ProD: Prompting-to-disentangle Domain Knowledge for Cross-domain Few-shot Image Classification

Tianyi Ma[1,2], Yifan Sun[2], Zongxin Yang[3], Yi Yang[3]
University of Technology Sydney[1], Baidu Inc.[2], Zhejiang University[3]
tianyi.ma@student.uts.edu.au, sunyf15@tsinghua.org.cn

Abstract

This paper considers few-shot image classification under the cross-domain scenario, where the train-to-test domain gap compromises classification accuracy. To mitigate the domain gap, we propose a prompting-to-disentangle (ProD) method through a novel exploration with the prompting mechanism. ProD adopts the popular multi-domain training scheme and extracts the backbone feature with a standard Convolutional Neural Network. Based on these two common practices, the key point of ProD is using the prompting mechanism in the transformer to disentangle the domain-general (DG) and domain-specific (DS) knowledge from the backbone feature. Specifically, ProD concatenates a DG and a DS prompt to the backbone feature and feeds them into a lightweight transformer. The DG prompt is learnable and shared by all the training domains, while the DS prompt is generated from the domain-of-interest on the fly. As a result, the transformer outputs DG and DS features in parallel with the two prompts, yielding the disentangling effect. We show that: 1) Simply sharing a single DG prompt for all the training domains already improves generalization towards the novel test domain. 2) The cross-domain generalization can be further reinforced by making the DG prompt neutral towards the training domains. 3) When inference, the DS prompt is generated from the support samples and can capture test domain knowledge through the prompting mechanism. Combining all three benefits, ProD significantly improves cross-domain few-shot classification. For instance, on CUB, ProD improves the 5-way 5-shot accuracy from 73.56% (baseline) to 79.19%, setting a new state of the art.

1. Introduction

Few-shot image classification aims to use very limited support samples to transfer the classifier from base training classes to novel test classes [10, 27, 28, 32, 34], which meets the requirement in application scenarios when training data is scarce. However, besides the insufficient data, in real-world applications, another critical challenge is the cross-domain problem, *i.e.*, there is usually a domain gap between the training set and the test set. This train-to-test domain gap further hinders the knowledge transfer between the training and test data, significantly compromising the classification accuracy [7, 12]. In this paper, we tackle the cross-domain problem for few-shot image classification.

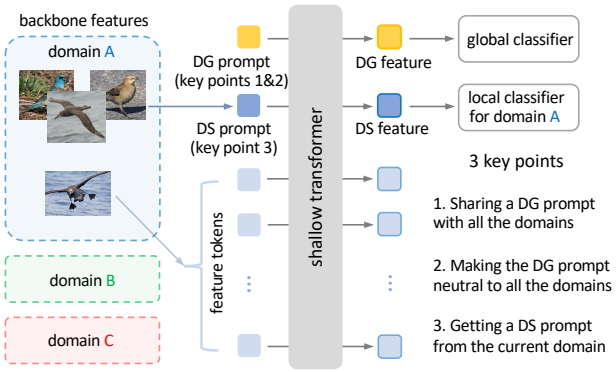

Figure 1. ProD flattens the CNN backbone feature into feature tokens and concatenates them with the DG and DS prompt. The DG prompt is learnable and shared by all the training domains for general knowledge. In contrast, the DS prompt is generated from same-domain features and thus can capture novel test domain knowledge from the support images during the test. The output of the DG/DS prompt is respectively supervised with a global/local classification head during training and concatenated as the final representation for inference.

Generally, there are two approaches for mitigating the domain gap, *i.e.*, domain generalization, and domain adaptation. Domain generalization improves the inherent generalization ability of the learned feature and directly applies it to novel domains without further tuning. In contrast, the domain adaptation uses samples from the novel domain to fine-tune the already-learned feature. For few-shot image classification, the domain generalization approach [14, 21, 29, 41] is more explored than the domain adaptation approach [12], because limited support samples hardly provide reliable clues for domain adaptation.

This paper proposes a prompting-to-disentangle (ProD) method through a novel exploration with the prompting mechanism. The prompting technique was first introduced in natural language processing and has become popular in computer vision [16, 43]. It aims to switch the transformer to different mapping functions without changing its parameters by using different prompts to condition (impact) the transformer. Compared with prior prompting techniques, our exploration is novel: with a single transformer, we simultaneously use two prompts to extract the domain-general (DG) knowledge and the domain-specific (DS) knowledge in parallel. Therefore, these two prompts switch a single transformer between two different outputs simultaneously, *i.e.*, DG and DS knowledge, yielding the so-called Prompting-to-Disentangle. The related work section (Sec.2.3) compares our method and the standard prompting mechanism.

Importantly, in ProD, both the DG and DS knowledge are beneficial, contrary to prior works [14, 19, 22] where the DS knowledge is harmful and discarded. The reason is: in ProD, the DS knowledge is not bound by the already-seen training domains. Instead, it can on-the-fly capture the novel domain knowledge from support samples through the prompting mechanism (as explained in the 3rd benefit below). Therefore, ProD benefits from the DS knowledge of the novel test domain. Specifically, as illustrated in Fig.1, ProD adopts the popular multi-domain training scheme [14, 29] and uses a Convolutional Neural Network (ResNet-10 [13]) to extract the backbone feature. Afterward, ProD flattens the backbone feature into multiple feature tokens, concatenates the feature tokens with a DS and a DG prompt, and feeds them into a lightweight transformer. The DG prompt is learnable and shared by all the training domains, while the DS prompt is generated with backbone features from the domain-of-interest (*i.e.*, the domain of the feature tokens) on the fly. In ProD, there are three key points for mitigating the domain gap:

1) Sharing a single prompt for all the training domains benefits cross-domain generalization. In other words, we need no special design to obtain a DG prompt but only to share a single prompt with multiple domains. During training, the output state of the DG prompt (*i.e.* the DG feature in Fig.1) is fed into a global classifier that contains the categories from all the training domains. Inference with the DG feature improves classification accuracy.

