Disentangled Prompt Representation for Domain Generalization

De Cheng[1*], Zhipeng Xu[1*], Xinyang Jiang[2†], Nannan Wang[1†], Dongsheng Li[2], Xinbo Gao[3]
[1]Xidian University, [2]Microsoft Research Asia, [3]Chongqing University of Posts and Telecommunications
{nnwang,dcheng}@xidian.edu.cn, xu_zhipeng@stu.xidian.edu.cn
{xinyangjiang, dongsli}@microsoft.com, gaoxb@cqupt.edu.cn

Abstract

Domain Generalization (DG) aims to develop a versatile model capable of performing well on unseen target domains. Recent advancements in pre-trained Visual Foundation Models (VFMs), such as CLIP, show significant potential in enhancing the generalization abilities of deep models. Although there is a growing focus on VFM-based domain prompt tuning for DG, effectively learning prompts that disentangle invariant features across all domains remains a major challenge. In this paper, we propose addressing this challenge by leveraging the controllable and flexible language prompt of the VFM. Observing that the text modality of VFMs is inherently easier to disentangle, we introduce a novel text feature guided visual prompt tuning framework. This framework first automatically disentangles the text prompt using a large language model (LLM) and then learns domain-invariant visual representation guided by the disentangled text feature. Moreover, we also devise domain-specific prototype learning to fully exploit domain-specific information to combine with the invariant feature prediction. Extensive experiments on mainstream DG datasets, namely PACS, VLCS, OfficeHome, DomainNet and TerraInc, demonstrate that the proposed method achieves superior performances to state-of-the-art DG methods.

1. Introduction

Most of the machine learning methods are built on the assumption that training and testing data are independently and identically distributed (*i.i.d.*) [7, 7, 26, 42]. However, in real-world scenarios, this assumption does not always hold where *distribution shift* between training and testing data occurs frequently. As a result, Domain Generalization (DG) task has been proposed, aiming to learn a generalized model to perform well on unseen target domain only using limited source domain data for training. The DG has long been an essential topic in the machine learning field and attracts considerable attention.

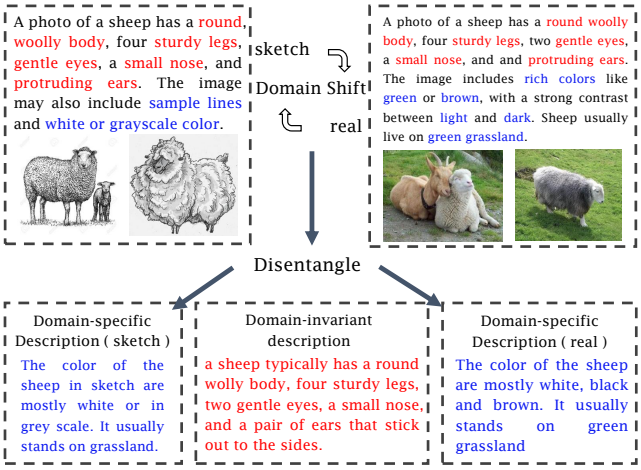

Figure 1. The upper portion of the image illustrates the differences among various domains, which are subsequently disentangled into domain-invariant and domain-specific descriptions via our text disentanglement process, as depicted in the lower portion of the image.

The main challenge for DG is to learn a model with good generalization ability, to extract domain-invariant representations across source domain datasets. Fortunately, latest researches show clear evidence that large-scale pre-trained model could greatly enhance the domain generalization power. Specially for the pre-trained VFMs (*e.g.*, CLIP), they are trained by utilizing large-scale (image, text) pairs for robust visual representation, which is inherently rich in semantic information of prior knowledge. Hence, such VFMs are able to encode the semantic meanings of visual descriptions, regardless of the image styles, which is in line with the goal of DG, *i.e.,* learning uniform visual semantic representations across different domains.

Most current DG methods use multi-domain training datasets to train a model generalizes to unseen domains. It is intuitive that we can also enhance the ability of foundation models by fine-tuning them using these datasets. Recent developments have introduced the idea of fine-tuning

*Equal contribution.
†Corresponding authors.

foundation models with prompts, that achieve better results on specific downstream tasks with very few training samples. However, directly applying prompt tuning in the context of domain generalization poses significant challenges. This is due to the fact that existing prompt tuning methods tune the foundation model to generate domain and task-specific features, whereas domain generalization requires the model to generate domain-invariant features that work well across different unseen domains. Therefore, in order to apply prompt tuning to domain generalization effectively, it is crucial to develop prompts that can guide the foundation model in disentangling invariant features across all domains from those specific to certain domains.

In this paper, our solution to this challenge is to fully leverage a distinctive aspect of VFM, which is the controllable and flexible language prompt. We believe the text prompt plays a vital role to guide the disentanglement of image feature. Intuitively, the text modality in VFM can be more easily disentangled, as the text/language is inherently rich in semantic information and well interpretable. For example, as shown in Fig. 1, given a comprehensive textual description of an image class 'sheep', one can easily separate the description into two parts. One is the text describing specific domains (e.g., overall color of the sheep and image), and the other is the description invariant to the domain (e.g., the shape and texture of a sheep). The generation of these class descriptions and their disentanglement can be easily achieved by leveraging large language models (LLMs) such as GPT.

As a result, we propose a novel prompt tuning framework for domain generalization with LLM-assist text prompt disentanglement and text-guided visual representation disentanglement model. Specifically, domain-invariant and domain-specific descriptions of each category are first generated with LLM, which is used for prompt tuning to learn disentangled textual features. Secondly, the learned disentangled textual features are utilized to guide the learning of domain-invariant and domain-specific visual features. Nevertheless, effectively leveraging disentangled visual feature to achieve good performance remains a challenge. Unlike many conventional methods that concentrate solely on domain-invariant features, we contend that, in order to classify images from an unseen domain, leveraging domain-specific knowledge from similar seen domains is also essential. Specifically, we propose domain-specific prototype learning (DSPL), where a prototype is learned for each class from each domain and suitable domain-specific prototypes will be selected for images from different unseen domains. The final output of the model is the combination of the prediction of the domain-invariant feature and DSPL.

Our main contributions can be summarized as follows:

- We propose a novel prompt tuning framework for domain generalization with LLM-assist text prompt disentanglement followed by text-guided visual representation disentanglement.
- We also devise the domain-specific prototype learning, to fully exploit domain-specific information to be combined with the domain-invariant prediction for final inference.
- Extensive experiments on mainstream DG datasets, namely PACS, VLCS, OfficeHome, DomainNet and TerraInc, show that the proposed DPR method achieves superior performances to state-of-the-art DG methods.

