CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Adaptive and Generalizable Vision-Language Models

Master's thesis in Computer science and engineering

Zhixing Li

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Adaptive and Generalizable Vision-Language Models

Zhixing Li

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Adaptive and Generalizable Vision-Language Models

Zhixing Li

Supervisor: Yinan Yu, Department of Computer Science and Engineering
Co-supervisor: Arsham Gholamzadeh Khoee, Department of Computer Science and Engineering
Examiner: Kivanc Tatar, Department of Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Adaptive and Generalizable Vision-Language Models

Zhixing Li
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Domain generalization remains a significant challenge for vision-language models, as they are required to perform reliably on previously unseen domains during inference. In this work, we introduce a domain prompt fusion framework aimed at improving the generalization capability of CLIP-based models under domain shift. Our approach integrates three core components: a dual-part soft prompt (comprising domain-agnostic and domain-specific prompts), a domain feature extractor, and a prompt fusion mechanism. The extractor generates domain representations from input images and computes source-domain prototypes, which guide the fusion of prompt-based text features. By weighting and combining domain-aware text features according to their similarity to the input images domain representation, the model achieves improved alignment between visual and textual modalities.

We evaluate the proposed method on two widely-used benchmarks: Office-Home and mini-DomainNet. The results demonstrate consistent performance gains over standard zero-shot CLIP and CoOp. Specifically, our method achieves average accuracies of 84.98% and 85.53% on Office-Home and mini-DomainNet, respectively. Extensive ablation studies and visualizations further validate the effectiveness of our design. While a small performance gap remains compared to the current state-of-the-art method DDSPL, our analysis identifies key areas for future enhancement, including prompt design refinement, class-dependent fusion strategies, and the use of latent domains in place of manual annotations.

Keywords: Vision-language model, prompt learning, domain generalization, prompts ensembling.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Yinan Yu, as well as Arsham Gholamzadeh Khoee, for their thoughtful and meticulous guidance throughout the course of my thesis work. They helped me define the direction of my research, pointed out the weaknesses in my proposed methods, assisted with the experimental design, and provided invaluable feedback on my writing. I was deeply impressed by their expertise and professionalism, and I will continue to learn from them as role models in my academic development.

Secondly, I would like to thank my examiner, Kivanc Tatar, who offered many insightful comments on both my planning report and halftime report. His feedback helped me improve the structure and academic rigor of my thesis. I am also grateful to my opponents, Filip Landin and Luca Modica, who provided valuable suggestions from a fellow students perspective. Their input helped me refine the details of my thesis and made it more understandable and coherent.

Finally, I would like to thank the C3SE division at Chalmers University for providing the computational resources that supported this work. We acknowledge the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking and hosted by CSC (Finland) and the LUMI consortium.

Zhixing Li, Gothenburg, 2025-06-08

Contents

List of Figures

List of Tables

1

Introduction

In the field of computer vision, if we examine the problem from the perspective of *how to train models*, we can observe that the development of training methodologies has generally progressed through three major stages. In the early stage, traditional training methods require collecting large-scale training data and labeling them for specific tasks [1], which is time-consuming and labor-intensive. Additionally, these methods typically suffered from slow convergence, often requiring a substantial number of training epochs to achieve satisfactory performance.

In the second stage, with the emergence of the *Pre-training, Fine-tuning, and Prediction* approach, it became unnecessary to train models from scratch. Instead, a pretrained model could be fine-tuned on the downstream task dataset [2]. Compared to training from scratch, this method offers faster convergence, requires less training data, and can even achieve superior performance. However, this approach still relies on a certain amount of labeled data from the downstream task. Moreover, the fine-tuned model often suffers from catastrophic forgetting [3], a phenomenon in which the model loses the knowledge acquired during pre-training, resulting in significantly reduced generalization ability and poor transferability to other downstream tasks.

Recently, with the rapid advancement of language models in the field of natural language processing (NLP), traditional vision models have begun to integrate with language models, giving rise to a new class of models known as vision-language models [4]. These models are pre-trained on large-scale image-text pairs and can be directly applied to downstream tasks without the need for additional fine-tuning. Vision-language models represent a groundbreaking intersection of computer vision and natural language processing, which aim to combine both visual and linguistic information, enabling machines to understand and reason about the world in a manner that closely mimics human cognitive processes.

1.1 Research Topic and Motivations

The potential for adaptability and generalizability in vision-language models is particularly exciting. Current vision models, while powerful, often struggle when faced with scenarios that deviate significantly from their training data [5]. This limitation hinders their practical application and scalability. Especially in increasingly practical application scenarios such as autonomous driving and intelligent robotics, models must handle complex, unpredictable real-world environments where it is nearly im-

possible to cover all possibilities in the training data. This necessitates models with excellent generalization capabilities, capable of handling unseen scenarios effectively based on the training data.

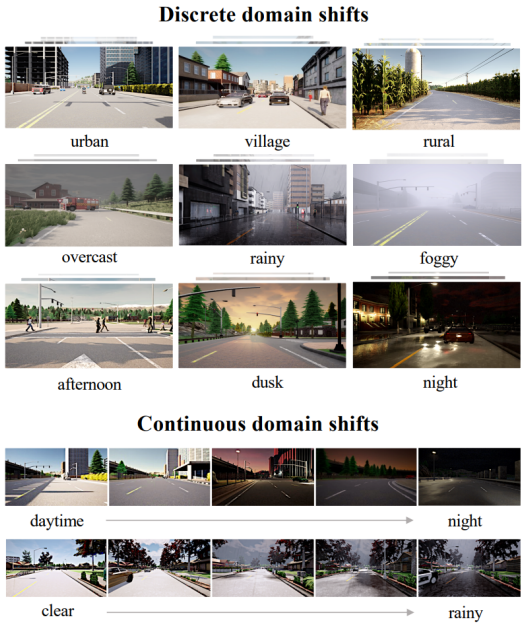


Figure 1.1: Common examples of domain shift in autonomous driving [6]. Variations in architectural styles, weather conditions, and lighting are typical scenarios where domain shift occurs.

Vision-language models, represented by CLIP [1], with their strong generalization ability, present a highly promising solution to these challenges. Therefore, this study focuses on exploring ways to further improve the generalization ability of vision-language models, overcoming challenges such as domain shift (as shown in Figure 1.1), and enabling the models to perform better in highly complex and dynamic real-world environments.

1.2 Goals and Challenges

This research has two objectives. First, we seek to advance the theoretical understanding of how semantic information and prompt learning can be effectively integrated into vision-language models to improve generalization. Second, we aim to develop practical techniques and architectures that implement these insights, resulting in models that demonstrate superior robustness and performance across diverse tasks and domains.

We expect our research to yield the following outcomes:

1. Improved cross-domain performance, demonstrating enhanced generalization.
2. Effective integration of prompt learning for improved Out-of-Distribution (OOD) generalization.

3. Better adaptability to new, unseen tasks and domains.
4. More robust representations from both modalities that capture deeper semantic relationships.

However, as described earlier, achieving these goals is highly challenging. For example, we may need to address the problem of domain shift, which typically refers to the situation where the training and testing data come from different domains, resulting in distributional discrepancies. Traditional vision models perform poorly in such scenarios, showing little to no generalization across different domains. Although CLIP has significantly improved this issue by raising the model's performance on domain-shifted data to an acceptable level [1], it still falls short of the ideal. One contributing factor is that CLIP relies on manually designed hard prompts, which are not only difficult to optimize but also lack flexibility. Prompt Learning (see Section 2.2 for a detailed definition), a technique that enables the automatic optimization of soft prompts, has been shown to effectively improve the generalization ability of vision-language models [7]. We want to further enhance vision-language models, especially using prompt learning, enabling them to adapt to a wider range of domains while minimizing the gap between their performance on domain-shifted data and the ideal scenario.

1.3 Research Questions

We aim to further enhance the generalization ability of vision-language models through prompt learning techniques. This is a lightweight adaptation strategy that allows the pretrained model to achieve better performance on downstream tasks by optimizing the prompt, without the need to modify the models parameters.

Although significant work has been done in this area (see Section 2.1), existing methods still have limitations in addressing the **Multi-Source Domain Generalization (MSDG)** [8] problem. Specifically, we consider a generalization problem with L source domains $S_l = \{(x_i^l, y_i^l)\}_{i=1}^{n_l}$, each associated with a joint distribution P_{XY}^l. Note that $P_{XY}^l \neq P_{XY}^{l'}, \forall l, l' \in \{1, \cdots, L\}$ and $l \neq l'$. The goal of the MSDG problem is to learn a predictive function $f : \mathcal{X} \to \mathcal{Y}$ using source domain data such that it minimizes the prediction error on an unseen target domain $S_{target}, P_{XY}^{target} \neq P_{XY}^l, \forall l \in \{1, \cdots, L\}$:

$$\min_f \mathbb{E}[\mathcal{L}(f(x^{target}), y^{target})], \tag{1.1}$$

where $\mathcal{L}(\cdot, \cdot)$ is the loss function.

We assume that we have access to sufficient labeled data from the source domains. However, we do not make any assumptions about the target domain, meaning it could be a completely unseen domain or an arbitrary combination of multiple different domains. Naturally, we also do not have access to any training data from the target domain. To simplify the problem, we also assume that the target domain and source domains share the same label set.

The research questions of this study are as follows:

1. How can we use soft prompt learning to improve the generalization capability of CLIP, enabling it to perform better in the context of the MSDG problem?
2. Can our domain-feature-guided prompt fusion mechanism improve modality alignment between image and text features, compared to existing domain generalization methods?
3. What factors contribute to the strengths and limitations of our method in achieving domain generalization, as revealed through comparative and ablation studies?