2) The DG prompt can be further reinforced by making it neutral towards all the training domains. To this end, we enforce a simple constraint: the learned DG prompt should have identical (or close) distance toward all the training domains. This constraint reduces the bias toward any domain and enriches the domain-general knowledge, bringing another round of improvement.

3) The DS prompt can capture the DS knowledge from the domain-of-interest on the fly and thus makes the DS knowledge beneficial. Specifically, during training, given an input, we use features from the same domain to generate a DS prompt. Correspondingly, the knowledge in the DS prompt is from the input domain specifically rather than from all the training domains. Moreover, the output DS feature is supervised by a local classifier, which contains only the categories in the current domain and thus avoids cross-domain interference. In the inference phase, we duplicate the DS prompt generation procedure onto the test domain, *i.e.*, generating the DS prompt from the support samples. Therefore, although the model remains unchanged, the on-the-fly DS prompt modifies the context of the model input and dynamically conditions the output to the test domain. Such a prompting and conditioning effect can be viewed as a test-time adaptation without fine-tuning the model.

ProD concatenates the DG and DS features as the final representation for inference, therefore integrating the benefits of good generalization and fast adaptation. Consequently, ProD effectively mitigates the domain gap and improves cross-domain few-shot classification. For example, on CUB, ProD improves the 5-way 5-shot recognition accuracy from 73.56% to 79.19% on CUB dataset, setting a new state of the art.

Our contributions can be summed as follow:

- We propose a Prompting-to-Disentangling (ProD) method for cross-domain few-shot image classification. ProD disentangles the domain-general (DG) and domain-specific (DS) knowledge through a novel exploration of the prompting mechanism.
- For the DG knowledge, we show that sharing and neutralizing a DG prompt for all the training domains benefits cross-domain generalization. For the DS knowledge, we condition model to the novel test domain through a DS prompt generated on-the-fly to replace fine-tuning.
- We conduct extensive experiments to validate the effectiveness of ProD. Ablation studies show that both the DG and DS prompt in ProD are effective.

2. Related Work

2.1. Few-shot Image Classification

Few-shot image classification aims at classifying images from novel categories with limited labeled samples. There are two few-shot learning schemes: 1) meta-learning and 2) transfer learning. Meta-learning [7, 10, 27, 28, 32, 34] focus on applying metric learning to simulate few-shot scenarios in the training phase and upgrading the training optimizer. Transfer learning [38], on the other hand, solves the problem in a transductive way by re-training a new classifier [7] or adapter [39] for the novel domain while fixing other parts of the network. [7, 12] point out that the meta-learning-based methods underperform the transfer learning

methods on the cross-domain scenario. In this paper, we follow the standard transfer learning routine.

2.2. Domain Generalization and Adaptation

Domain generalization [36] aims to improve cross-domain performance without needing to access the target domain. Generalization can be achieved by data augmentation and generation [21, 33], domain-invariant representation learning & feature disentangle [1, 3, 19], and general training strategies [9, 37]. In the few-shot learning scenario, [21] proposes a noise-enhanced supervised autoencoder to generate augmentation samples. [14] use the model with average parameters from multiple iterations as the general domain teacher. In contrast, domain adaptation relies on the target domain data to transfer an already-trained model from the source domain to the target domain [2, 6, 11, 15, 25, 30, 40]. [20] applies an adapter to transform the general feature into a specific one for few-shot learning. The proposed ProD may be viewed as combining the benefits of domain generalization (through DG knowledge) and domain adaptation (through DS knowledge). We note that there is no fine-tuning to achieve the domain adaptation effect: When ProD is applied to the novel test (target) domain, the entire model for extracting the feature is fixed. Instead, ProD conditions the DS knowledge to the novel test domain by generating a DS prompt on-the-fly.

2.3. Prompting Mechanism

The prompting technique [5] was first introduced in natural language processing. It modifies the pre-trained language model for different downstream tasks by changing the prompt instead of tuning the deep model. Recently, the prompting mechanism has been applied in vision tasks for efficient fine-tuning [16]. Compared with the existing methods [23], our ProD has close connections and significant differences. Similar to prior works, in ProD, the prompt changes the mapping function of the deep model by modifying the context of the model input and does not change the model parameters. Still, there are two significant differences regarding training and inference. 1) *Training*: recent prompting techniques usually require a pre-trained model. Then, the prompts are injected and tuned for novel downstream tasks. When tuning the prompt, the pre-trained model is frozen. In contrast, in the proposed ProD, the "base model" (the CNN and the transformer head) and the prompts are trained simultaneously from scratch in an end-to-end manner. 2) *Inference*: Different prompts are usually employed separately in recent popular prompting techniques. In contrast, ProD simultaneously injects two prompts to activate two different knowledge in parallel. Moreover, the prompting objective, *i.e.*, using different prompts to disentangle domain-general and domain-specific knowledge is novel, so far as we know.