2. Related Work

2.1. Domain Generalization.

DG involves training a model using one or multiple source domains to achieve robust performance on unseen target domains. Early-stage theoretical methods have addressed domain shifts from multiple angles, including domain alignment, meta-learning, data augmentation, disentangled representation learning, and capturing causal relations. Domain alignment refers to learning domain-invariant representations by removing domain-specific knowledge [3, 14, 30, 38, 59]. Meta-learning improves generalization performance by simulating domain shift through a training procedure that divides the source domain into meta-train and meta-test domains [4, 13, 29, 55]. [6, 47, 60] find data augmentation can play a significant role when augmentations can approximate some variations between domains. [21, 42, 51] using disentangled representation learning to disentangle features into a domain-invariant content space and a domain-specific attribute space, thus learning a domain-invariant representation from data across multiple domains. There also contain other works[1, 24] explore to capture causal relations to solve this problem. Meanwhile, some work [8–12, 20, 27, 33, 35, 43, 46, 52, 58] has also made contributions.

2.2. DG with Pre-trained Models.

Recent studies show the pre-trained model can bring out-of-distribution generalization capabilities [39]. [16] demonstrates that simple ERM [49] outperforms the majority of early methods utilizing pre-trained ResNet-50 [17]. [7] propose a regularization method called MIRO, which aims to improve model generalization by minimizing the mutual information between the pre-trained and oracle models. [40] utilizes CLIP [44], a large-scale vision-language pre-trained model, to extract domain-unified representations by generating diverse prompts. [26] propose a new gradient-based method that learns task-specific knowledge while preserving the generalization ability of large-scale pre-trained models. [32] propose a specialized model-sample matching method for DG. Similarly, our work also employs pre-trained CLIP, and utilizes LLM to generate domain-invariant and specific descriptions to guide the training.

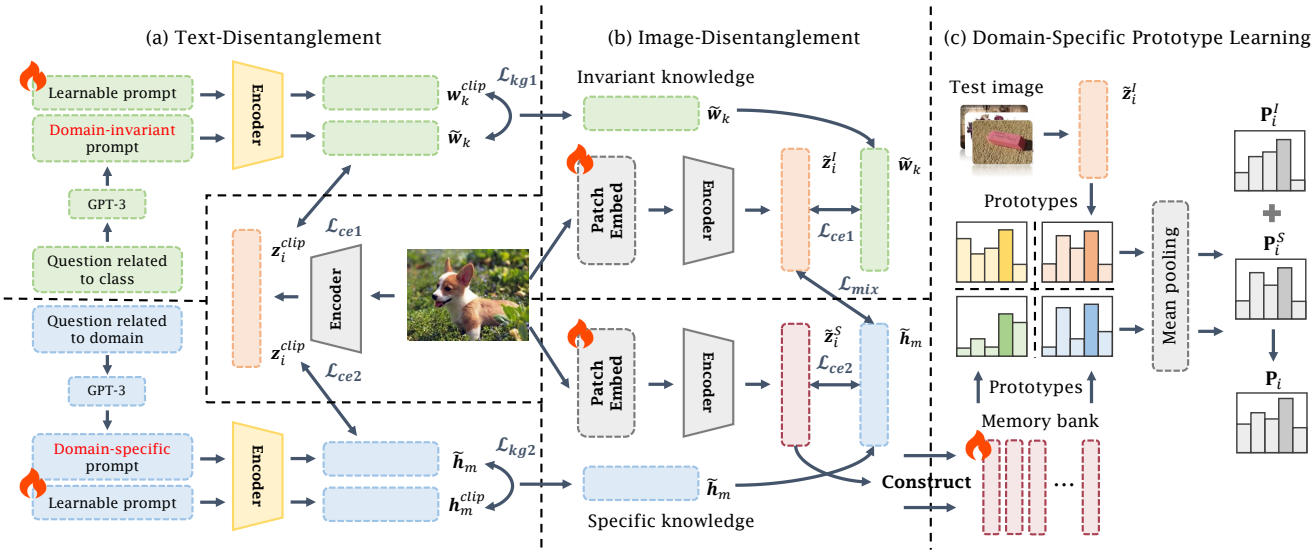

Figure 2. Framework of our proposed method. The process of text disentanglement (a) (Described in Sec. 3.2) involves training both domain-invariant textual embedding $\tilde{\mathbf{w}}_k$ and domain-specific textual embedding $\tilde{\mathbf{h}}_m$, which serve as the guidance of the disentangled visual representations, denoted as $\tilde{\mathbf{z}}_i^I$ and $\tilde{\mathbf{z}}_i^S$ in (b) (Described in Sec. 3.3). Subsequently, in step (c) (Described in Sec. 3.4), we employ prototype learning to effectively capture domain-specific information. The resulting new prediction, denoted as $\mathbf{P}_i^S$, is then combined with the domain-invariant prediction $\mathbf{P}_i^I$ to obtain the final inference $\mathbf{P}_i$.

2.3. Disentangled Representation Learning.

There are several prior studies of disentangled representation learning related to our work. [34, 36, 37] utilizes generative adversarial networks (GANs) [15, 23] and variational autoencoders (VAEs) [22, 45] to learn an interpretable representation. Another work [42] introduces a deep adversarial disentangled autoencoder (DADA) and a novel domain agnostic learning (DAL) schema to achieve representation disentanglement. Different from existing work, our proposed method guides the entire training process by generating domain-invariant and domain-specific descriptions through text disentanglement. Moreover, during the inference stage, we leverage prototypes to fully exploit domain-specific information.

3. Methodology

Problem Definition. Let $\mathcal{D}^S$ and $\mathcal{D}^T$ bet sets of the source domain and target domain. Specifically, $\mathcal{D}^S = \{\mathcal{D}_m^S\}_{m=1}^{N_s}$, where $\mathcal{D}_m^S$ is a distribution over the input space $\mathcal{X}$, and N_s is the total number of source domains. For each source domain, $\mathcal{D}_m^S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_m^s}$, where each data sample $(\mathbf{x}_i, y_i)$ consists of the input image $\mathbf{x}_i$ and its corresponding label y_i. The unseen target domain can also be defined as $\mathcal{D}^T = \{\mathcal{D}_m^T\}_{m=1}^{N_t}$, where the number of target domain is usually set to one in the experiment. The goal of DG is to build a model $\phi_\theta(\cdot)$ that can perform well on the target domain, when only the source domain data are available during model training. The main challenge is to address the domain distribution shift between the source and target data. Note that, this study is verified on solving the image classification problem.

Overall Framework of the DPR Method. As shown in Fig. 2, the overall framework of the proposed DPR method for DG consists of the following modules: 1) The GPT-Assist text-disentanglement module; 2) The image disentanglement module; 3) The relevance-inspired domain-specific prototype learning. The process of text disentanglement in Fig. 2 (a) entails training both domain-invariant and domain-specific text embeddings, which subsequently serve as guidance for image disentanglement in Fig. 2 (b). Finally, in Fig. 2 (c), domain-specific information is effectively leveraged to combine with domain-invariant prediction for the final inference. In this way, the proposed DPR method can capture domain-specific and domain-invariant information for DG through disentangled prompt tuning.

