving Jiggle puzzles, denoted as JiGen. The proposed method is also compared to the Cumix in Mancini et al. (2020), which wants to deal with the zero-shot learning problem and domain generalization problem with the mixed up samples generated from multiple source domains and categories during training. Besides, we also include MMLD in Matsuura & Harada (2020) and ER in Zhao et al. (2020), which both belong to the domain-invariant feature learning approach.

4.4 Results

PACS The compared results on the PACS data set are shown in Table 1. From the table, it is clear that the proposed method achieves the best performance. It is impressive that the proposed method outperforms the compared methods with a clear gap on target domains of Cartoon and Sketch. Especially, on the cartoon test domain, the proposed method outperforms the second best one by more than 2%.

Office-Home The experimental results are shown in Table 2 with the comparison to state-of-the-art methods. From the table, it can be seen that the proposed method also achieves the best performance, which outperforms the second best method DADG by more than 4% on average. On every sub task with different target domains, the proposed method also achieves the best performance with a clear improvement over the best existing results.

Table 4: Experimental results with comparison to baselines. D^2SDK_{18} is with ResNet-18 while D^2SDK_{50} is with ResNet-50.

	Methods	Photo	Art-paiting	Cartoon	Sketch	Average
PACS	ResNet-18	95.19	80.41	75.94	73.28	81.20
	D^2SDK_{18}	95.85	83.60	81.24	76.47	84.29
	ResNet-50	97.09	87.29	81.00	74.11	84.87
	D^2SDK_{50}	97.47	88.67	84.96	78.78	87.47
	Methods	Art	Clipart	Product	RealWorld	Average
Office-Home	ResNet-18	54.06	47.56	72.17	74.22	62.00
	D^2SDK_{18}	60.34	52.23	75.05	77.64	66.32
	ResNet-50	62.09	53.29	76.67	78.67	67.68
	D^2SDK_{50}	68.92	57.62	80.25	82.27	72.26
	Methods	Caltech	Labelme	Pascal	Sun	Average
VLCS	ResNet-18	96.55	62.50	72.19	66.52	74.44
	D^2SDK_{18}	97.41	62.63	75.48	69.04	76.14
	ResNet-50	98.34	63.13	74.50	69.93	76.47
	D^2SDK_{50}	97.47	63.03	78.12	70.53	77.29

VLCS The evaluated results on this dataset are shown in Table 3 along with the compared methods. It should be noted that we used ResNet-18 as the backbone for the proposed method for convenience, while the compared methods used the AlexNet. Though the comparison is not very fair, we would like to share a new set of results with a more widely used backbone on the classic dataset. Considering this, it is no doubt that the proposed method achieves the best performance.

4.5 DISCUSSIONS

The experimental results demonstrate that the proposed method has a good generalization ability for domain generalization to unseen domains, thanks to the mechanism in cross-domain Transformer where the source domain knowledge is encoded and dynamically decoded for the inference of new images from unseen domains.

Note that, all the results reported for the proposed method are based on the model learned at the last epoch in training. They are close to the results with the best model validated on the validation set. This means that we do not use early stopping or select the best results among the epochs with the test set. Actually, we found that the performance with converged and stable model on training set and validation set are not as good as the best-epoch performance on the test set monitored during training. On the PACS and VLCS datasets, the gaps are larger than on the Office-Home dataset. More results with the best epochs can be found in the Appendix.

More experimental results can be found in Table 4, where the proposed D^2SDK is compared to the baselines with ResNet-18 and ResNet-50 on the PACS, Office-Home and VLCS datasets. From the comparison we can see that, the proposed method has a clear improvement to the baselines on all evaluated datasets. Especially, on the Office-Home dataset, D^2SDK outperforms the baselines by more than 4% on average.

Besides, we also evaluated the influence of parameter λ for the weight of domain expert loss. This evaluation is done on the PACS dataset with ResNet-18 as the backbone. The results are displayed in Table 5. From the table, it can be seen that with descending values of parameter λ, the proposed D^2SDK has an increasing accuracy with the target domain of Photo. However, when λ is smaller than 0.2, the performance is very close to each other. The same finding also holds with the Art-painting as the target domain. As for the Cartoon, it seems like the λ has very little influence on the performance. However, with the Sketch domain as target, the performance has a slow dropping down corresponding to the descending of λ. As a result, the overall average results are close to each other, except that when λ is too big, indicating that this parameter is not sensitive. Therefore, considering the random influence, we simply set λ to 0.1 for all tasks in the experiments.

Table 5: Experimental results with different λ of the proposed method.