3. Methodology

3.1. Problem Formulation

In this paper, we adopt the popular multi-domain training paradigm [14, 17, 19, 24] for solving the cross-domain few-shot learning task. Specifically, we use a group of training datasets corresponding to different domains $\mathcal{D} = \{\mathtt{D}_0, \mathtt{D}_1, ..., \mathtt{D}_N\}$. In each training iteration, we randomly sample a dataset and conduct a few-show learning episode on the corresponding domain.

The test dataset is from a novel domain $\mathtt{D}_t$ and contains images with disjoint labels.

In the test phase, only a few labeled samples are provided for fine-tuning the model. Each testing episode performs a C-way K-shot task by randomly sampling a support set and a query set from the test domain $\mathtt{D}_t$. The support set consists of $C \times K$ (C is the number of classes, and each class has K samples) samples, and the query set consists of multiple unlabeled images from these C classes.

3.2. Overall Pipeline of ProD

The overall pipeline of ProD is illustrated in Fig.2 (a). In each training iteration, we randomly select a training domain $\mathtt{D}_n \in \mathcal{D}$ and sample multiple images $\{x_i^n\}$ (the superscript n indicates the n-th domain). ProD feeds these images into a CNN backbone, denoted as $f()$, to produce their backbone features. Correspondingly, for each image x (the superscript and upper-script are omitted), its backbone feature is a convolutional feature map $f(x) \in \mathbb{R}^{D\times H\times W}$ (D, H, W are the dimension, height, and width).

Given a backbone feature $f(x)$, ProD flattens it into a feature sequence consisting of $H \times W$ tokens (each token is D-dimensional), *i.e.*, $\mathbf{F} \in \mathbb{R}^{(HW)\times D}$. These feature tokens are then concatenated with a DG prompt ($\mathbf{G}$) and a DS prompt ($\mathbf{S}$). Since the concatenated tokens are to be input into the multi-block transformer, we add a superscript "0" to indicate their position (*i.e.*, $\mathbf{F}^0$, $\mathbf{G}^0$ and $\mathbf{S}^0$). The DG prompt $\mathbf{G}^0$ consists of multiple tokens and is learnable (Sec.3.3). The DS prompt $\mathbf{S}^0$ is generated from some other backbone features in the current domain $\mathtt{D}^n$ during training (Sec.3.4). During testing, $\mathbf{S}^0$ is generated from the support samples in $\mathtt{D}_t$ to capture the novel domain knowledge. ProD feeds the concatenated tokens into a transformer with L blocks (we empirically set $L = 2$), which is formulated as:

$$[\mathbf{F}^l, \mathbf{G}^l, \mathbf{S}^l] = B_l([\mathbf{F}^{l-1}, \mathbf{G}^{l-1}, \mathbf{S}^{l-1}]), \tag{1}$$

where B_l ($l = 1, 2, \cdots, L$) is the l-th transformer block, $[\ \ ]$ is the concatenation operation.

During training, the output state of GD and GS prompts (*i.e.*, $\mathbf{G}^L$ and $\mathbf{S}^L$) are fed into a global and local classification head, respectively. The global classification head