3.1. Preliminaries

Prompt Tuning on CLIP Model. We adopt CLIP as the pre-trained vision-language foundation model for prompt tuning. Since CLIP is trained with 400M $(image, text)$ association pairs, it contains two types of encoders: 1) the visual encoder $\mathbf{f}(\mathbf{x})$ to map the input image $\mathbf{x}$ into the visual embedding; 2) the text encoder $\mathbf{g}(\cdot)$ to map the text description $\mathbf{t}$ onto the textual embedding. For CLIP-based prompt tuning, it utilizes the hand-craft prompt or the learnable prompt to adapt the pre-trained CLIP model to downstream tasks, with frozen visual and text encoders.

For the downstream task with N_c categories, CLIP employs a hand-craft/learnable prompt, denoted as $\mathbf{t}_k$ for the k_{th} class, to generate its corresponding textual embedding $\mathbf{w}_k^{clip} = \mathbf{g}(\mathbf{t}_k)$, and the textual embeddings for all cate-

gories can be denoted as $\mathbf{W}^{clip} = \{\mathbf{w}_k^{clip}\}_{k=1}^{N_c}$. Given an image $\mathbf{x}_i$ with corresponding label y_i, the visual embedding can be obtained by the visual encoder $\mathbf{f}(\cdot)$ as: $\mathbf{z}_i^{clip} = \mathbf{f}(\mathbf{x}_i)$, After that, the prediction probability for image $\mathbf{x}_i$ can be computed as follows:

$$\mathbf{p}_k(y \mid \mathbf{z}_i^{clip}; \mathbf{W}^{clip}) = \frac{\exp(< \mathbf{z}_i^{clip}, \mathbf{w}_k^{clip} > /\tau)}{\sum_{k=1}^{N_c} \exp(< \mathbf{z}_i^{clip}, \mathbf{w}_k^{clip} > /\tau)}, \tag{1}$$

where we can define $\mathbf{p} = [\mathbf{p}_1, \cdots, \mathbf{p}_k, \cdots, \mathbf{p}_{N_c}]$ as the collection of probabilities for classifying $\mathbf{x}_i$, and $\mathbf{p}_k$ in Eq. 1 represents the probability of $\mathbf{x}_i$ belonging to the k_{th} label. τ is a hyper-parameter to control the sharpness of the output, and $< \cdot, \cdot >$ is the dot product which can be termed as the cosine similarity as the features are normalized. Combined with cross-entropy loss, we can train the CLIP model or fine-tune it.

Text Prompt Tuning. Although Eq. 1 can be easily applied to Zero-Shot classification by employing a fixed hand-craft prompt, (*i.e.*, $\mathbf{t}$ ='*a photo of a [class]*'), to generate the textual embedding, it can not be well adapted to the downstream task. Therefore, the learnable prompt tuning method is proposed by learning a set of continuous vectors for generating task-related textual embedding, *e.g.*, CoOp [62]. Specifically, the learnable prompt for the input of the text encoder can be expressed as:

$$\mathbf{t}_k = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_L, \mathbf{CLS}_k], \quad \forall k \in 1 \sim N_c, \tag{2}$$

where $\mathbf{v}_i \in \mathbb{R}^d$ is a learnable vector and d is the prompt dimension, L is the length of the prompt, and $\mathbf{CLS}_k$ is the class token for the k_{th} text prompt $\mathbf{t}_k$. During training, only the learnable prompt is updated while the original visual and text encoder are frozen. The output of the text encoder can be represented as $\mathbf{w}_k = \mathbf{g}(\mathbf{t}_k)$. Specifically, we adopt the CoOp [62] mechanism to learn the text prompt enabling the CLIP model to better adapt to downstream tasks.

Visual Prompt Tuning. To fine-tune the visual encoder $\mathbf{f}(\cdot)$, we adopt the deep VPT technique [19] to insert a set of learnable prompts into the transformer layer of the visual encoder. VPT employs multiple layers of Transformers to capture the intricate relationship between images and text. Notably, it inserts learnable prompts between the patch embeddings and the class token at the input of each layer.

The VPT model links the output of the last class token to a fully connected layer and applies a softmax function to generate a probability over classes. During training, the model employs cross-entropy loss to compare the predicted class probability with the ground-truth label. It only updates the learnable prompt parameters while keeping the pre-trained model parameters unchanged.

3.2. GPT-Assist Text-Disentanglement

To learn disentangled prompt representation for DG, we first perform text disentanglement, as the text modality in the visual foundation model is inherently rich in semantic information and can be more easily disentangled.

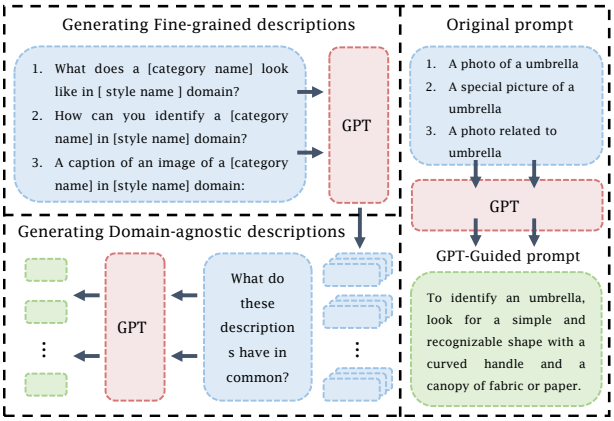

Figure 3. The figure illustrates the two components of text disentanglement. Firstly, fine-grained descriptions are generated for the input questions. Secondly, these fine-grained descriptions are summarized to generate domain-invariant descriptions. The right half of the figure demonstrates the distinction between the hand-craft description and the domain-invariant description.

To achieve text disentanglement, we first adopt GPT-3 to generate fine-grained descriptions for each domain-specific class and the specific domain itself. Firstly, we design input questions for GPT-3 to generate fine-grained descriptions by utilizing the following question formats: 1) *'What does a [class] look like in [domain]'*; 2) *'How can you identify an [class] in [domain]'*; 3) *'A caption of an image of an [class] in [domain]'*, as well as some other prompts: *'Please provide a detailed description of the object under this [domain] and [class]'*. After generating a sufficient number of fine-grained descriptions. We employ GPT-3 once more to summarize and analyze the shared and common attributes among all the fine-grained descriptions, finally obtaining more robust and domain-invariant descriptions for each class. Meanwhile, we also adopt GPT-3 to generate descriptions for each domain. Fig. 2 shows the process of text disentanglement by GPT-3. We can clearly see that the obtained domain-invariant description for the 'umbrella' by our method contains much more semantic information than the hand-craft descriptions.