1.4 Limitations and Risks

Due to time constraints, our main focus is on text prompt learning rather than multimodal prompt learning. The former is simpler but overlooks adjustments to the visual branch. By designing a more effective visual prompt or establishing a better interaction mechanism between the text and visual branches, prompts could potentially generalize better.

Moreover, we assume that the source domain and target domain share the same label set. However, in real-world scenarios, their label sets are not always fully overlapping, and the target domain often introduces unknown labels. This makes the problem more complex. Due to time constraints, we do not consider this case for now, leaving it for future improvements.

Finally, given that our focus is on the domain generalization problem, we do not conduct additional evaluations of the model on standard image classification tasks. This is primarily because domain shift is absent in such settings, rendering most of the components in our proposed framework ineffective.

In this project, we used publicly available datasets and code developed by others. The dataset authors have granted permission for unrestricted use of the data for non-commercial academic research purposes. The code is released under the MIT License, which allows free usage within the licenses terms and conditions. The datasets do not contain any sensitive information, and as such, we believe their use poses minimal privacy risks. For a more detailed discussion of ethic topics, please refer to Chapter 8.

1.5 Thesis Outline

This thesis is organized into 8 chapters.

In Chapter 1, we introduce the project background, research motivation, objectives, questions, and limitations.

In Chapter 2, we present the main findings from our literature review, covering commonly used approaches for addressing the domain generalization problem, as

well as techniques related to prompt learning, feature adapters, ensemble learning, and established benchmarks.

In Chapter 3, we provide a detailed explanation of the theoretical foundations of our work, specifically the operational principles of vision-language models and the design philosophy of prompt learning.

In Chapter 4, we introduce recent methods that leverage prompt learning techniques to improve the generalization ability of vision-language models. We also provide a brief comparison of the similarities and differences among these approaches.

In Chapter 5, we elaborate on our proposed method, detailing the overall framework, the design rationale for each module, and the associated theories.

In Chapter 6, we present the evaluation results of our method, including the datasets used, baseline comparisons, implementation details, and testing metrics, along with a comprehensive analysis of the outcomes that highlights both strengths and weaknesses.

In Chapter 7, we summarize the key contributions and findings of our work.

In Chapter 8, we discuss ethical topics related to our project.

2

Background

There is still a desire to further improve the generalization ability of vision language models, especially in cases requiring a certain level of prior knowledge. The spectrum of typical methods and the number of parameters that need to be adjusted are shown in Figure 2.1. The simplest and most straightforward approach is to fine-tune the entire pre-trained model or a few layers of it. Theoretically, this method can achieve better performance on specific tasks. However, since pre-trained models are usually quite large, fine-tuning the entire model is time-consuming and demands significant computational resources. Moreover, during the fine-tuning process, the model often loses some of the knowledge learned during pre-training, leading to a decline in its generalization ability. This issue is known as catastrophic forgetting.

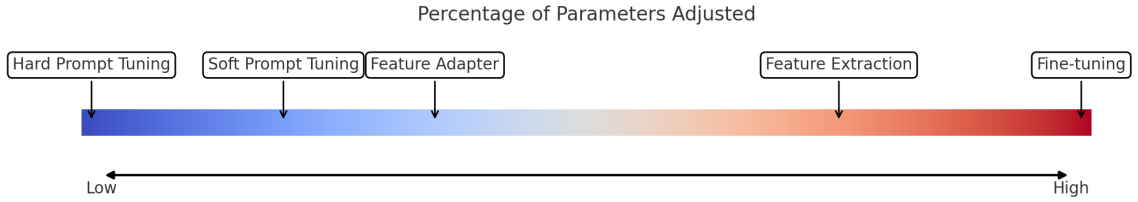


Figure 2.1: The spectrum of the number of parameters modified by different methods to adapt pre-trained models to downstream tasks.

To address these issues, prompt learning [9] offers a promising solution. This approach avoids fine-tuning the pre-trained model. Instead, it learns an optimal prompt, enabling the pre-trained model to better adapt to the downstream task. Alternatively, we can attach a lightweight network to the pre-trained model to modify the features extracted by it. This approach is known as feature adapter [10].

2.1 Domain Generalization

Traditional machine learning and deep learning theories generally assume that source and target data are independently and identically distributed. However, this assumption is often difficult to satisfy in real-world applications. To ensure that algorithms or models achieve stable and accurate results in practice, these methods must be capable of handling out-of-distribution (OOD) data and even counteracting domain shift. Domain shift refers to the discrepancy between the distribution of training data and that of test data [11]. The objective of the domain generalization (DG)

problem [12] is to address domain shift, and the specific definition is provided in Section 1.3.

Another problem related to domain shift is domain adaptation (DA) [13], which assumes access to (unlabeled) training data from the target domain. Compared to DG, DA makes a stronger assumption about the availability of target domain information. For example, in the context of autonomous driving, it is challenging to include every combination of road traffic environments from all cities, weather conditions, time periods, and potential traffic hazard types in a dataset. This implies that some scenarios are inevitably not represented in the collected data. Consequently, DG places even higher demands on algorithms, requiring strong generalization capabilities to properly or reasonably handle unseen scenarios or conditions.

Numerous approaches have been proposed to address the DG problem. Domain alignment is perhaps the most extensively studied method, the core idea of which is to minimize the discrepancies in data distributions across source domains for learning domain-invariant representations [14], like Figure 2.2. Typically, these methods employ a metric to quantify the differences between distributions, such as moments [15], KL divergence [16], or Maximum Mean Discrepancy (MMD) [17], among others. Alternatively, contrastive learning [18] or adversarial learning [19] techniques are used to promote the extraction of domain-invariant features.

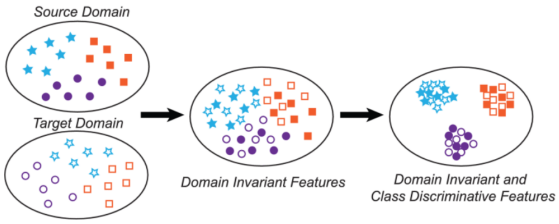


Figure 2.2: Domain invariant features. [17]

Meta-learning is another widely used strategy. A typical example is Model-Agnostic Meta-Learning (MAML) [20], which further divides the training data into meta-train and meta-test sets, training on the meta-train set to enhance performance on the meta-test set. However, these methods generally aim to learn an improved initialization, so that only a few additional training rounds on the target task are needed to achieve satisfactory performance.

In addition, data augmentation is a widely used technique for enhancing the generalization capabilities of machine learning and deep learning models. Its core idea is straightforward: by applying transformations such as scaling, cropping, rotating, and altering color [21], or by mixing image features [22], it is possible to simulate the effects of domain shift to a certain extent, thereby enabling the model to learn how to extract more robust features. Beyond manually designed augmentation strategies, some researchers have explored using neural networks to learn effective data augmentation patterns that specifically improve the model's generalization performance [23].

Except for these three common approaches mentioned above, numerous other meth-

ods can be employed to address the DG problem, such as ensemble learning, learning disentangled representations, regularization techniques, and reinforcement learning [14]. However, most current approaches are confined to traditional vision models, such as CNNs [24] or ViTs [25], and there is relatively little research on domain generalization in vision-language models. Our research is dedicated to bridging this gap.

2.2 Prompt Learning

In summary, prompt learning primarily focuses on three directions: text prompt learning, visual prompt learning, and multimodal prompt learning. Unlike discrete prompt engineering in natural language, text prompt learning optimizes prompts in the continuous word embedding space, which may not correspond to any actual natural language text. These soft prompts may encode information that is more effective for the model than natural language text, allowing them to better guide the model in performing downstream tasks.

For instance, CoOp [7] embeds a category name into a string like "$[V]_1, [V]_2, ..., [V]_M$, [CLASS]" where each $[V]$ represents a word embedding. By minimizing the classification loss on the downstream task, an optimal prompt representation (shared or distinct) for each category can be learned. CoCoOp [26] extends this idea further by finding an optimal prompt for each image to better describe its content. DAPrompt [27] proposes encoding domain information into the text prompt to effectively enhance the model's domain generalization ability. They achieve this by learning a domain-agnostic prompt to capture domain-independent information and training domain-specific prompts for each domain.

There are two main implementations of visual prompts. The first approach, similar to text prompts, adds a set of learnable parameters as prompts in the input layer of a Vision Transformer, such as VPT [28]. The second approach introduces perturbations to the input image to serve as a prompt, these perturbations serve as visual cues to guide the model in extracting more informative features [29]. Multimodal prompt learning combines both text and visual prompt learning approaches, like MaPLe [30]. Currently, an increasing number of studies are attempting to leverage multimodal information for complementary advantages and mutual enhancement to improve the generalization ability of vision-language models.