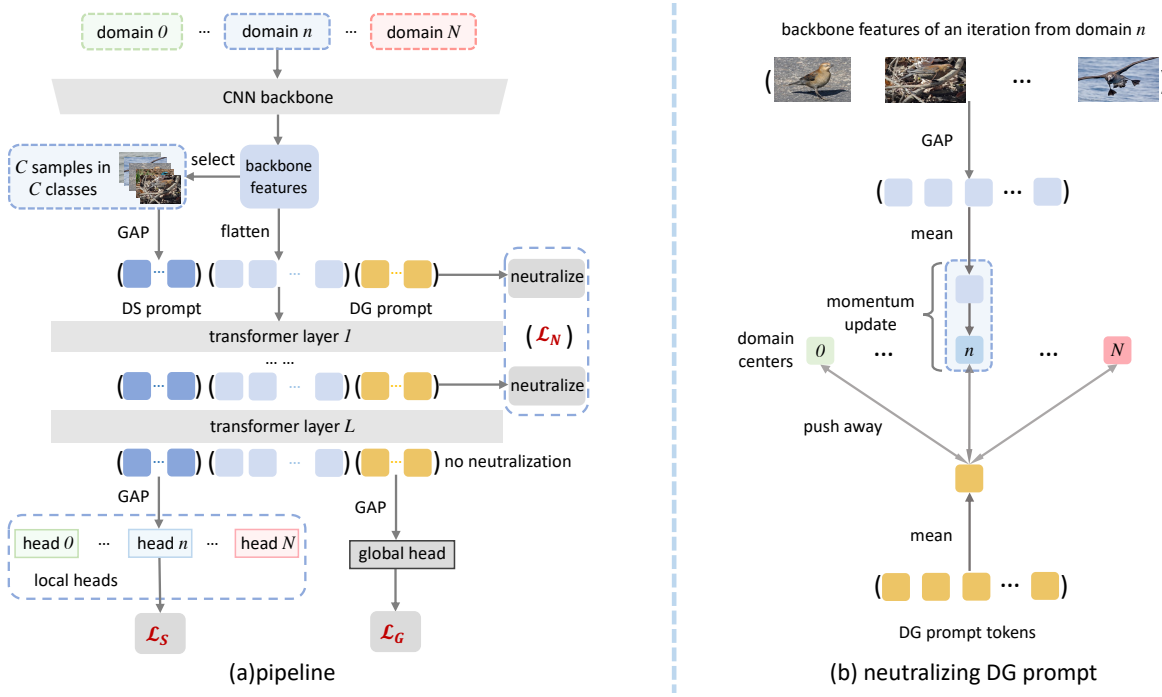

Figure 2. The architecture of ProD. (a) and (b) respectively depict the overall pipeline and the DG prompt neutralization. GAP represents global average pooling. In (a), ProD first extracts the backbone feature with a CNN backbone. Then, it flattens the backbone feature into feature tokens and concatenates them with a learnable DG prompt and an on-the-fly DS prompt. The outputs of the DG / DS prompt are fed into a global / local classification head, respectively. In (b), we maintain the domain centers using momentum update, use the domain centers to push the DG prompt (in all the transformer blocks except the final output state) far away, therefore neutralizing the DG prompt.

$\mathtt{H}^{\mathtt{global}}$ contains the holistic classes from all the domains $\mathcal{D} = \{\mathtt{D}_0, \mathtt{D}_1, \cdots, \mathtt{D}_N\}$. In contrast, the local classification head $\mathtt{H}^n$ only contains the classes from the current domain $\mathtt{D}_n$ ($n \in 1, 2, \cdots, N$).

During testing, we use the concatenation of $\mathbf{G}^L$ and $\mathbf{S}^L$ as the final representation and discard $\mathbf{F}^l$. $\mathbf{G}^L$ contains domain-general knowledge, while $\mathbf{S}^L$ contains domain-specific knowledge conditioned by the novel test domain. Combining these two features brings complementary benefits for cross-domain evaluation (Sec.4.3).

The following Sec.3.3 elaborates on the DG prompt, neutralizing DG prompt for better domain generalization and the corresponding global classification head. Sec.3.4 elaborates on the DS prompt and the corresponding local classification head.

3.3. Learning Domain-General Feature

3.3.1 DG Prompt and Global Classification

Domain-General prompt $\mathbf{G}^0$ contains multiple trainable tokens. When the feature tokens and the DG tokens ($[\mathbf{F}^0, \mathbf{G}^0]$) proceed in the transformer, they interact with each other through the attention mechanism. After L transformer blocks (Eqn.1), we consider the output $\mathbf{G}^L$ as containing domain-general knowledge and use a global classification head to supervise $\mathbf{G}^L$, which is formulated as:

$$\mathcal{L}_G = -\log \frac{\exp\left(\mathbf{w}_y \cdot \overline{\mathbf{G}^L}\right)}{\sum_j \exp\left(\mathbf{w}_j \cdot \overline{\mathbf{G}^L}\right)}, \tag{2}$$

where $\mathbf{w}_j$ enumerates all the weight vectors in the global classification head. $\mathbf{w}_y$ is the weight vector of the ground-truth class. " $\cdot$ " is the inner product operation. " $\overline{\ }$ " is the average pooling operation, through which multiple tokens in $\mathbf{G}^L$ are pooled into a vector $\overline{\mathbf{G}^L}$. This global classification head covers all the classes from the entire training domains $\mathcal{D} = \{\mathtt{D}_1, \mathtt{D}_2, \cdots, \mathtt{D}_N\}$.

Empirically, we find that adding DG prompt brings considerable improvement (*e.g.*, in Sec.4.3, + 1.82 top-1 accuracy on CUB) over the "CNN+Transformer" baseline. Analytically, it is because the DG prompt brings additional input tokens shared by all the training domains. Intuitively, if the number of DG prompt tokens is extremely large, different input images will have almost the same representation. Under this extreme assumption, the deep representation has minimal domain bias. However, it lacks basic discriminative ability and might not be recognized properly. Therefore, the number of DS prompt tokens matters and is empirically set to 5, as illustrated in Sec.4.3.

3.3.2 Neutralizing DG Prompt

We further neutralize the DG prompt to reinforce its generalization ability. The intuition is that if the DG prompt

has no bias to any training domains $\mathcal{D}$, it can capture more general knowledge and further benefit cross-domain generalization. To this end, given a DG prompt, we measure its domain bias through the cosine similarity towards different training domain centers, as illustrated in Fig.2 (b). A domain center is the averaged feature of all the samples in the corresponding domain, which can be online approximated by momentum update:

$$\mathbf{p}^n \leftarrow \lambda \mathbf{p}^n + (1-\lambda)\frac{\sum_{i=1}^{B}\overline{f(x_i^n)}}{B}, \tag{3}$$

where the superscript n indicates the n-th training domain, λ is the momentum rate, B is the batch size, $\overline{f(x_i^n)}$ is a vector pooled from the backbone feature through the average pooling "$\overline{\ }$".

Given the DG prompt, the neutralizing loss minimizes its cosine similarity to all the domain centers, which is formulated as:

$$\mathcal{L}_N = \frac{1}{N}\sum_{n=1}^{N}(|\frac{\overline{\mathbf{G}}\cdot\mathbf{p}^n}{\|\overline{\mathbf{G}}\|\|\mathbf{p}^n\|}|). \tag{4}$$

The above neutralizing loss (Eqn.4) is performed to G^l ($l = 0, 1, \cdots, L-1$). We choose not to neutralize G^L because it conflicts with the global classification loss (which is also on G^L).