After obtaining the domain-invariant description for each class, we generate the domain-invariant embedding for each class by the text encoder $\mathbf{g}(\cdot)$ of CLIP. Such domain-invariant embedding $\mathbf{w}_k^{clip}$ will later serve as guidance for learning domain-invariant visual prompts $\mathbf{t}_k$. Specifically, we adopt the $L2$ distillation loss [18] as follows:

$$\mathcal{L}_{kg1} = \frac{1}{N_c} \sum_{k=1}^{N_c} ||\tilde{\mathbf{w}}_k - \mathbf{w}_k^{clip}||_2^2, \tag{3}$$
$$where \quad \tilde{\mathbf{w}}_k = \mathbf{g}(\mathbf{t}_k), \mathbf{t}_k = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_L, \mathbf{CLS}_k],$$

N_c is the number of class, $||\cdot||_2^2$ is the $L2$ norm. Note

that $\mathbf{w}_k^{clip}$ is obtained by the original CLIP model. To learn the domain-invariant text prompt, we simultaneously optimize the above-mentioned knowledge distillation loss and the contrastive learning objective $\mathcal{L}_{ce1}$ between the textual and visual embeddings as follows:

$$\mathbf{p}_k(y \mid \mathbf{z}_i^{clip}; \tilde{\mathbf{W}}) = \frac{\exp(< \mathbf{z}_i^{clip}, \tilde{\mathbf{w}}_k > /\tau)}{\sum_{k=1}^{N_c} \exp(< \mathbf{z}_i^{clip}, \tilde{\mathbf{w}}_k > /\tau)}, \quad (4)$$

$$\mathcal{L}_{ce1} = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i \cdot \log \mathbf{p}(y \mid \mathbf{z}_i^{clip}; \tilde{\mathbf{W}}), \quad \mathbf{p} \in \mathbb{R}^{N_c}, \quad (5)$$

where $\mathbf{y}_i \in \mathbb{R}^{N_c}$ in Eq. 5 represents one-hot label and N_c is the label dimension. For each input image $\mathbf{x}_i$, the actual label $\mathbf{y}_i$ is assigned as $[1, 0, \cdots, 0]$.

Similarly, we learn the domain-specific textual embedding $\tilde{\mathbf{h}}_m$ for m_{th} domain with the learnable prompt under the guidance of $\mathbf{h}_m^{clip}$ generated by GPT-Assist domain-specific textual descriptions. Additionally, define the textual embedding for all domains as $\tilde{\mathbf{H}} = \{\tilde{\mathbf{h}}_m\}_{m=1}^{N_s}$. In line with training method for domain-invariant text prompts, we employ contrastive learning objective $\mathcal{L}_{ce2}$ and $L2$ distillation loss $\mathcal{L}_{kg2}$. Finally, the overall loss function arrives at:

$$\mathcal{L}_{text} = \mathcal{L}_{ce1} + \alpha_1 * \mathcal{L}_{ce2} + \beta_1 * (\mathcal{L}_{kg1} + \mathcal{L}_{kg2}). \quad (6)$$

3.3. Image-Disentanglement Guided by Text

After achieving the text disentanglement, we keep the text encoder and its domain-invariant and domain-specific text prompt fixed. Then, we perform the image disentanglement under the guidance of the disentangled textual embeddings. Specifically, we adopt the deep VPT as the visual encoder to extract the domain-invariant and domain-specific image features, denoted as $\tilde{\mathbf{z}}_i^I$ and $\tilde{\mathbf{z}}_i^S$:

$$\begin{aligned} \tilde{\mathbf{z}}_i^I &= \mathbf{f}_I(\mathbf{x}_i; \mathbf{E}_I), \\ \tilde{\mathbf{z}}_i^S &= \mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s), \end{aligned} \quad (7)$$

where $\mathbf{E}_I$ and $\mathbf{E}_s$ represent the learnable visual prompts in the domain-invariant and domain-specific visual encoders ($i.e.$, $\mathbf{f}_I$ and $\mathbf{f}_s$), respectively. To finetune the visual encoder $\mathbf{f}_I(\cdot)$ in Eq. 7, we adopt the following loss terms: 1) the contrastive learning objective $\mathcal{L}_{ce1}$ between $\tilde{\mathbf{z}}_i^I$ and $\tilde{\mathbf{w}}_k$ to optimize $\mathbf{E}_I$; 2) the domain confusion regularization $\mathcal{L}_{mix}$:

$$\mathbf{p}_m(y_d \mid \tilde{\mathbf{z}}_i^I; \tilde{\mathbf{H}}) = \frac{\exp(< \tilde{\mathbf{z}}_i^I, \tilde{\mathbf{h}}_m > /\tau)}{\sum_{m=1}^{N_s} \exp(< \tilde{\mathbf{z}}_i^I, \tilde{\mathbf{h}}_m > /\tau)}, \quad (8)$$

$$\mathcal{L}_{mix} = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{o}_i \cdot \log \mathbf{p}(y_d \mid \tilde{\mathbf{z}}_i^I; \tilde{\mathbf{H}}), \quad \mathbf{p} \in \mathbb{R}^{N_s}, \quad (9)$$

where y_d in Eq. 8 and Eq. 9 is the domain label for each input image $\mathbf{x}_i$. To train the domain confusion regulation, we set one-hot domain label $\mathbf{o}_i \in \mathbb{R}^{N_s}$ in Eq. 9 to $[1/N_s, 1/N_s, \cdots, 1/N_s]$ for all the images from different domains. 2) Similarly, to enhance the generalization ability of $\tilde{\mathbf{z}}_i^I$, we also utilize the $\mathcal{L}_{kg}$ loss [54] to reduce the distance between $\tilde{\mathbf{z}}_i^I$ and $\mathbf{z}_i^{clip}$: $\mathcal{L}_{kg} = ||\tilde{\mathbf{z}}_i^I - \mathbf{z}_i^{clip}||_2^2$.

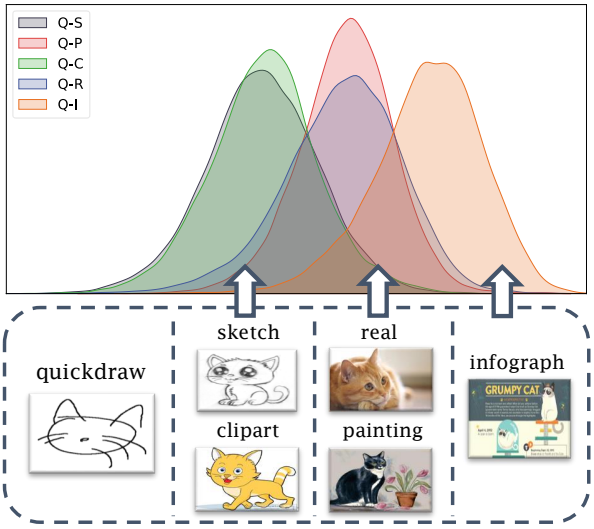

Figure 4. This figure demonstrates the distribution of distances between the quickdraw domain and the other five domains (sketch, clipart, infograph, painting, and real) within the same category in the DomainNet dataset. It is evident that quickdraw exhibits closer proximity to the sketch and clipart domains, while it is comparatively farther from the infograph domain.