Although prompt learning has been proven to effectively enhance the generalization ability of vision-language models and improve their performance on specific downstream tasks, it also has certain limitations. For instance, while it avoids fine-tuning the pre-trained model, the training cost of text prompt learning may remain high, especially when generating descriptions tailored to specific images [26]. Moreover, the performance improvement on downstream tasks often comes at the expense of reduced generalization ability on other tasks [7]. Whether it is worth sacrificing the knowledge already learned by the pre-trained model for the sake of performance on a specific task requires careful consideration and case-by-case analysis. Finally, these learned prompt representations are often difficult to interpret. We do not have

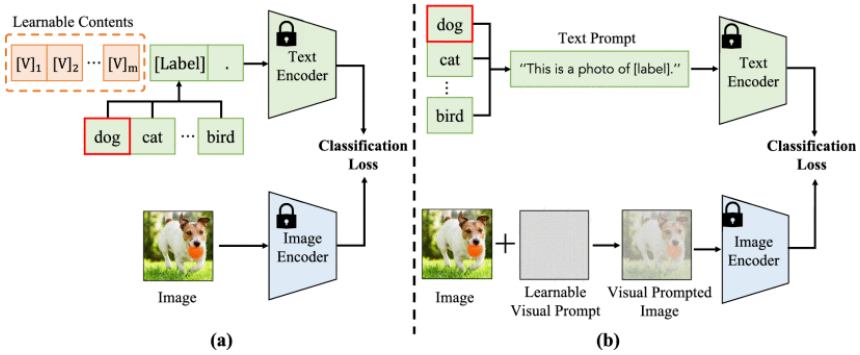


Figure 2.3: Illustration of text prompt learning [7] in (a) and visual prompt learning [29] in (b). [4]

a clear understanding of the circumstances under which these representations are applicable or when they may fail.

2.3 Feature Adapter

Unlike prompt learning, feature adapter does not improve generalization ability by modifying the input to the pre-trained model. Instead, it focuses on modifying the features extracted by the pre-trained model, enabling the model to capture more effective and task-specific features. For example, CLIP-Adapter [31] adds a lightweight MLP network after the original image encoder and text encoder to further extract features, which are then combined with the original features through residual connections.

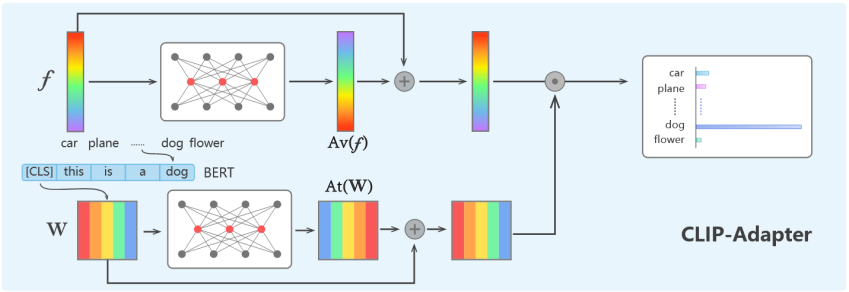


Figure 2.4: Illustration of CLIP-Adapter [31]. f is the image feature, and W is the text features.

This method is simpler to implement and train compared to text prompt learning, while the residual fusion ensures that the model can better balance the knowledge already learned by the pre-trained model with the new knowledge acquired for specific downstream tasks. Of course, one drawback is that feature adapters typically require more trainable parameters than prompt learning. Additionally, they involve more hyperparameters, which often makes the tuning process more complex.

2.4 Ensemble Learning

Ensemble learning typically refers to the simultaneous training of multiple copies of the same model, with each copy trained on different subsets of the training data to ensure diversity [32]. During inference, the ensemble aggregates the outputs of the individual sub-models to obtain a more accurate prediction. This design results in predictions that are more robust than those of a single sub-model, demonstrating a stronger ability to resist noise and perturbations, and has proven effective in addressing the DG problem.

A typical strategy for enhancing model generalization is to train a separate backbone or classifier head for each source domain. For instance, [33] proposed Domain Adaptive Ensemble Learning (DAEL), which comprises a CNN feature extractor shared across all source domains alongside individual classifier heads for each domain. By coordinating the outputs of these domain-specific classifier heads, the model can effectively handle input images from various domains.

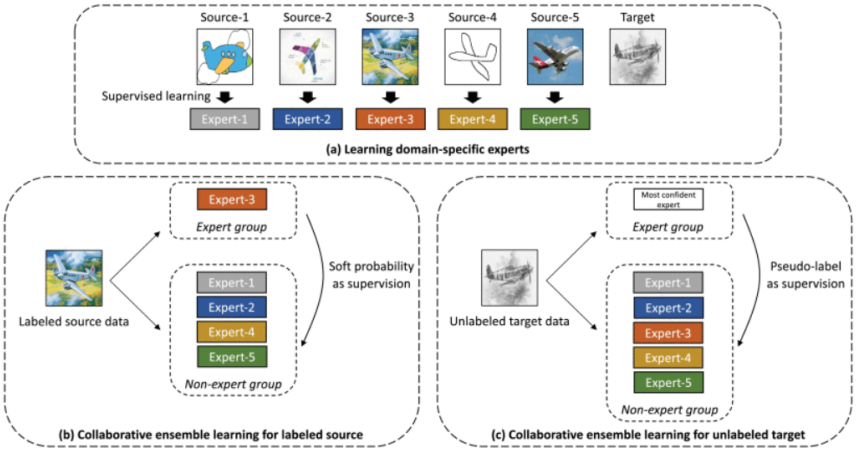


Figure 2.5: Domain Adaptive Ensemble Learning. [33]

Another strategy avoids explicitly training multiple sub-models and instead aggregates weights from different training stages of a single model. Since the model tends to focus on different features at various stages of training, integrating these weights not only improves generalization but also significantly reduces the time and space overhead compared to training several models. In [34], the authors introduced PromptSRC, a method that combines prompt learning with ensemble learning by performing a weighted fusion of soft prompts from different training stages, where the weights are sampled from a Gaussian distribution. They argue that self-ensemble of soft prompts enables the integration of useful knowledge acquired at various stages, thereby effectively enhancing model generalization.

Although ensemble learning methods have demonstrated considerable potential in enhancing model generalization, they also come with certain drawbacks. Training multiple models undoubtedly reduces training and inference efficiency, introduces additional parameters, and increases storage requirements. Moreover, designing appropriate ensemble weights is crucial for effective integration; if the weights are

not properly calibrated, they may not only fail to improve generalization but could also degrade model performance.

2.5 Benchmarks

Currently, there are numerous datasets available for evaluating the performance of vision-language models. We focus on using vision-language models for image classification tasks. General-purpose datasets such as ImageNet [35] and CIFAR-10 [36] can be used to evaluate a model's performance on general tasks. Additionally, task-specific datasets like Stanford Cars [37] and EuroSAT [38], which may require certain prior knowledge, can be used to assess the model's performance on specific tasks. The most commonly used metric is accuracy and its variants, such as the arithmetic mean or harmonic mean accuracy across different categories or domains.

To evaluate the generalization ability of a model, the most common approach is zero-shot learning [39], where the model is applied directly to a new task without any fine-tuning. Additionally, linear probing [1] can be used to assess the feature extraction capability of the model. In this method, the backbone network is frozen, and a linear classifier is trained on a new dataset. The classification performance is then used to evaluate the effectiveness of the extracted features. Currently, the generalization ability of models is often evaluated by directly applying the trained model to new datasets. However, there are also datasets specifically designed to test a model's ability to handle domain shift, such as Office-Home [40] and DomainNet [41]. Accuracy is typically used as the evaluation metric in these cases.

3

Theory

In this chapter, we introduce two key theoretical foundations of our work. First, we provide a detailed explanation of the operational principles of vision-language models; subsequently, we present the design philosophy behind prompt learning techniques.

3.1 Vision-Language Models

In simple terms, a vision-language model is a type of pre-trained vision model designed to better address zero-shot prediction problems in visual recognition tasks by learning associations between images and text. Typically, this neural network consists of two components: an image encoder responsible for extracting image features and a text encoder for extracting text features, as shown in Figure 3.1. Given an image-text pair as input, the model solves visual recognition tasks by performing some form of matching between the image features and text features. As a pre-trained model, the vision-language model can be applied to various downstream tasks, including image classification, semantic segmentation, object detection, and image/text generation, among others.

For the image encoder, models based on convolutional neural networks, such as ResNet[24] and ResNet-D[42], are commonly used. Alternatively, transformer-based models like Vision Transformer (ViT) [25] are also popular choices. For the text encoder, Transformer [43] and their variants remain the primary models in use.

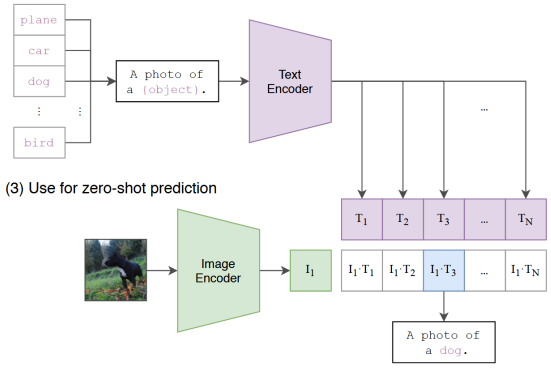


Figure 3.1: Vision-language model architecture. [1]

3.1.1 Text Encoder

In this section, we briefly introduce the Transformer architecture used as the text encoder. As shown in the Figure 3.2, the architecture consists of two main components: an encoder and a decoder, each composed of multiple stacked Transformer blocks. The encoder transforms the input text sequence into a sequence of embedding vectors, while the decoder reconstructs the text sequence based on these embeddings. Each Transformer block contains several key components, including a multi-head self-attention layer, a fully connected feed-forward network, residual connections, and layer normalization.