3.4. Learning Domain-Specific Feature

3.4.1 Domain-Specific Prompt

The DS prompt $\mathbf{S}^0$ is on-the-fly generated from some random backbone features $f(x)$ in the current domain through average pooling, as illustrated in Fig.2 (a). Specifically, during training, we choose C training classes, randomly sample 1 backbone feature $f(x)$ from each class, and use the average-pooled vector $\overline{f(x)}$ as a corresponding token. Consequently, $\mathbf{S}^0$ contains C tokens, *i.e.*, $\mathbf{S}^0 \in \mathbb{R}^{C\times D}$. During testing ($C$-way K-shot), we duplicate the generation procedure onto the novel testing domain, *i.e.*, using C support samples (1 from each class) to derive the DS prompt.

A side-effect of the DS prompt is that each DS token contains the underlying domain knowledge and the class information from the initialization. While the DS prompt intends to inject the domain knowledge, the class information is NOT desired. It might become a distraction (because among all the C DS tokens, $C-1$ tokens belong to different classes as the query image).

3.4.2 Local Classification Head

To suppress the undesired class information, we design an "changing identity" objective: after the DS prompt proceeds in the transformer along with the feature tokens $\mathbf{F}$, each DS token should lose its own class identity and changes the identity to the same as $\mathbf{F}$. To this end, ProD feeds the output state of the DS prompt (after average pooling), *i.e.,* $\overline{\mathbf{S}^L}$ into a local classification head. The local classifier only covers the training classes in the current n-th domain, which is formulated as:

$$\mathcal{L}_S = -\log\frac{\exp\left(\mathbf{u}_y^n\cdot\overline{\mathbf{S}^L}\right)}{\sum_j\exp\left(\mathbf{u}_j^n\cdot\overline{\mathbf{S}^L}\right)}, \tag{5}$$

where $\mathbf{u}_j^n$ enumerates all the weight vectors in the n-th local classification head, u_y^n is the weight vector for the ground-truth category of the input feature tokens $\mathbf{F}^0$, rather than any ground-truth categories for generating the GS prompt.

The reason for using the local classification head instead of a global head is: in a global classification head, $\overline{\mathbf{S}^L}$ (from a specific domain $\mathtt{D}_n$) interacts with all the weight vectors from the entire training domains $\mathcal{D} = \{\mathtt{D}_1, \mathtt{D}_2, \cdots, \mathtt{D}_N\}$. The global interaction will propagate domain knowledge from other domains ($\mathtt{D}_{j\neq n}$) to $\overline{\mathbf{S}^L}$ and thus blur its domain-specific knowledge.

The overall loss for ProD is calculated as:

$$\mathcal{L} = \mathcal{L}_G + \alpha\mathcal{L}_N + \beta\mathcal{L}_S, \tag{6}$$

where $\mathcal{L}_G$, $\mathcal{L}_N$, $\mathcal{L}_S$ are the global classification loss (Eqn.2), neutralizing loss (Eqn.4), and the local classification loss (Eqn.5), respectively. α and β are the balancing hyper-parameters.

4. Experiments

4.1. Setting

Datasets. Following the popular multi-domain training scheme, we use miniImageNet [26] and four fine-grained datasets, *i.e.*, CUB [4], Cars [18], Plantae [31] and Places [42]. We adopt the leave-on-out setting, *i.e.*, choosing one fine-grained dataset for inference and using the other three fine-grained datasets along with miniImageNet for training.

The baseline of ProD consist of CNN (ResNet-10 [13]) and a lightweight Transformer head. Without DG or DS prompt, this "CNN+Transformer" baseline achieves 72.32% 5-way 5-shot accuracy on CUB and outperforms the popular pure CNN baseline (68.98%) by +3.34%, as to be detailed in Sec.4.5. We note that: 1) a strong "CNN+Transformer" baseline only contributes a small portion to the superiority of ProD because ProD further improves the baseline by a large margin (e.g., +6.87% on CUB), and 2) adding the transformer head increases the model size (5.3M → 8.5M, as discussed in Sec.4.5) but is still efficient (smaller than ResNet-18 but achieves higher accuracy).

Inference. In the C-way K-shot testing phase, the transformer and CNN backbone are both frozen. Then, we randomly select C support samples (each from a class) and

Methods	CUB	CARS	Plantae	Places
RelationNet [28]	35.21 ± 0.46	30.12 ± 0.49	31.99 ± 0.51	49.79 ± 0.57
MatchingNet [32]	42.28 ± 0.61	28.91 ± 0.56	33.02 ± 0.56	48.53 ± 0.62
RelationNet+LFT [29]	48.10 ± 0.62	32.26 ± 0.58	35.21 ± 0.59	51.02 ± 0.56
MatchingNet+LFT [29]	43.38 ± 0.58	30.68 ± 0.59	35.10 ± 0.54	52.63 ± 0.55
RelationNet+ATA [35]	48.49 ± 0.61	31.92 ± 0.58	33.62 ± 0.49	51.00 ± 0.50
DSL [14]	50.15 ± 0.80	37.13 ± 0.69	41.17 ± 0.80	53.16 ± 0.88
Baseline	48.56 ± 0.72	33.15 ± 0.64	37.94 ± 0.71	49.81 ± 0.69
ProD	$\mathbf{53.97 \pm 0.71}$	$\mathbf{38.02 \pm 0.63}$	$\mathbf{42.86 \pm 0.59}$	$\mathbf{53.92 \pm 0.72}$

Table 1. Comparison with the state of the arts on 5-way 1-shot task.