In addition, we also train domain-specific visual encoder $\mathbf{f}_s(\cdot)$ in order to generate the prototype in Sec. 3.4 for subsequent steps. We employ the contrastive learning objective $\mathcal{L}_{ce2}$ to guide the training of domain-specific visual prompt. Finally, the overall loss function can be written as follows:

$$\mathcal{L}_{img} = \mathcal{L}_{ce1} + \alpha_2 * \mathcal{L}_{ce2} + \beta_2 * \mathcal{L}_{kg} + \beta_3 * \mathcal{L}_{mix}. \quad (10)$$

3.4. Domain-Specific Prototype Learning

To fully exploit the domain-specific information for DG, we further propose the relevance-inspired domain-specific prototype learning mechanism to devise a domain-specific predictor. As shown in Fig. 4, The domain-specific prediction is built on the observation that the distance between different domain distributions varies. Given an unseen domain image, we can intuitively adopt relevance-inspired prototype prediction to well utilize the domain-specific image feature, which could reduce the *distribution shift* between source and target domains during inference.

Prototype Initialization. Given the pre-trained domain-specific visual encoder $\mathbf{f}_s(\cdot)$, we generate the prototype for each class within each specific domain. Specifically, given N_c classes under N_s source domains, we can obtain a prototype tensor: $\mathcal{C} \in \mathbb{R}^{N_c N_s \times d}$ as follows:

$$\mathcal{C}_{m,k,:} = \frac{1}{M} \sum_{\mathbf{x}_i \in \mathcal{D}_m^S, \, y_i = k} \mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s), \quad (11)$$

where d is the feature dimension, y_i is the corresponding label of the input image $\mathbf{x}_i$, M is the total number of instances in the m_{th} source domain $\mathcal{D}_m^S$ with class label y_i. For simplicity, we re-write the prototype tensor $\mathcal{C}$ as a two-dimensional matrix as $\mathcal{C} \in \mathbb{R}^{N_c N_s \times d}$ in the following and store it in the memory bank during model training.

After constructing the prototype memory bank, we also transfer their corresponding labels into one-hot encoding as each prototype corresponds to one class label. Therefore, we can term the prototype learning as a key-value cache model [56], where the key is the prototype for each class $\mathcal{C} \in \mathbb{R}^{N_c N_s \times d}$, and their corresponding value is the label set $\mathbf{L}_c \in \mathbb{R}^{N_c N_s \times N_c}$. During inference, given a test image $\mathbf{x}_i$ which serves as the query for retrieving from the cache model, we first extract the domain-specific image features $\mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s) \in \mathbb{R}^{1 \times d}$ by the visual encoder $\mathbf{f}_s(\cdot)$. Then the domain-specific prediction $\mathbf{P}_i^S$ can be calculated as follows:

$$\mathbf{P}_i^S = \varphi(\mathbf{x}_i, \mathcal{C})\mathbf{L}_c, \tag{12}$$

where $\varphi(\mathbf{x}_i, \mathcal{C}) \in \mathbb{R}^{1 \times N_c N_s}$ denotes the affinities between the query feature $\mathbf{f}_s(\mathbf{x}_i)$ and the prototypes $\mathcal{C} \in \mathbb{R}^{N_c N_s \times d}$ stored in the memory bank. It can be calculated as:

$$\varphi(\mathbf{x}_i, \mathcal{C}) = \exp(-\beta(1 - \mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s)\mathcal{C}^\top)), \tag{13}$$

where β is a modulating hyper-parameter to control the sharpness of the similarity output. $\mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s)\mathcal{C}^\top$ can be viewed as the cosine similarity between the test image feature $\mathbf{f}_s(\mathbf{x}_i; \mathbf{E}_s)$ and the prototypes $\mathcal{C}$ for all domain classes, as both of the key and query features are $L2$ normalized. After that, the domain-specific prediction based on the cache model can be obtained by the linear combination of the domain-specific cache values $\mathbf{L}_c$ weighted by query-key similarities, as illustrated in Eq. 12.

Finally, to exploit the domain-invariant and specific information for DG, we make the final prediction by combining the domain-invariant and specific predictions as follows:

$$\mathbf{P}_i = \mathbf{P}_i^I + \alpha_3 * \mathbf{P}_i^S, \tag{14}$$

where $\mathbf{P}_i^I$ is the domain-invariant prediction and can be calculated as $\mathbf{P}_i^I = \mathbf{p}(y \mid \tilde{\mathbf{z}}_i^I; \tilde{\mathbf{W}})$ by utilizing the domain invariant visual features and the corresponding textual embeddings. α_3 is the trade-off parameter.

Besides, we further treat the prototypes $\mathcal{C} \in \mathbb{R}^{N_c N_s \times d}$ in the memory bank as learnable parameters with the mean features as initialization illustrated in Eq. 11, then we fine-tune $\mathcal{C}$ via SGD for several epochs. Updating the prototypes in the memory bank can boost the estimation of affinities, which is able to calculate the cosine similarities between the test feature and the prototypes more accurately. In contrast, the values $\mathbf{L}_c$ are one-hot encodings representing the ground-truth annotations and should be kept frozen to well memorize the category information during training.

4. Experiments

4.1. Datasets and Evaluation Protocols

In the experiment, we follow experimental settings of DomainBed. We adopt five commonly used datasets in domain generalization tasks for evaluation: PACS [28] (4 domains, 9,991 samples, 7 classes), VLCS [28] (4 domains, 10,729 samples, 5 classes), OfficeHome [50] (4 domains, 15,588 samples, 65 classes), DomainNet [41] (6 domains, 586,575 samples, 345 classes) and TerraIncognita [5] (4 domains, 24,788 samples, 10 classes). In domain generalization experiments, it is customary to designate one domain as the target domain while considering the remaining domains as the source domains. To ensure reliable results, we calculate the average performance across multiple experiments.