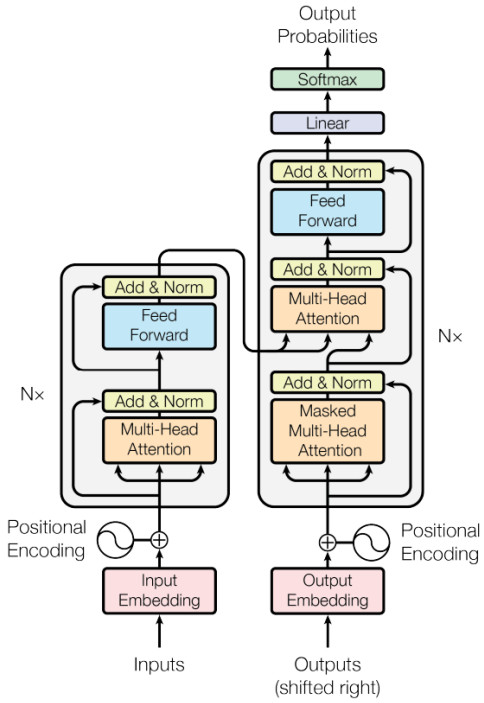


Figure 3.2: Transformer architecture. [43]

The attention mechanism is the core component of the Transformer architecture. Its fundamental principle is illustrated in the Figure 3.3. In simple terms, it involves computing a similarity score between a query vector $\mathbf{q}$ and a key vector $\mathbf{k}$, and then using this score to perform a weighted aggregation of the corresponding value vectors $\mathbf{v}$. The matrix formulation is given as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_k}})V. \tag{3.1}$$

To enable the model to capture different aspects of the same sequence, the self-attention mechanism is replicated multiple times, resulting in the multi-head attention mechanism. In simple terms, the Q, K, V matrices are first linearly projected into multiple subspaces. Within each subspace, attention is computed independently, and the outputs are then concatenated. This approach enhances the models

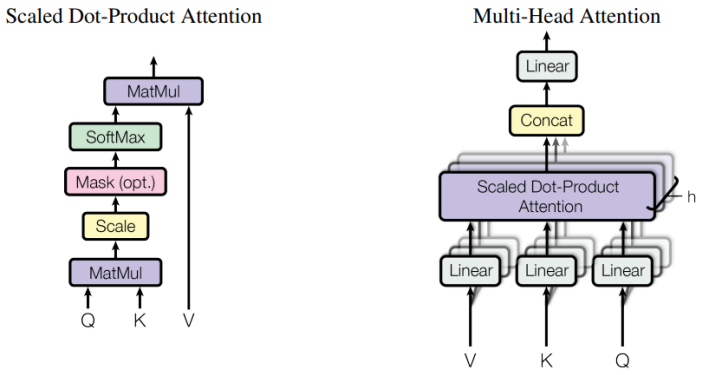


Figure 3.3: Attention mechanism. [43]

representational capacity by allowing it to attend to diverse semantic information simultaneously. The formula is as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \cdots, \text{head}_h)W^O \\ \text{where} \quad \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \end{aligned} \tag{3.2}$$

Here, h is the number of head, W is the projection matrix, $W^O \in \mathbb{R}^{d_{model} \times d_{model}}, W_i^Q \in \mathbb{R}^{d_{model} \times d_q}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, where d_{model} is the embedding dimension, and $d_q = d_k = d_v = \frac{d_{model}}{h}$.

In addition to the self-attention layer, another important component within each Transformer block is the feed-forward network, which further transforms the representation of each token individually. It consists of two fully connected layers with a ReLU activation function in between:

$$\text{FFN}(X) = W_2\sigma(W_1X), \tag{3.3}$$

where W_1, W_2 are two parameter matrix, and $\sigma(\cdot)$ represents the activation function. Furthermore, both the self-attention layer and the feed-forward network are followed by residual connections and layer normalization [44] to stabilize training and facilitate gradient flow.

It is worth noting that the Transformer architecture does not contain any recurrent structures; all input tokens are processed in parallel. To incorporate positional information from the sequence, positional encodings are added to the embedding of each token:

$$\text{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right), \tag{3.4}$$

$$\text{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right), \tag{3.5}$$

where pos is the position and i is the dimension. These encodings enable the Transformer to learn both the absolute positions of individual tokens and the relative positional relationships between them.

In CLIP, the authors employed a 12-layer Transformer model with a hidden size of 512 and 8 attention heads. The input text sequence is enclosed with special tokens: [SOS] at the beginning and [EOS] at the end. The textual representation is extracted from the [EOS] token at the final layer of the Transformer, followed by layer normalization and a linear projection into the multimodal embedding space. [1]

3.1.2 Image Encoder

In this section, we introduce the Vision Transformer (ViT) model, which is commonly used as the image encoder. Although CNN-based visual models are also widely used, their performance is generally inferior to that of ViT [1]. Due to time constraints, this study focuses exclusively on vision-language models based on ViT.

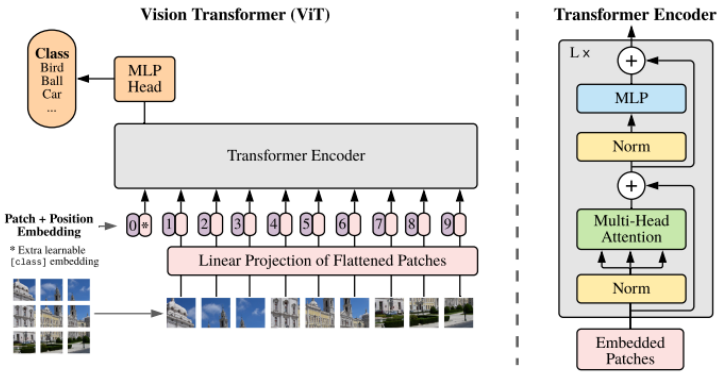


Figure 3.4: Vision transformer. [25]

The architecture of ViT is illustrated in the Figure 3.4. Overall, it closely follows the structure of the standard Transformer model [43]. However, ViT utilizes only the encoder part of the Transformer to extract image features and employs an MLP network to perform the classification task.

Since the Transformer can only process 1D sequence inputs, the authors first divide the original 2D image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a series of image patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 C)}$, where (H, W) is the resolution of the original image, C is the number of channels, and (P, P) is the resolution of the image patch. The number of resulting patches is given by $N = \frac{HW}{P^2}$. After patching, each 2D image patch is further transformed into a 1D vector (embedding) via a linear projection, making it suitable for processing by the Transformer.

To obtain a representation for each image, or equivalently the image feature, the authors adopt a method similar to that used in BERT [45]. Specifically, a special [CLS] token is prepended to the sequence of image patch embeddings to learn a holistic representation of the image. The final image feature is then extracted from the output corresponding to the [CLS] token at the last layer of the Transformer. As in the original Transformer, 1D positional encodings are added to each embedding vector to incorporate the sequential information among image patches.

The authors point out that a significant distinction between ViT and CNN-based models lies in the amount of inductive bias. In CNNs, inductive priors such as locality, two-dimensional spatial structure, and translation or rotation invariance are inherently embedded in the model through the use of convolutional kernels. In contrast, ViT exhibits much less such bias. This is because the attention mechanism operates globally, with limited emphasis on local spatial relationships, and the positional encodings do not impose any explicit assumptions about spatial structure. As a result, all knowledge about spatial relationships must be learned from scratch during training. This reduced inductive bias is considered a key characteristic of ViT and may partly explain its superior performance compared to traditional CNN models.

In the CLIP model, the authors adopted an implementation that is nearly identical to the one provided in the original ViT paper [1]. After extracting image features using ViT, no further processing is applied, meaning that the image feature space directly serves as CLIPs multimodal embedding space. Aligning text features and image features within the same feature space is essential to ensure the meaningfulness of subsequent similarity comparisons.

3.1.3 Modality Alignment

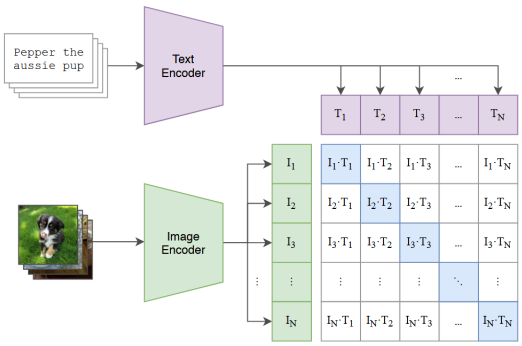


Figure 3.5: Contrastive learning [1]

The design of the objective function is also a key aspect. Contrastive objectives [46] are among the most commonly used objectives for vision-language models, as exemplified by CLIP [1]. The basic idea is to maximize the cosine similarity between the features of true image-text pairs while minimizing the cosine similarity between all other combinations of features, as shown in Figure 3.5. More precisely, in CLIP, the contrastive loss between image (I) and text (T) consists of two components: image-to-text contrastive loss $\mathcal{L}_{I \to T}$ and text-to-image contrastive loss $\mathcal{L}_{T \to I}$. The formula is as follows [47]:

$$\mathcal{L}_{T \to I} = -\sum_{i=1}^{B} \log \frac{\exp\left(z_i^T \cdot z_k^I / \tau\right)}{\sum_{j=1}^{B} \exp\left(z_i^T \cdot z_j^I / \tau\right)}, \tag{3.6}$$

$$\mathcal{L}_{I\to T} = -\sum_{i=1}^{B} \log \frac{\exp\left(z_i^I \cdot z_k^T/\tau\right)}{\sum_{j=1}^{B} \exp\left(z_i^I \cdot z_j^T/\tau\right)}. \tag{3.7}$$

Here, B represents the batch size, τ is the temperature, and z denotes the image or text features. Finally, the total loss is $\mathcal{L} = \mathcal{L}_{T\to I} + \mathcal{L}_{I\to T}$.