Methods	CUB	CARS	Plantae	Places
RelationNet	51.10 ± 0.62	38.26 ± 0.58	62.99 ± 0.62	46.01 ± 0.57
MatchingNet	57.21 ± 0.63	36.98 ± 0.56	62.83 ± 0.62	43.68 ± 0.55
RelationNet+LFT	65.02 ± 0.55	43.51 ± 0.51	50.48 ± 0.46	67.34 ± 0.52
MatchingNet+LFT	61.44 ± 0.56	43.12 ± 0.52	48.49 ± 0.51	65.09 ± 0.48
RelationNet+ATA	59.42 ± 0.48	42.99 ± 0.42	45.51 ± 0.51	67.10 ± 0.41
NSAE [21]	68.17 ± 0.54	54.77 ± 0.56	59.51 ± 0.55	70.93 ± 0.54
DSL	73.57 ± 0.65	58.53 ± 0.73	62.10 ± 0.75	74.10 ± 0.72
Baseline	72.32 ± 0.77	53.17 ± 0.71	60.05 ± 0.69	69.13 ± 0.60
ProD	$\mathbf{79.19 \pm 0.59}$	$\mathbf{59.49 \pm 0.68}$	$\mathbf{65.82 \pm 0.65}$	$\mathbf{75.00 \pm 0.72}$

Table 2. Comparison with the state of the arts on 5-way 5-shot task.

use their backbone features to generate the DS prompt on the fly. Finally, the DS and DG prompt outputs are concatenated as the feature representation. Finally, we use the support features to learn a new linear classification head and then use the new head to classify the query samples, consistent with the standard few-shot classification pipeline.

Implementation details. We set the parameters for balancing the neutralizing loss and local classification loss to $\alpha = 1$ and $\beta = 1$ in Eqn.6. The transformer has only two blocks with an 8-head attention layer. Feature channels of the backbone feature and transformer layers are all 512. The DS and DG prompts both consist of 5 tokens. We train the model for 500 epochs and adopt the standard 5-way 1-shot and 5-way 5-shot test routine [7, 10, 27, 28, 32] for evaluation. More training and inference details are provided in the supplementary material.

4.2. Effectiveness of ProD

We compare the proposed ProD with the baseline and state of the art in Tab.1 (5-way 1-shot) and Tab.2 (5-way 5-shot). For a fair comparison, all the competing methods use the multi-domain training scheme, which is usually better than the single-domain counterpart. A more comprehensive comparison, including the single-domain competing methods, is provided in the supplementary. We draw two observations:

First, ProD improves the "CNN+Transformer" baseline by a large margin. For example, under the 5-way 5-shot setting, ProD increases the accuracy by $+6.87\%$, $+6.32\%$, $+5.77\%$, $+5.87\%$ on CUB, CARS, Plantae, Places, respectively. ProD only adds ten prompt tokens (5 DG tokens + 5 DS tokens) over the baseline and incurs very small computational overhead. The improvement validates the effectiveness of the proposed Prompting-to-Disentangling mechanism.

Second, ProD achieves accuracy on par with state of the art. Under the 1-shot setting, ProD surpasses the strongest competitor DSL by $+3.82\%$, $+0.89\%$, $+1.69\%$, $+0.86\%$ on CUB, CARS, Plantae, Places, respectively. Under the 5-shot setting, the superiority of ProD is even larger, *i.e.*, $+6.87\%$, $+0.96\%$, $+3.72\%$, $+0.90\%$ higher accuracy on CUB, CARS, Plantae, Places, respectively.

4.3. Ablation Study

4.3.1 DG and DS prompts

Tab.3 investigates two key components, *i.e.*, the domain-general (DG) and the domain-specific (DS) prompt through ablation on CUB. Based on the result, we draw three observations below:

First, adding the DG / DS prompt independently improves the baseline (*e.g.*, $+2.80\%$ / $+2.59\%$ 5-way 5-shot accuracy). It indicates that DG and DS knowledge are both beneficial. We note that making the DS knowledge beneficial is particularly difficult in few-shot learning because the few samples are insufficient for fine-tuning the DS knowl-

Methods	CUB	
	1-shot	5-shot
Basel.	48.56 ± 0.59	72.32 ± 0.67
Basel. + DG	51.89 ± 0.63	75.12 ± 0.69
Basel. + DS	51.48 ± 0.71	74.91 ± 0.68
Basel. + DG + DS	52.69 ± 0.66	77.63 ± 0.74
Basel. + DG + $\mathcal{L}_N$	53.08 ± 0.74	78.65 ± 0.68

Table 3. Evaluation of key components: DG prompt (DG), neutralizing loss ($\mathcal{L}_N$), and DS prompt (DS).

Methods	CUB	
	1-shot	5-shot
Basel.	48.56 ± 0.59	72.32 ± 0.67
Basel. + DS (global)	50.39 ± 0.71	73.87 ± 0.66
Basel. + DS (local)	51.48 ± 0.71	74.91 ± 0.68
ProD (global)	52.08 ± 0.74	77.65 ± 0.68
ProD (local)	53.97 ± 0.71	79.19 ± 0.63

Table 4. Comparison between the local and global classification heads on the DS prompt.

edge. Therefore, most of the prior works usually discard the DS knowledge. In contrast, ProD on-the-fly conditions the DS knowledge to the novel test domain without fine-tuning the model, therefore making the DS knowledge beneficial.