4.2. Implementation details

We choose ViT-B/16 as a backbone network, the same as the baseline model (CLIP) [44]. To construct prompts for the text and visual encoders, we refer to the open-source implementations of CoOp and VPT. In our experiments, we adopt a few-shot training strategy where each class is randomly sampled with 16 shots. For DPR, we set the text and visual prompt lengths to 16 and 2, respectively. The hyper-parameter α_1 and β_1 in Eq. 6 for text disentanglement is set to 1.0 and 8.0. The hyper-parameter α_2, β_2, and β_3 in Eq. 10 for image disentanglement is set to 0.8, 2.0, and 0.8. The hyper-parameter α_3 in Eq. 14 for DSPL is set to 5.0. The models are trained for 10 epochs using a batch size of 128 and a learning rate of 0.002 for each stage. We utilize the SGD optimizer and conduct the training on a single NVIDIA RTX3090 GPU. To ensure the reliability of our proposed algorithms, we independently repeat all experiments three times and report the average results obtained from these trials.

4.3. Comparison with SOTA Methods

The test accuracies of DPR and recent domain generalization (DG) methods [2, 7, 14, 25, 26, 31, 40, 44, 48, 49, 53, 57, 61] on five benchmark datasets are reported in Tab. 1. Notably, DPR achieves the highest average performance and set new state-of-the-art (SOTA) results at 97.45%, 86.43%, 86.13%, 62.05%, and 57.10% on all benchmark datasets. The proposed method has three main advantages: (1) Unlike DADA [42], our method fully utilizes the disentangled descriptions and does not require the design of complex disentangling losses. (2) DPR fully leverages both domain-specific and domain-invariant information and cleverly combines them during inference. (3) Distinguished from other methods that use large-scale pre-trained models, we achieve superior generalization performance by using less training data. Overall, our method successfully leverages the generalization ability of large-scale

Table 1. Comparsion with the state-of-the-art methods on PACS, VLCS, OfficeHome, DomainNet and TerraInc.

Method	Venue	PACS	VLCS	OfficeHome	DomainNet	TerraInc	Avg
ResNet-50 Pre-trained by ImageNet.							
DANN [14]	IJCAI'16	83.60 ±0.4	78.60 ±0.4	65.90 ±0.6	38.30 ±0.1	46.40 ±0.5	65.56
Fish [48]	ICML'22	85.50 ±0.3	77.80 ±0.3	68.60 ±0.4	42.70 ±0.2	45.10 ±1.3	63.94
DAC-SC [25]	CVPR'23	87.50 ±0.1	78.70 ±0.3	70.30 ±0.2	44.90 ±0.1	46.50 ±0.3	65.60
SAGM [53]	CVPR'23	86.60 ±0.1	80.00 ±0.1	70.10 ±0.1	45.00 ±0.1	48.80 ±0.1	66.10
ViT-B/16 Pre-trained by CLIP.							
SWAD [7]	NIPS'21	91.30 ±0.1	79.40 ±0.4	76.90 ±0.1	51.70 ±0.8	45.40 ±0.5	68.94
CLIP [44]	-	96.20 ±0.1	81.70 ±0.1	82.00 ±0.1	57.50 ±0.1	33.40 ±0.1	70.16
SMA [2]	NIPS'22	92.10 ±0.2	79.70 ±0.2	78.10 ±0.1	55.90 ±0.2	48.30 ±0.7	70.82
ERM [49]	ICLR'21	93.70 ±0.1	82.70 ±0.1	78.50 ±0.1	53.80 ±0.1	52.30 ±0.1	72.20
DUPRG [40]	ICLR'23	97.10 ±0.2	83.90 ±0.5	83.60 ±0.3	59.60 ±0.3	42.00±1.3	73.24
CoOp [61]	IJCV'22	96.20 ±0.1	77.60 ±0.2	83.90 ±0.1	59.80 ±0.1	48.8 ±0.1	73.26
MIRO [7]	ECCV'22	95.60 ±0.8	82.20 ±0.3	82.50 ±0.1	54.00 ±0.3	54.30 ±0.4	73.72
SEDGE [31]	-	96.10 ±0.1	82.20 ±0.1	80.70 ±0.2	54.70 ±0.1	56.80 ±0.3	74.10
DPL [57]	-	97.30 ±0.2	84.30 ±0.4	84.20 ±0.2	56.70 ±0.1	52.60 ±0.6	75.02
GESTUR [26]	ICCV'23	96.00 ±0.0	82.80 ±0.1	84.20 ±0.1	58.90 ±0.1	55.70 ±0.2	75.52
Ours	-	**97.45** ±0.1	**86.43** ±0.3	**86.13** ±0.2	**62.05** ±0.1	**57.10** ±0.2	**77.83**

Table 2. Ablation study on individual components of our method on VLCS (VL), OfficeHome (OH), and DomainNet (DN).

GPT-Assist	VPT	$\mathcal{L}_{\mathbf{mix}}$	DSPL	VL	OH	DN	Avg.
Text-Disentanglement only.							
–	–	–	–	77.6	83.9	59.8	73.8
✓	–	–	–	79.0	85.2	61.5	75.3
Image-Disentanglement only.							
–	✓	–	–	84.9	85.2	59.8	76.6
–	✓	✓	–	85.1	85.4	60.1	76.9
–	✓	–	✓	85.5	85.5	60.5	77.2
–	✓	✓	✓	86.0	85.7	60.6	77.4
Image and Text-Disentanglement.							
✓	✓	–	–	85.3	85.5	61.5	77.4
✓	✓	✓	–	85.6	85.7	61.8	77.7
✓	✓	–	✓	86.0	85.9	62.0	78.0
✓	✓	✓	✓	86.4	86.1	62.1	78.2

pre-trained models and demonstrates the effectiveness of our hypothesis compared to other baseline methods.

4.4. Ablation Study

To validate the effectiveness of each component of our method, we conduct an ablation experiment on VLCS, OfficeHome, and DomainNet datasets in Tab. 2. Our baseline (see 1^{st} row) for text disentanglement involves CoOp with learnable domain-invariant and domain-specific text prompts but does not incorporate the guidance of GPT-Assist disentangled descriptions.

The effectiveness of Text-Disentanglement. Our text disentanglement achieves improvements of +1.4%, +1.3%, and +1.7% with baseline (see 1^{st} row and 2^{nd} row in Tab. 2) and our Image-Disentanglement achieved improvements of +0.5%, +0.3% and +0.7% with text guidance (see 4^{th} row and 8^{th} row in Tab. 2). This observation indicates that the text modality is easy to disentangle and the text prompt plays a vital role in guiding the process of image disentanglement. We further investigate the impact of text descriptions on disentanglement in Tab. 3. Under the conditions of hand-craft, fine-grained, and domain-invariant descriptions, we find that using domain-invariant descriptions achieves the best performances. Furthermore, we demonstrate that as the descriptions L_a generated by GPT-3 increases, the results also improves. This result indicates that by increasing the generation of fine-grained descriptions, we can extract a larger number of domain-invariant descriptions, which in turn leads to improved classification performance.