It is important to note that CLIP employs a cosine similarity-based InfoNCE loss [48], rather than other contrastive loss functions such as Euclidean distance-based pair loss [49] or triplet loss [50]. Although the original paper does not explicitly discuss the rationale behind this choice, we speculate that it may be due to the following reasons:

1. **Direction is more important than magnitude.** When computing Euclidean distance, differences in vector magnitude often dominate the distance calculation. In contrast, cosine similarity normalizes the feature vectors, projecting them onto a unit hypersphere. This normalization encourages contrastive learning to focus on angular differences, enabling the model to learn more effective representations in which similar samples lie close together and dissimilar samples are evenly distributed across the hypersphere [51].
2. **The curse of dimensionality.** In high-dimensional spaces, vector distributions tend to be extremely sparse, and distances between points can become uniformly large. Under such conditions, Euclidean distance becomes less discriminative, making it difficult to distinguish between different feature types.
3. **Numerical stability.** Cosine similarity has a bounded range of $[-1, 1]$, whereas Euclidean distance ranges from 0 to ∞. This bounded nature of cosine similarity helps improve numerical stability during training, reducing the risk of exploding gradients or instability caused by large distance values.

Additionally, for generative tasks, such as generating images or textual descriptions, a generative objective can be employed. For tasks involving matching between images and text, an alignment objective can be used.

Vision-language models are typically trained on extremely large datasets. Take CLIP as an examplethe authors constructed a new dataset containing 400 million (image, text) pairs, named WebImageText (WIT) [1]. Unfortunately, the authors did not release their dataset, so we have no way of knowing more details about the training data.

3.1.4 Downstream Tasks

As previously mentioned, vision-language models can be applied to a wide range of downstream tasks. Here, we use CLIP for a zero-shot image classification task as an example to illustrate how a pretrained vision-language model can be applied to downstream tasks.

To enable CLIP to perform an zero-shot image classification task, in addition to providing the image to be classified, we usually need to manually create a prompt

that includes the possible class names, such as "a photo of a [CLASS].", where [CLASS] represents the potential category labels.

After extracting the image and text features using the image encoder and text encoder respectively, we calculate the cosine similarity between the image feature and each text feature. The classification probabilities that the input image $\mathbf{x}_i$ belongs to class k can then be computed using the following formula:

$$P(y_i = k|\mathbf{x}_i, \mathbf{w}_k) = \frac{\exp(\cos(g(\mathbf{w}_k), f(\mathbf{x}_i))/\tau)}{\sum_{j=1}^{K} \exp(\cos(g(\mathbf{w}_j), f(\mathbf{x}_i))/\tau)}. \tag{3.8}$$

Here, $g(\cdot)$ represents the text encoder, $f(\cdot)$ represents the image encoder, $\mathbf{w}$ represents the designed prompt, τ is the temperature parameter, and $\cos(\cdot, \cdot)$ denotes the cosine similarity:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||}. \tag{3.9}$$

The reason for using cosine similarity as the metric for computing classification probabilities is closely related to CLIPs pre-training objective. As introduced in Section 3.1.3, CLIP is trained using a contrastive loss, which in practice maximizes the cosine similarity between true image-text pairs. Although the original loss formulation uses the dot product, the features are normalized to unit length prior to similarity computation, meaning that the dot product is effectively equivalent to cosine similarity. As a result, in the embedding space, an image feature will have the highest cosine similarity with its corresponding text feature. If alternative distance metrics, such as Euclidean distance, were used, this correspondence would no longer be guaranteed.

Compared to traditional classification approaches that rely on an MLP-based classifier, CLIP exhibits stronger classification capabilities. First, cosine similarity-based classification does not rely on a fixed decision boundary, enabling CLIP to handle previously unseen categories. Second, by simply modifying the prompt fed into the text encoder, one can make the textual description better match the image content, allowing for improved classification performance even without fine-tuning the model. This is also the key principle underlying prompt learning.

3.2 Prompt Learning

Since manually designed prompts are difficult to optimize, finding a well-performing prompt can be time-consuming and labor-intensive. To address this, researchers have proposed replacing natural language text with learnable word embeddings. Because these embeddings are continuously distributed in the word embedding space, optimization algorithms can automatically find the optimal prompt.

For example, CoOp designs the prompt as a set of learnable embeddings:

$$\mathbf{p}_k = [V]_1, [V]_2, ..., [V]_M, [\text{CLASS}]_k, \tag{3.10}$$

where $[V]$ represents the word embedding vector in the word embedding space. It can encode class-specific information, domain-specific information, or any other type of information that may guide the model to complete the downstream task. M is a hyperparameter, control the length of soft prompts.

During training, all other parameters of the pre-trained model are kept frozen, and only the soft prompt parameters are optimized using the cross-entropy loss function:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} y_i \log P(\hat{y} = i|\mathbf{x}). \tag{3.11}$$

After obtaining the optimized soft prompt, the probability of an image belonging to each category is calculated using the same formula as CLIP (Equation 3.8).

4

Related Works

In this chapter, we introduce a series of state-of-the-art methods that leverage prompt learning techniques to enhance the generalization ability of vision-language models. These methods will serve as baselines for comparison with our proposed approach in Chapter 6. We also provide a brief analysis of the similarities and differences among them.

Broadly speaking, these methods can be categorized into two groups. The first group uses a fixed soft prompt, meaning that the soft prompt remains unchanged during inference after being trained. For example, CoOp [7], ProDA [52], and BPL [53]. The second group allows the soft prompt to be dynamically adjusted or generated during inference, as seen in methods such as CoCoOp [26], DPL [54], StyLIP [55], DDSPL [56], and SPG[57].

4.1 Methods Based on the Fixed Soft Prompt

CoOp is the first method to introduce prompt learning into the field of vision-language models and remains one of the most well-known and influential approaches in this area. However, its core idea is relatively simple: it replaces manually designed hard prompts with a set of learnable word embedding vectors. We have already provided a detailed introduction to this method in Section 2.2.

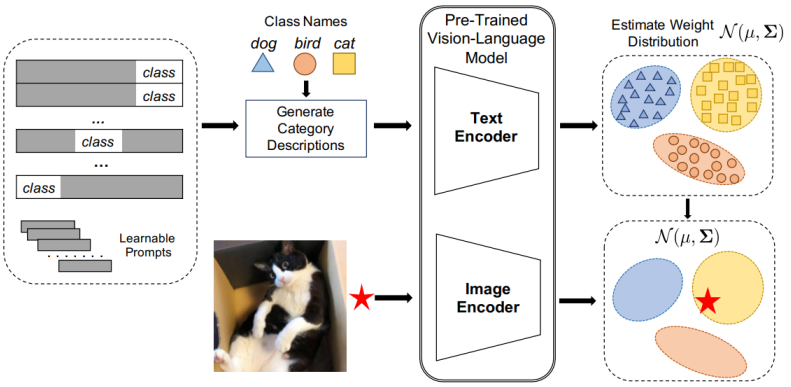


Figure 4.1: Architecture of ProDA. [52]

ProDA aims to enhance the generalization ability of vision-language models by learning a distribution over prompts. The method introduces a learnable set of soft

prompts, where each class is associated with multiple prompts. After being encoded by the text encoder, these prompts produce a distribution of text features for each class in the embedding space. Rather than modeling the prompt set directly, ProDA estimates the distribution of each class by the text features generated from its corresponding prompts. Using multivariate Gaussian modeling, ProDA defines an optimizable upper-bound loss function for training. To promote diversity among prompts during training, the method introduces positional variations in the prompt structure, as well as a semantic orthogonality constraint to enhance the expressiveness of the prompt set. At inference time, the mean of the text feature distribution for each class is used as its representative text feature.

BPL formulates prompt learning as a variational inference problem by introducing a Bayesian framework into the prompt space. Each prompt is composed of a set of fixed learnable vectors added to a global residual vector, which is treated as a latent variable modeled by a learnable Gaussian distribution $r \sim \mathcal{N}(\mu, \Sigma)$. During training, a residual vector is sampled from this distribution and added to all prompt tokens to form a complete prompt. The model is trained by maximizing the variational lower bound, which includes a log-likelihood term for label prediction and a KL divergence term between the posterior and prior of the prompt residual. At inference time, multiple prompts are generated by sampling residuals from the learned distribution, and the corresponding text features are used to produce multiple predictions. The final classification result is obtained by averaging these predictions.

4.2 Methods Based on the Dynamically Adjusted Soft Prompt

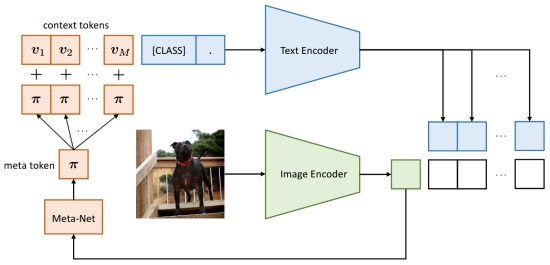


Figure 4.2: Architecture of CoCoOp. [26]

CoCoOp builds upon CoOp by introducing image-conditioned prompt modeling to improve generalization to unseen classes. Specifically, it incorporates a residual adjustment mechanism conditioned on the input image. This mechanism is implemented via a lightweight neural network called Meta-Net, which takes the output of the image encoder as input and produces a conditioning vector $\pi = h_\theta(x)$. This vector is then used to adjust each prompt token v_m as $v_m(x) = v_m + \pi$.