Second, comparing "Basel. + DG + DS" against "Basel. + DG (or DS)", we find that combining the DG and DS prompt brings further improvement. It indicates that the DG and conditioned DS knowledge achieve complementary benefits for the cross-domain challenge.

Third, neutralizing the DG prompt is beneficial and brings another round improvement of $+0.39\%$ and $+1.02\%$ accuracy under the 1-shot and 5-shot setting, respectively.

4.3.2 Local Classification for DS Prompt

To learn the DS prompt and the corresponding DS knowledge, ProD uses a local classification head that contains only the classes in the current domain. Tab.4 validates this choice by replacing the local head with a global one. The result shows that the local heads better cooperate with the DS prompt. For example, global head reduces the 5-way 5-shot accuracy by $+1.33\%$ (only DS prompt) / $+1.54\%$ (full ProD with DG + DS prompt). We infer that the global classification head makes each DS feature interact with class-specific prototypes (*i.e.*, the weight vectors) across all the training domains. Such cross-domain interaction brings interference and blurs the DS knowledge.

4.3.3 Choice of Inference Features

ProD provides three outputs, *i.e.*, feature tokens $\mathbf{F}^L$, DG tokens $\mathbf{G}^L$ and DS tokens $\mathbf{S}^L$. Each output is averaged into a vector through pooling layers. Tab.5 investigates how to derive the most discriminative representation with these vectors. This ablation is based on a complete ProD, different from Tab.3 where several key components are removed during training. We draw two observations below:

Inference Input	CUB	
	1-shot	5-shot
Feature Token	51.51 ± 0.72	76.13 ± 0.68
DG	53.01 ± 0.74	78.17 ± 0.61
DS	52.07 ± 0.69	77.64 ± 0.63
DG+DS	53.97 ± 0.71	79.19 ± 0.63
DG+DS+Feature Token	52.18 ± 0.75	78.04 ± 0.72

Table 5. Comparison between different features for inference with a complete ProD model.

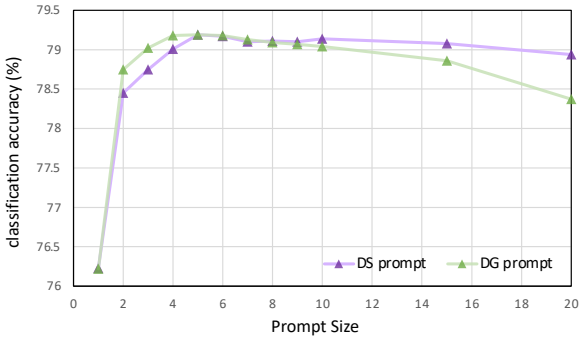

Figure 3. Evaluation of different sizes for DG and DS prompts with a complete ProD model.

First, when each type is used alone, the "DG" and "DS" feature tokens are better than the main "Feature Token". It validates that the DG and DS prompt effectively activate the DG and DS knowledge from the original backbone features and is thus superior. We also note that the main feature token in ProD is better than the main feature token in the baseline ("Basel." in Tab.4). It is because, in ProD, the DG and DS knowledge can be partially propagated to the main feature token through the attention in the transformer.

Second, comparing two combination strategies against each other, we find "DG+DS" is better. It indicates that the DG and DS prompt achieve a complementary benefit while further adding the main feature token compromises ProD. Therefore, we use "DG+DS" as the final representation.

4.4. Analysis on Hyper-parameters

4.4.1 DG and DS Prompt Size

We investigate the impact of prompt size, *i.e.*, the number of tokens in DG and DS prompt. The results are shown in Fig.3, from which we draw two observations below:

First, as the DG prompt size increases, the achieved accuracy undergoes a sharp increase and a following slow decrease. We infer that the reason is two-fold. On the one

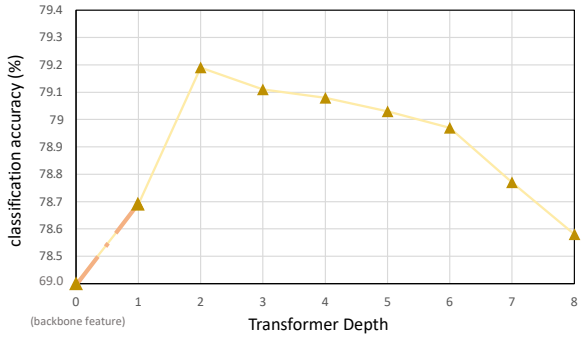

Figure 4. Analysis of the transformer depth (CUB dataset, 5-way 5-shot test). The ordinate is NOT linearly scaled for the "69.0→78.5" interval.

hand, increasing the DG prompt size enhances its capability for capturing domain-general knowledge and is thus beneficial. However, on the other hand, an oversized DG prompt might suppress the difference between samples (because all the samples share the same DG prompt), thus decreasing the discriminative ability. Therefore, we set the DG prompt size to 5, which achieves 79.19% 5-way 5-shot accuracy on CUB, for all the datasets.

Second, increasing the DS prompt size brings a similar trend of "increasing → slightly decreasing" the accuracy. In the supplementary material, we further investigate DS prompt size under the 10-way 5-shot setting. The results show that $5 \sim 10$ DS tokens achieve close results, indicating that the DS prompt size does not have to increase along with C when C is large.