Table 3. Analysis of the influence of the parameter L_a on text disentanglement. Here, L_a represents the number of responses generated by GPT-3 for each input question.

method		VL	OH	DN	Avg.
Baseline		77.64	83.92	59.83	73.80
+ hand-craft description		78.02	84.48	60.45	74.32
L_a = 1	+ fine-grained	78.24	84.67	60.66	74.52
	+ domain-invariant	78.56	84.95	61.01	74.84
L_a = 3	+ fine-grained	78.35	84.83	60.84	74.67
	+ domain-invariant	78.74	85.17	61.27	75.06
L_a = 5	+ fine-grained	78.51	84.99	60.98	74.83
	+ domain-invariant	78.89	85.22	61.39	75.17
L_a = 7	+ fine-grained	78.55	85.09	61.04	74.89
	+ domain-invariant	79.00	85.24	61.57	75.27
L_a = 9	+ fine-grained	78.54	85.11	61.10	74.91
	+ domain-invariant	78.95	85.29	61.49	75.24
L_a = 11	+ fine-grained	78.59	85.07	61.08	74.91
	+ domain-invariant	78.91	85.33	61.54	75.26

The effectiveness of Image-Disentanglement. Our image disentanglement achieves +0.2%, +0.2% and +0.3% improvement without text disentanglement (see 4^{th} row and 6^{th} row in Tab. 2) and achieves +0.3%, +0.2% and +0.3% improvement with Text-Disentanglement (see 7^{th} row and 8^{th} row in Tab. 2). We select three domains: real, clipart,

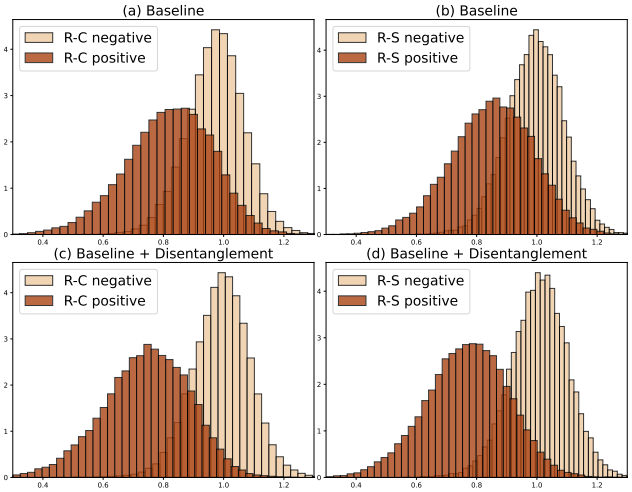

Figure 5. The distance distributions of randomly selected positive and negative pairs between the real domain and clipart domain, as well as between the real domain and sketch domain in the DomainNet dataset, were visualized.

and sketch from DomainNet and define samples in different domains but the same category as positive pairs and those in different categories as negative pairs. We calculate the distances with 34,500 positive and negative pairs before and after disentanglement in Fig. 5. The results show significant decrease in the distance between positive samples after disentanglement, which confirms the correctness of our assumption regarding text disentanglement.

Table 4. Analysis of the influence of both training and training-free conditions on DSPL.

method	VL	OH	DN	Avg.
DSPL (training-free)	85.39	85.53	60.28	77.07
DSPL (training)	85.99	85.65	60.66	77.43

The effectiveness of Domain-Specific Prototype Learning. The DSPL module provides a performance gain of +0.9%, +0.3%, and +0.5% when directly adding it to our baseline (see 4^{th} row and 6^{th} row in Tab. 2). When based on the text disentanglement, the DSPL further improves the performance of +0.8%, +0.4%, and +0.3% (see 8^{th} row and 10^{th} row in Tab. 2). This indicates that utilizing partial domain-specific information is advantageous for classification. Furthermore, we conduct experiments to evaluate the effectiveness of DSPL under two different conditions: 'training-free' and 'training-based' in Tab. 4. The experimental results indicate that, under the given training conditions, the performance improvement achieved by DSPL is +0.60%, +0.12%, and +0.38%, respectively, compared to the training-free baseline.

4.5. Parameter Analysis.

The proposed method includes six parameters in DPR, i.e., α_1 and β_1 in Eq. 6, α_2, β_2 and β_3 in Eq. 10, α_3 in Eq. 12. To study the effect of the above six parameters, we set them to different values, as shown in Fig. 6. We evaluate

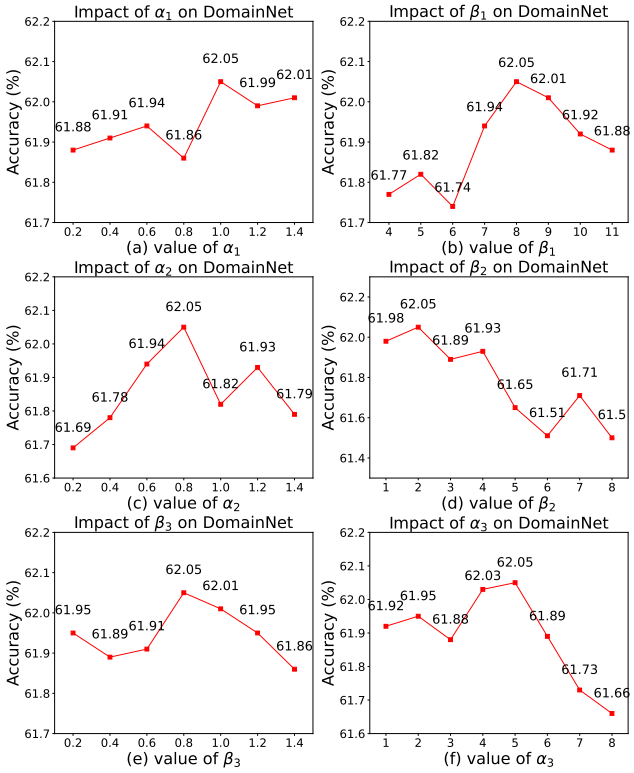

Figure 6. Parameter analysis of α_1, α_2, β_1, β_2 β_3 and α_3.

the effect of α_1 when $\beta_1 = 8.0$ in Fig. 6 (a). We evaluate the effect of β_1 when $\alpha_1 = 1.0$ in Fig. 6 (b). We evaluate the effect of α_2 when $\beta_2 = 2.0$, $\beta_3 = 0.8$ in Fig. 6 (c). We evaluate the effect of β_2 when $\alpha_2 = 0.8$, $\beta_3 = 0.8$ in Fig. 6 (d). We evaluate the effect of β_3 when $\alpha_2 = 0.8$, $\beta_2 = 2.0$ in Fig. 6 (e). Finally, we find when $\alpha_1 = 1.0$, $\beta_1 = 8.0$, $\alpha_2 = 0.8$, $\beta_2 = 2.0$, $\beta_3 = 0.8$, and $\alpha_3 = 5.0$, our method achieves the best performance.