DPL does not directly optimize the soft prompts themselves. Instead, it trains a prompt generator, which is a lightweight MLP network capable of dynamically generating soft prompts based on the image features of the input. To improve

CLIPs generalization ability when dealing with images from different domains, DPL averages the soft prompts generated for all samples within each source domain to obtain a domain-specific prompt.

StyLIP enhances the generalization ability of CLIP in cross-domain image classification tasks by introducing a multi-scale style-conditioned prompt learning mechanism. The core idea is to leverage the style information of an image to guide prompt generation. Specifically, StyLIP uses CLIPs image encoder to extract statistical information (mean and variance) from multi-level convolutional feature maps, forming both style features and multi-scale content features. The style features are processed by a set of Transformer encoders (style projector) to generate conditional embeddings, which are used to control the generation of a set of prompt tokens. Meanwhile, the multi-scale content features are processed by a content projector and then fused with the text features generated by the text encoder. The resulting fused text features are finally used for image classification.

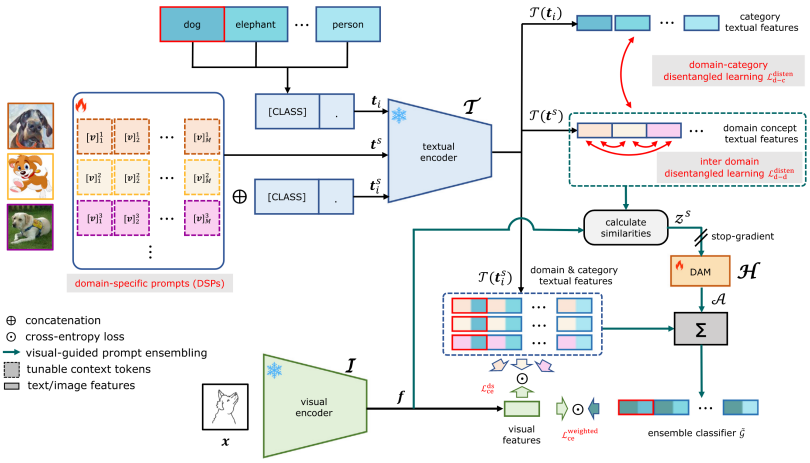


Figure 4.3: Architecture of DDSPL. [56]

DDSPL employs disentangled prompt learning to separate information from different domains, as well as to disentangle domain-specific information from class-specific information. Each source domain is associated with a domain-specific prompt and a corresponding domain concept textual feature. During inference, the image feature of the input is first compared with the domain concept textual features to compute similarity scores. These scores are then processed by a domain attribution module, which generates weights for each domain-specific text feature. Finally, the text features are fused through a weighted combination based on these computed weights.

SPG proposes a generative adversarial framework for prompt learning, aiming to generate domain-adaptive soft prompts. The method involves two training stages: In the first stage, optimal prompt vectors that best represent the characteristics of each source domain are independently learned through training on each domain. In the second stage, a conditional generative adversarial network (GAN) is constructed, consisting of a generator and a discriminator. The generator takes image features and random noise as input and generates soft prompts. The discriminator aims to distinguish whether a generated prompt matches a real prompt. During inference,

only the trained generator is used to generate personalized prompts based on the target image, which are then combined with the class label to perform classification.

4.3 Comparison Analysis

Next, we briefly analyze the main similarities and differences among the selected methods. It is important to note that in selecting state-of-the-art methods, we prioritized diversity, and as such, the differences among these approaches are significantly greater than their commonalities.

1. ProDA, BPL, and CoCoOp all model the semantic information of prompts from a probabilistic perspective. However, ProDA models the distribution of text features directly in the embedding space, whereas BPL and CoCoOp model the distribution of embedding vectors in the word embedding space.
2. Although CoCoOp, DPL, StyLIP, and SPG all adopt dynamic prompt generation strategies by training lightweight networks to generate soft prompts on the fly, DDSPL instead trains a set of soft prompts and performs prompt fusion during inference.
3. Among all the methods, only ProDA and DDSPL involve direct optimization of text features. In combination with the results presented in Section 6.3, it is evident that directly optimizing text features is more beneficial for achieving modality alignment, thereby improving the generalization ability of the model.

5

Methods

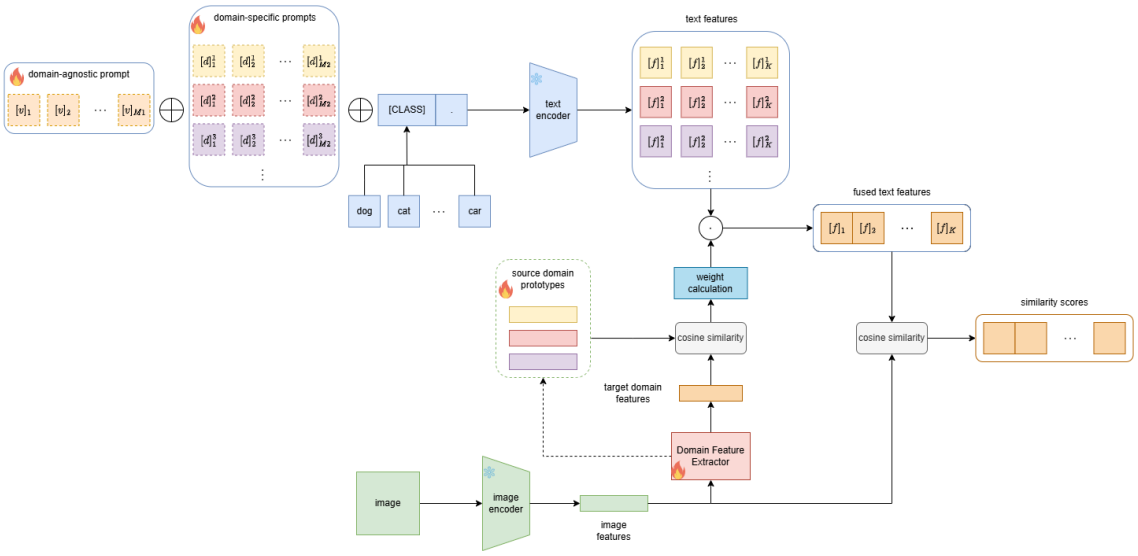


Figure 5.1: Domain Prompt Fusion (DPF) architecture. During training, the text encoder and image encoder are frozen, and only the soft prompts (including the domain-agnostic part and domain-specific part) as well as the Domain Feature Extractor (DFE, including the source domain prototypes) are updated.

For the domain adaptation problem, since we already have information about the target domain and access to some (unlabeled) training data from it, we can design soft prompts specifically tailored to the target domain. This problem has been extensively studied [27], [58]–[61]. However, in the domain generalization scenario, we have no prior knowledge of the target domain. Therefore, it is necessary to design effective mechanisms that fully leverage the information available from the source domains.

Inspired by [56], [62], we propose a Domain Prompt Fusion (DPF) framework. It dynamically fuses soft-prompt text features from different source domains based on the domain feature extracted from the input image, in order to achieve better generalization.

The overall architecture of our method is shown in the Figure 5.1. Our framework is built upon the CLIP model, which consists of a text encoder for extracting textual features, an image encoder for extracting visual features, and a classification mechanism based on cosine similarity. For a detailed explanation of these components,

please refer to Section 3.1. On top of this foundation, our framework additionally includes the following three modules:

1. Soft Prompt, which is further divided into:
   - Domain-agnostic prompt: captures domain-invariant or domain-independent information.
   - Domain-specific prompt: captures domain-relevant information specific to each source domain.
2. Domain Feature Extractor (DFE): extracts domain features from the input image to represent its domain characteristics.
3. Prompt Fusion Mechanism: computes fusion weights based on the extracted domain features, enabling the dynamic combination of text features from different source domains.

Compared to prior works such as CoOp and DDSPL [56], our approach introduces the following key innovations:

1. **Adaptation of soft prompt decomposition to domain generalization.** This design was originally proposed in the context of single-source domain adaptation [27]. While we adopt a similar conceptual structure, we modify the training procedure to make it suitable for multi-source domain generalization. For instance, we introduce an orthogonality loss on the domain-specific prompts to encourage disentanglement and improve domain generalization performance.
2. **A domain-feature-based strategy for computing fusion weights.** We observe that samples from different domains exhibit varying degrees of distributional shift in the embedding space. Based on this observation, we hypothesize that domain information embedded in the input images can be leveraged to guide text feature fusion. Accordingly, we propose a fusion weight computation strategy driven by domain features, which takes into account the similarity among domains.

   By contrast, although DDSPL also adopts a similar fusion mechanism, it computes fusion weights based on the similarity between image features and domain-specific text features, without fully leveraging the domain information embedded in the visual modality of the source domains.

5.1 Soft Prompt Design

In our settings, each prompt is divided into three parts: domain-agnostic, domain-specific, and class label, as follows:

$$\mathbf{p}_k = \underbrace{[v]_1[v]_2\cdots[v]_{M1}}_{\text{domain-agnostic tokens}}\ \underbrace{[d]_1[d]_2\cdots[d]_{M2}}_{\text{domain-specific tokens}}[\text{CLASS}]_k. \tag{5.1}$$

Here, $[v]$ represents the domain-agnostic tokens, $[d]$ represents the domain-specific tokens, and [CLASS] represents the class label. M_1 and M_2 are hyperparameters that control the lengths of the two types of prompts, and k represents the k-th class.