λ	Photo	Art-paiting	Cartoon	Sketch	Average
0.7	93.35	81.30	81.40	79.98	84.00
0.5	94.48	82.80	81.65	78.51	84.36
0.3	95.32	83.71	81.43	77.34	84.45
0.2	95.65	84.12	80.86	78.36	84.74
0.1	95.85	83.60	81.24	76.47	84.29
0.05	95.70	83.64	80.87	76.91	84.28
0.02	95.64	83.75	81.43	77.35	84.54
0.01	95.71	84.05	81.21	76.40	84.34

5 CONCLUSION

In this paper, we show that, given domain-specific local experts and query features of the current input, Transformers are effective in discovering domain relationships and in turn help generalizing the inference of images from unseen domains. This is possible thanks to the self-attention mechanism in Transformer encoders and cross-attention mechanism in Transformer decoders. In our design, unknown samples are able to exploit the encoded source domain knowledge for the inference of their labels. With this mechanism, the proposed method is shown to be very promising in addressing the domain generalization challenge. In the future, it is interesting to explore whether the proposed method would be even powerful in addressing much more number of domains, where domain relationships could be more diverse and fine-grained.

REFERENCES

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, pp. 213–229, 2020.

Fabio M. Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Keyu Chen, Di Zhuang, and J. Chang. Discriminative adversarial domain generalization with meta-learning based cross-domain validation. *ArXiv*, 2020.

Yongxing Dai, Xiaotong Li, Jun Liu, Zekun Tong, and Ling yu Duan. Generalizable person re-identification with relevance-aware mixture of experts. *CVPR*, 2021.

Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, pp. 187–198, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Qi Dou, Daniel C. Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.

Muhammad Ghifary, W. Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *International Conference on Computer Vision (ICCV 2015),*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, 2012.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018.

Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018.

Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: Domain generalization through source-specific nets. In *ICIP*, pp. 1353–1357, 2018.

Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *The Proceedings of European Conference on Computer Vision*, pp. 466–483. Springer, 2020.

Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:11749–11756, 2020.

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. *The 30th International Conference on Machine Learning (ICML 2013)*, 28:10–18, 2013.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, volume 48, pp. 1842–1850, 2016.

Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *Proceedings of the European Conference on Computer Vision 2020*, pp. 68–83, 2020.

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

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition, 2021.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1528, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017.

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

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *CoRR*, abs/2103.03097, 2021. URL `https://arxiv.org/abs/2103.03097`.

Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16096–16107, 2020.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *arXiv preprint arXiv:2003.07325*, 2020.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.

A Appendix

In DG experimental setting, all the source domain images are divided into a training set and a validation set, and the learned model are tested on the unseen target domain as test set. No information from the target domain is provided in the training. In practice, we found that the best model selected on the validation set and the finally learned model at the last epoch have a similar performance on the target domain. However, usually these results are not as good as the best performance among all epochs monitored on the test set. We call this test-set best-epoch performance. As far as we know, some existing methods report this kind of best-epoch results monitored on the test set, which is not quite fair.

In the main text of this work, all the results reported for the proposed method are based on the learned model at the last epoch. In the appendix, we would like to additionally provide the test-set best-epoch performance as well, in case there is a need for such kind of comparison. However, we strongly discourage doing so.

The results are displayed in Table 6. From the comparison of Table 6 to Table 4 in the main paper, it can be seen that the test-set best-epoch results on PACS and VLCS are clearly better than our last-epoch results, with about 2%-3% performance gap. As for the Office-Home dataset, the performance of the finally learned model is close to the best-epoch results in Table 6. Furthermore, considering that PACS contains only seven categories and VLCS contains only five categories, we think PACS and VLCS have more chances to gain a better result on local minimums. In contrast, the Office-Home has 65 categories, which contains more objects compared to PACS and VLCS. Thus, Office-Home is a more challenging dataset, and thus it may be more reliable for the evaluation of DG methods.

Table 6: The test-set best-epoch performance on three datasets.

	Methods	Photo	Art-paiting	Cartoon	Sketch	Average
PACS	D^2SDK_{18}	96.69	85.99	84.30	79.90	86.72
	D^2SDK_{50}	98.07	90.64	86.75	81.76	89.30
	Methods	Art	Clipart	Product	RealWorld	Average
Office-Home	D^2SDK_{18}	61.01	53.20	75.44	77.81	66.86
	D^2SDK_{50}	69.53	59.11	80.55	82.56	72.94
	Methods	Caltech	Labelme	Pascal	Sun	Average
VLCS	D^2SDK_{18}	98.94	67.52	78.77	72.09	79.33
	D^2SDK_{50}	99.43	67.65	81.36	74.93	80.84