5. Conclsion

In this paper, we propose the disentangled prompt representation based on pre-trained LLM for DG. Our method leverages GPT-Assist text disentanglement to learn domain-invariant and domain-specific visual representations. Furthermore, through analyzing the distributional differences between domains, we introduce relevance-inspired domain-specific prototype learning to effectively utilize domain-specific information. Extensive experiments on the DomainBed dataset have demonstrated that our framework outperforms existing state-of-the-art DG methods.

Acknowledgements: This work was supported in part by the National Key R&D Program of China under Grant No.2023YFA1008600, in part by NSFC under Grant NO.62176198 and U22A2096, in part by the Key R&D Program of Shaanxi Province under Grant 2024GX-YBXM-135, in part by the Key Laboratory of Big Data Intelligent Computing under Grant BDIC-2023-A-004.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2

[2] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. In *NeurIPS*, 2022. 6, 7

[3] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. pages 528–539, 2020. 2

[4] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, pages 1006–1016, 2018. 2

[5] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, pages 456–473, 2018. 6

[6] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019. 2

[7] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *ECCV*, pages 440–457, 2022. 1, 2, 6, 7

[8] Jin Chen, Zhi Gao, Xinxiao Wu, and Jiebo Luo. Meta-causal learning for single domain generalization. In *CVPR*, pages 7683–7692, 2023. 2

[9] Sentao Chen, Lei Wang, Zijie Hong, and Xiaowei Yang. Domain generalization by joint-product distribution alignment. *PR*, 134:109086, 2023.

[10] De Cheng, Jingyu Zhou, Nannan Wang, and Xinbo Gao. Hybrid dynamic contrast and probability distillation for unsupervised person re-id. *IEEE TIP*, 31:3334–3346, 2022.

[11] De Cheng, Lingfeng He, Nannan Wang, Shizhou Zhang, Zhen Wang, and Xinbo Gao. Efficient bilateral cross-modality cluster matching for unsupervised visible-infrared person reid. pages 1325–1333, 2023.

[12] De Cheng, Gerong Wang, Bo Wang, Qiang Zhang, Jungong Han, and Dingwen Zhang. Hybrid routing transformer for zero-shot learning. *PR*, 137:109270, 2023. 2

[13] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pages 6447–6458, 2019. 2

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. 17(59):1–35, 2016. 2, 6, 7

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014. 3

[16] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727, 2022. 4

[20] Xueying Jiang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Domain generalization via balancing training difficulty and model capability. In *ICCV*, pages 18993–19003, 2023. 2

[21] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, pages 158–171, 2012. 2

[22] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[23] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NeurIPS*, pages 3581–3589, 2014. 3

[24] David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). pages 5815–5826, 2021. 2

[25] Sangrok Lee, Jongseong Bae, and Ha Young Kim. Decompose, adjust, compose: Effective normalization by playing with frequency for domain generalization. In *CVPR*, pages 11776–11785, 2023. 6, 7

[26] Byounggyu Lew, Donghyun Son, and Buru Chang. Gradient estimation for unseen domain risk minimization with pre-trained models. pages 4438–4448, 2023. 1, 2, 6, 7

[27] Chenming Li, Daoan Zhang, Wenjian Huang, and Jianguo Zhang. Cross contrasting feature perturbation for domain generalization. In *ICCV*, pages 1327–1337, 2023. 2

[28] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5542–5550, 2017. 6

[29] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pages 3490–3497, 2018. 2

[30] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *AAAI*, pages 3579–3587, 2018. 2

[31] Ziyue Li, Kan Ren, Xinyang Jiang, Bo Li, Haipeng Zhang, and Dongsheng Li. Domain generalization using pretrained models without fine-tuning. *arXiv preprint arXiv:2203.04600*, 2022. 6, 7

[32] Ziyue Li, Kan Ren, Xinyang Jiang, Yifei Shen, Haipeng Zhang, and Dongsheng Li. Simple: Specialized model-sample matching for domain generalization. In *ICLR*, 2022. 2

[33] Shiqi Lin, Zhizheng Zhang, Zhipeng Huang, Yan Lu, Cuiling Lan, Peng Chu, Quanzeng You, Jiang Wang, Zicheng Liu, Amey Parulkar, et al. Deep frequency filtering for domain generalization. In *CVPR*, pages 11797–11807, 2023. 2

[34] Alexander H. Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *NeurIPS*, pages 2595–2604, 2018. 3

[35] Fangrui Lv, Jian Liang, Shuang Li, Jinming Zhang, and Di Liu. Improving generalization with domain convex game. In *CVPR*, pages 24315–24324, 2023. 2

[36] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders, 2016. 3

[37] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. 2016. 3

[38] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. pages 10–18, 2013. 2

[39] Hiroki Naganuma and Ryuichiro Hataya. An empirical investigation of pre-trained model selection for out-of-distribution generalization and calibration. *arXiv preprint arXiv:2307.08187*, 2023. 2

[40] Hongjing Niu, Hanting Li, Feng Zhao, and Bin Li. Domain-unified prompt representations for source-free domain generalization. *arXiv preprint arXiv:2209.14926*, 2022. 2, 6, 7

[41] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 6

[42] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. pages 5102–5112, 2019. 1, 2, 3, 6

[43] Sanqing Qu, Yingwei Pan, Guang Chen, Ting Yao, Changjun Jiang, and Tao Mei. Modality-agnostic debiasing for single domain generalization. In *CVPR*, pages 24142–24151, 2023. 2

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. pages 8748–8763, 2021. 2, 6, 7

[45] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. pages 1278–1286, 2014. 3

[46] Mattia Segu, Alessio Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization. *PR*, 135:109115, 2023. 2

[47] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018. 2

[48] Yuge Shi, Jeffrey Seely, Philip H. S. Torr, Siddharth Narayanaswamy, Awni Y. Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *ICLR*, 2022. 6, 7

[49] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 2, 6, 7

[50] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 6

[51] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, pages 6677–6686, 2020. 2

[52] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *CVPR*, pages 3769–3778, 2023. 2

[53] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *CVPR*, pages 3769–3778, 2023. 6, 7

[54] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *CVPR*, pages 6757–6767, 2023. 5

[55] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. In *NeurIPS*, pages 23664–23678, 2021. 2

[56] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510, 2022. 6

[57] Xin Zhang, Shixiang Shane Gu, Yutaka Matsuo, and Yusuke Iwasawa. Domain prompt learning for efficiently adapting clip to unseen domains. *Transactions of the Japanese Society for Artificial Intelligence*, 38(6):B–MC2_1, 2023. 6, 7

[58] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *CVPR*, pages 16036–16047, 2023. 2

[59] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. In *NeurIPS*, 2020. 2

[60] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 2

[61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 6, 7

[62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 4