All domains share the same domain-agnostic prompt, while an independent domain-specific prompt is trained for each domain. The training of the soft prompts is performed using the cross-entropy loss function $\mathcal{L}_{ce}$ (Equation 3.11). Additionally, we introduce an orthogonality constraint to ensure that the prompts for different domains are as distinct as possible:

$$\mathcal{L}_{orth} = ||G - diag(G)||_F^2 = \left(\sqrt{\sum_{i \neq j}(w_i w_j)^2}\right)^2, \tag{5.2}$$

where $w_i \in \mathbb{R}^{M_2 \cdot dim \times 1}$ is the domain-specific prompt vector, dim is the word embedding dimension of each tokens (which is 512 in the CLIP). G is the gram matrix of the domain-specific prompt vector, $G_{i,j} = w_i \cdot w_j$, and $|| \cdot ||_F$ is the Frobenius norm.

5.2 Domain Feature Extractor

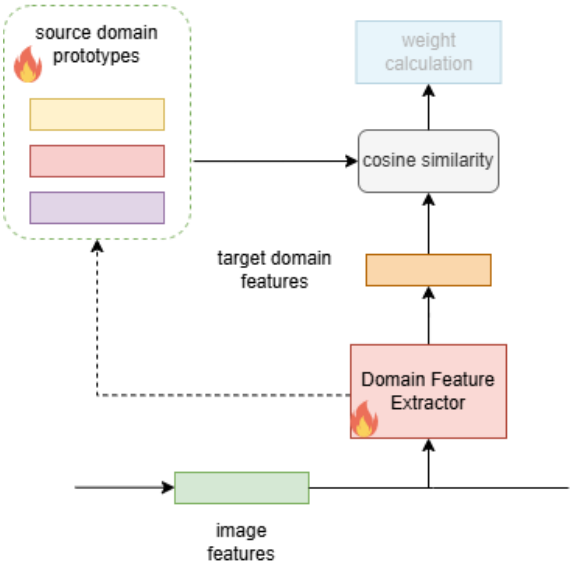


Figure 5.2: Local architectural diagram of the domain feature extractor. The dashed lines indicate processes that occur only during training.

In our design, we aim to fully leverage the domain information embedded in the image to help determine which domain the input image belongs to, or to compute its similarity to different domains. To achieve this, we introduce the Domain Feature Extractor (DFE) module. The DFE is essentially a lightweight MLP network placed after the image encoder, as shown in Figure 5.2. It further extracts domain-related features from the image features produced by the encoder. The DFE consists of three fully connected layers, two dropout layers, and uses the ReLU activation function.

Through visualization analysis (see Section 6.8.1), we observe that image features from different domains are mixed together in the embedding space, making them

difficult to distinguish. In order to differentiate images from different domains, we want the DFE to project image features from the embedding space into a new domain feature space, where features from the same domain are as close as possible, and features from different domains are as far apart as possible. And domain features should not exhibit pronounced class clustering; samples of different classes within the same domain should be uniformly mixed rather than forming separate clusters.

We use the prototypical loss [63] as the objective function to achieve this. The computation of the prototypical loss can be viewed as a form of hard clustering, which aligns perfectly with our design objective. Other loss functions, such as contrastive loss or triplet loss, are based on the similarity or dissimilarity between sample pairs, but they do not simultaneously consider the inter-domain similarity across all samples within a domain or the inter-domain dissimilarity across all samples from different domains. Moreover, the computed prototypes can be directly used to determine the domain of an input sample. This approach is simple, efficient, and highly interpretable, as it eliminates the need to train an additional classifier to separate samples from different domains.

Let $f(\cdot)$ represent the image encoder and $h(\cdot)$ represent the DFE. First, we compute the prototype for each domain, which can be interpreted as the cluster center in k-means:

$$p_m = \frac{1}{|S_m|} \sum_{x_i \in S_m} h(f(x_i)), \tag{5.3}$$

where S_m is the sample set of domain m. Then, the probability of a sample belonging to domain m can be computed using the following formula:

$$p\Big(y = m \mid x_i\Big) = \frac{\exp(-d(h(f(x_i)),\, p_m))}{\sum_{j=1}^{M} \exp(-d(h(f(x_i)),\, p_j))}, \tag{5.4}$$

where $d(\cdot)$ denotes the Euclidean distance in the domain feature space,

$$d\Big(h(f(x_i)),\, p_m\Big) = \parallel h(f(x_i)) - p_m \parallel^2. \tag{5.5}$$

The use of Euclidean distance in this context follows the design choices made in the original paper [63]. However, in accordance with recommendations from [51], we apply normalization to the domain features $h(f(x_i))$ prior to computing the prototypical loss, projecting them onto a unit hypersphere. Under this condition, the difference between Euclidean distance and cosine similarity becomes negligible, as $||x - y||^2 = 2 - 2xy = 2 - 2\cos(x, y)$.

The total loss function is defined as follows:

$$\mathcal{L}_{proto} = -\frac{1}{N} \sum_{i=1}^{N} \log p\Big(y = m \mid x_i\Big). \tag{5.6}$$

After training, the DFE saves the prototypes of each source domain computed in the final iteration. During inference, the DFE projects the target domain image features into the domain feature space, and then calculates their cosine similarity to each of the stored source domain prototypes.

5.3 Prompt Fusion Mechanism

The Prompt Fusion Mechanism is the core of our proposed method. Its computed weights directly determine the effectiveness of the fused prompt and ultimately affect the classification accuracy. In our design, we use the text features from different domains as a set of basis vectors, and then compute a linear combination of these basis vectors to obtain the text feature for the target domain:

$$\tilde{\mathbf{f}}_i = \sum_m \alpha_m \mathbf{f}_i^m. \tag{5.7}$$

We fuse the text features of different domains for each class separately. Here, $\mathbf{f}_i^m$ denotes the text feature of class i from domain m, $\tilde{\mathbf{f}}_i$ denotes the fused text feature of class i, and α represents the weights of the domain text features. The fusion weights are computed based on the cosine similarities obtained from the DFE. Currently, we use a relatively simple calculation method: we compute the probability of the target domain feature belonging to each source domain using the cosine similarity, and directly use these probabilities as the fusion weights:

$$\alpha_m = \frac{\exp(\cos(h(f(x_i)),\, p_m)/\tau)}{\sum_{j=1}^{M} \exp(\cos(h(f(x_i)),\, p_j)/\tau)}. \tag{5.8}$$

After obtaining the fused text feature for each class, we compute the cosine similarity between the image feature and these fused text features to perform image classification.

5.4 Training Strategy

During training, we keep the original parameters of the CLIP model (both the text encoder and image encoder) frozen, and only train the soft prompts and the Domain Feature Extractor (DFE).

- When training the domain-agnostic prompt, we use only the cross-entropy loss ($\mathcal{L}_{ce}$).
- When training the domain-specific prompts, we apply both the cross-entropy loss and the orthogonality loss ($\mathcal{L}_{ce} + \mathcal{L}_{orth}$) to encourage diversity between prompts.
- For training the DFE, we use the prototypical loss ($\mathcal{L}_{proto}$).

Both the soft prompts and DFE are trained on a dataset that includes all data from all source domains. This means we train them jointly across all source domains, rather than sequentially training on each source domain separately.

6

Results

In this chapter, we will introduce the datasets used, the baselines for comparison, implementation details, and the results of various tests. We will also provide a detailed analysis of these results.

6.1 Dataset and Baseline

Our main goal is to evaluate the models domain generalization ability, so we will primarily conduct experiments on the Office-Home dataset. Office-Home is a domain generalization dataset, which includes four domains (Art, Clipart, Product, and Real-World) with 65 categories collected from everyday objects, totaling over 15,500 images. We also trained and evaluated our method on mini-DomainNet [33], a benchmark dataset specifically designed for domain generalization, which is constructed by sampling a subset of images from the larger DomainNet [41] dataset. It comprises 140,006 images divided into four domains: Real, Painting, Sketch, and Clipart. We will use the leave-one-domain-out evaluation protocol [33], meaning that in each experiment, we leave one domain as the test set while using the other domains as the training set. The evaluation metric is accuracy.

(a) Office-Home (b) Mini-DomainNet

Figure 6.1: Example figures from Office-Home and mini-DomainNet

We compare our method with zero-shot CLIP and prompt-learning or adapter-based approaches. Zero-shot CLIP refers to directly using the pre-trained CLIP model for image classification on the target domain, without any additional training or fine-tuning. We consistently use the prompt template "a photo of a [CLASS]." as the input to the text encoder.

We also select CoOp [7], CoCoOp [26], ProDA [52], CLIP-Adapter [31], DPL [54],

BPL [53], StyLIP [55], DDSPL [56], and SPG [57] as our baselines. Additionally, we retrained and evaluated only zero-shot CLIP and CoOp; results for the remaining methods were obtained from the literature [55]–[57] without further validation.

6.2 Experimental Setup

We implement our code based on the CoOp framework and the Dassl library [14], [33]. For the pre-trained CLIP model, we use ViT-B/16 [25] as the image encoder. Both the domain-agnostic prompt and domain-specific prompt are set to a length of 8 tokens. When initializing the soft prompts, we use random initialization, sampling from a Gaussian distribution with a mean of 0 and a variance of 0.02, consistent with the setup in CoOp. The coefficient of the orthogonal loss for the domain-specific prompt is 10.0. The Domain Feature Extractor (DFE) consists of three randomly initialized fully connected layers, the input dimension matches the feature dimension in the embedding space, which is 512; The hidden layer dimension is 256; The output layer dimension is 512. We place a dropout layer with a dropout rate of 0.2 between every two fully connected layers. Except for the number of training epochs, both the soft prompts and DFE share the same optimizer settings. We use the Adam [64] optimizer. The initial learning rate is set to 0.002, and we apply a cosine annealing learning rate scheduler to gradually decrease the learning rate during training. Batch size is 32 on the Office-Home dataset and 256 on mini-DomainNet. The soft prompts are trained for 10 epochs, and the DFE is trained for 200 epochs. The fusion temperature is set to 0.8.