4.4.2 Transformer Depth

We analyze the impact of transformer depth on CUB under the 5-way 5-shot setting. All the models are the full ProD model except for the "0 layer backbone feature". The results are shown in Fig.4. We draw two observations below:

First, when the transformer has only one block, it already significantly improves +9.70% over the CNN backbone feature (68.98%→78.68%). We note that: 1) the additional computation cost within a 1-block transformer is minimal (less than 1.6M), and 2) adding a transformer without our prompting-to-disentangling mechanism only brings slight improvement. Combining these two facts, we infer that the improvement is mainly contributed by our prompting-to-disentangling mechanism rather than the transformer itself.

Second, when the transformer depth increases, the accuracy increases to its peak of 79.19% and gradually decreases. We infer that the reason is two-fold. On the one hand, two transformer blocks are already sufficient for depicting the required prompting-to-disentangling effect. On the other hand, training a large transformer generally requires a large-scale dataset [8], while the small-scale training data in few-shot learning is insufficient. Therefore, we set the transformer depth to 2 as the optimized result.

Method	Size	CUB 5-shot
Res10	5.3M	68.98 ± 0.81
Res18	11.7M	72.39 ± 0.84
Basel. (Res10 + Trans)	8.5M	72.32 ± 0.77
ProD (Res10 + Trans + Prompt)	8.6M	79.19 ± 0.59

Table 6. Analysis of the computational efficiency. "Res10", "Res18" and "Trans" denote ResNet-10, ResNet-18 and the transformer head, respectively.

4.5. Computational Efficiency

Tab.6 analyzes the computational efficiency by comparing the model size and the achieved accuracy. Comparing "Basel. (Res10 + Trans)" against "Res10", we observe that the transformer head increases 3.2M parameters and brings +3.34% accuracy improvement (68.98% → 72.32%). This improvement is due to the inherent capability of the transformer. Based on the baseline, ProD further brings +6.87% accuracy improvement while adding only about 0.1M parameters. It indicates that the prompting-to-disentangling mechanism is the major reason for the superiority of ProD and is very efficient. Moreover, when compared with the larger pure CNN model (ResNet-18), ProD (based on ResNet-10) is still more accurate while being smaller.

5. Limitation

A limitation is that we have not investigated the proposed ProD on a pure-transformer network (*e.g.*, ViT [8]). There are two reasons: 1) prior works are based on the CNN backbone and are not entirely comparable with a pure-transformer method, and 2) training a ViT model requires a large-scale dataset, which is not satisfied in the few-shot learning task. In the feature, we will consider developing an efficient pure-transformer for few-shot learning.

6. Conclusion

This paper proposes a prompting-to-disentangling (ProD) method for cross-domain few-shot image classification. ProD uses two parallel prompts to disentangle the domain-general and domain-specific knowledge from a single backbone feature. While the domain-specific knowledge in prior works is usually bound to the already-seen training domain and is thus harmful, the domain-specific knowledge in ProD can be conditioned to the novel test domain through the on-the-fly DS prompt. Therefore, ProD benefits from both domain-general and domain-specific knowledge and significantly improves the baseline. Moreover, the achieved results are on par with state of the art.

References

[1] Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games, 2020. 3

[2] Rahaf Aljundi and Tinne Tuytelaars. Lightweight unsupervised domain adaptation by convolutional filter reconstruction, 2016. 3

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2019. 3

[4] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, pages 438–451. Springer, 2010. 5

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 3

[6] Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *CoRR*, abs/1206.4683, 2012. 3

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 1, 2, 6

[8] 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, 2020. 8

[9] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization, 2021. 3

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. 1, 2, 6

[11] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2551–2559, 2015. 3

[12] Yunhui Guo, Noel Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogério Feris. A broader study of cross-domain few-shot learning. In *ECCV (27)*, pages 124–141, 2020. 1, 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 5

[14] Zhengdong Hu, Yifan Sun, and Yi Yang. Switch to generalize: Domain-switch learning for cross-domain few-shot classification. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 6

[15] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3635–3649. Curran Associates, Inc., 2021. 3

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 2, 3

[17] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 158–171, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 3

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[19] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018. 2, 3

[20] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Cross-domain few-shot learning with task-specific adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7161–7170, 2022. 3

[21] Hanwen Liang, Qiong Zhang, Peng Dai, and Juwei Lu. Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9404–9414, 2021. 1, 3, 6

[22] Bingyu Liu, Zhen Zhao, Zhenpeng Li, Jianan Jiang, Yuhong Guo, and Jieping Ye. Feature transformation ensemble model with batch spectral regularization for cross-domain few-shot classification, 2020. 2

[23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. 3

[24] Yahao Liu, Jinhong Deng, Jiale Tao, Tong Chu, Lixin Duan, and Wen Li. Undoing the damage of label shift for cross-domain semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7042–7052, 2022. 3

[25] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8505–8514, October 2021. 3

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2, 6

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 2, 6

[29] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation, 2020. 1, 2, 6

[30] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):41–57, jun 2016. 3

[31] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 5

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 1, 2, 6

[33] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018. 3

[34] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation, 2019. 1, 2

[35] Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation, 2021. 6

[36] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization, 2021. 3

[37] Jingge Wang, Yang Li, Liyan Xie, and Yao Xie. Class-conditioned domain generalization via wasserstein distributional robust optimization, 2021. 3

[38] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. 2

[39] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation, 2021. 2

[40] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844, 2021. 3

[41] An Zhao, Mingyu Ding, Zhiwu Lu, Tao Xiang, Yulei Niu, Jiechao Guan, and Ji-Rong Wen. Domain-adaptive few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1390–1399, January 2021. 1

[42] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 5

[43] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2