As a baseline, CoOp is trained on all source domains. The prompt length is set to 16 tokens, and all other hyperparameters are kept consistent with those used when training our proposed method.

All training and testing procedures were conducted on the Alvis server cluster provided by NAISS. We primarily utilized compute nodes equipped with a NVIDIA Tesla A40 GPU and an Intel(R) Xeon(R) Gold 6338 CPU @ 2GHz.

6.3 Comparison with Baselines

6.3.1 Evaluations on Office-Home

Table 6.1 presents the testing results of our method compared to the baselines on the Office-Home dataset. Our method outperforms zero-shot CLIP by 2.95% and the classic prompt-learning approach CoOp by 1.47%, demonstrating its effectiveness in enhancing CLIPs robustness to domain shift. Furthermore, our approach achieves better performance than most baselines. However, we acknowledge that our method trails the state-of-the-art DDSPL by 0.61% and ranks third overalljust 0.07% behind ProDA, indicating room for further improvement.

On each domains, compared to zero-shot CLIP, our method achieves the largest gain on Clipart (+4.4%) and the smallest gain on Real World (+1.7%). Relative to the best-performing methods, our largest deficit occurs on Art, where we are 2.13%

Methods	Art	Real world	Clipart	Product	Average
CoCoOp	79.60	86.32	69.35	87.51	80.70
Zero-shot CLIP	80.50	89.10	70.20	88.30	82.03
CLIP-Adapter	82.76	88.02	70.08	88.04	82.23
CoOp	80.70	90.13	72.47	90.77	83.51
SPG	81.60	89.90	72.70	90.20	83.60
BPL	83.02	90.83	72.01	90.21	84.02
DPL	82.50	91.50	71.70	91.20	84.23
StyLIP	**84.93**	90.64	72.61	90.35	84.63
DPF (Ours)	82.80	90.80	74.60	91.70	84.98
ProDA	83.44	91.13	73.79	91.84	85.05
DDSPL	83.26	**91.51**	**75.03**	**92.54**	**85.59**

Table 6.1: Comparison of accuracy across different methods on the Office-Home dataset. The name of each column represents the target domain used during testing, while the other three domains serve as the source domains for training in that setting. Results are sorted in ascending order of average accuracy.

below StyLIP, and the smallest deficit is on Clipart, where we are 0.43% below DDSPL. Based on the subsequent visualization analysis (Section 6.8.1), we hypothesize that our methods poorer performance on the Art domain may be due to the fact that, for certain classes, the Art-domain image features differ only marginally from those of other domains, while for other classes the differences are more pronounced. This uneven discrepancy results in fused text features that fail to achieve consistent modality alignment with image features across all classes.

Methods	Painting	Real	Clipart	Sketch	Average
Zero-shot CLIP	82.50	91.60	82.70	79.60	84.10
CoOp	83.63	89.73	84.87	79.43	84.42
BPL	83.01	92.21	85.03	80.85	85.28
DPF (Ours)	**84.60**	91.60	84.50	**81.40**	85.53
CoCoOp	83.78	91.60	86.50	81.34	85.81
ProDA	84.39	92.20	86.23	81.29	86.03
DDSPL	84.58	**92.37**	**86.59**	81.37	**86.23**

Table 6.2: Comparison of accuracy across different methods on the mini-DomainNet dataset. The name of each column represents the target domain used during testing, while the other three domains serve as the source domains for training in that setting. Results are sorted in ascending order of average accuracy.

6.3.2 Evaluations on Mini-DomainNet

Table 6.2 presents the testing results of our method compared to the baselines on the mini-DomainNet dataset. On mini-DomainNet, our method still outperforms zero-shot CLIP and CoOp by 1.43% and 1.11%, respectively. However, it trails the

state-of-the-art DDSPL by 0.7% and underperforms ProDA (by 0.5%) and CoCoOp (by 0.28%), ranking fourth among all methods. This may be because we applied the exact same hyperparameters used for Office-Home without any adjustment; further hyperparameter tuning could potentially improve our results.

Analysis by domain shows that our method achieves the best performance among all methods on Painting and Sketch. However, on Real, our method performs on par with zero-shot CLIP but is 0.77% behind DDSPL; on Clipart, although our method outperforms zero-shot CLIP by 1.8%, it falls 0.37% short of CoOp and 2.09% short of DDSPL. Due to time constraints, we did not visualize the image feature distributions on mini-DomainNet, but we suspect the causes are similar to those observed on Office-Home.

6.4 Ablation Study

Setting	Art	Real world	Clipart	Product	Average
DAP only	82.5	90.7	73.4	91.6	84.55
DSP only	**83.2**	90.3	73.2	91.3	84.50
remove DSP's $\mathcal{L}_{orth}$	**83.2**	90.3	74.1	91.1	84.68
remove DFE	82.2	**90.8**	74.4	91.0	84.60
greedy fusion	82.3	90.4	73.6	90.3	84.15
average fusion	82.2	**90.8**	**74.6**	91.1	84.68
DPF (Ours)	82.8	**90.8**	**74.6**	**91.7**	**84.98**

Table 6.3: Ablation study results on Office-Home dataset. "DAP only" refers to using only the domain-agnostic prompt, with all other modules unchanged. "DSP only" refers to using only the domain-specific prompt, with all other modules unchanged. "Remove DFE" means no longer using the domain feature extractor to guide fusion; instead, the raw image feature is used directly. "Greedy fusion" refers to replacing weighted fusion with directly using the text feature from the domain with the highest similarity. "Average fusion" refers to omitting similarity calculations and directly using the mean of the text features from all source domains.

The ablation study results are presented in the Table 6.3. In this experiment, we remove or replace individual modules in our framework and observe the impact on overall performance. As shown, the full configuration achieves the best results, confirming the validity of our design; eliminating or altering any component leads to performance degradation. For example, removing the domain feature extractor and using the raw image feature to guide fusion results in a 0.38% drop in accuracy, indicating that the image feature alone does not sufficiently distinguish samples from different domains. Similarly, replacing weighted fusion with greedy fusion yields an accuracy decrease of 0.83%, demonstrating that even the most similar source domains text feature cannot reliably achieve modality alignment with the target-domain image feature and that weighted fusion is necessary.

However, domain-specific observations reveal unexpected findings. On the Art

domain, using only the domain-specific prompt outperforms the combination of domain-agnostic and domain-specific prompts, suggesting that the domain-invariant information learned from the source domains may not transfer well to the target domain and can mislead the model. Additionally, on the Real World and Clipart domains, average fusion produces results equivalent to our weighted fusion method, implying that the features in these domains are effectively the mean mixture of the other three source domains.

6.5 Analysis of Changing Soft Prompt Length

Prompt length	Art	Real world	Clipart	Product	Average
4+4	82.7	90.5	73.1	91.2	84.38
8+8	82.8	90.8	74.6	91.7	84.98
16+16	81.6	90.7	74.1	92.0	84.60

Table 6.4: Comparison results on Office-Home dataset. The prompt length is given in the order of "domain-agnostic prompt + domain-specific prompt".

We experimented with changing the length of the soft prompts and conducted repeated tests on the target domain, as Table 6.4. We evaluated three length configurations. The results show that the medium-length soft prompt (8+8) exhibits the strongest generalization capability, followed by the longest (16+16), while the shortest (4+4) performs worst, which is 0.6% lower than the medium setting. This outcome may be due to the fact that shorter soft prompts cannot capture sufficient informative cues to guide the models classification, whereas longer soft prompts may encode excessive source-domain specific information, leading to some overfitting and reduced generalization.

6.6 Analysis of Domain Feature Extractor

6.6.1 Change the Dimension of the Domain Feature

The results of varying the output dimensionality of the domain feature extractor (the dimension of the domain feature) are shown in the Table 6.5. Due to time constraints, we tested on only one target domain, but the findings should generalize across all domains. Note that on the training set, we assess the domain feature extractor by its domain-classification accuracy (i.e., its ability to correctly identify the domain of each sample), rather than by class-classification accuracy. This is because an ideal domain feature should be class-agnostic, meaning that samples from the same domain, regardless of class, should yield similar domain features. On the test seti.e., the target domainwe evaluate using class-classification accuracy, since our focus there is the domain featureguided text-feature fusion capability, and because the target-domain samples cannot be mapped onto any of the source domains for domain classification.

Dimension	Source domain accuracy	Target domain accuracy
512	89.90	74.60
256	89.68	74.50
128	89.61	74.70
64	90.10	74.30

Table 6.5: Training and testing accuracies for different domain feature dimensions. Results were obtained on the Office-Home dataset with source domains Real World, Product, and Art, and target domain Clipart. "Source domain accuracy" refers to the accuracy on the domain classification task over the source domains, and "Target domain accuracy" refers to the accuracy on the image classification task in the target domain.

We observe that when the domain feature dimension is between 128 and 512, the training and testing accuracies are roughly equivalent, with only minor differences. However, when the domain feature dimension is lower, such as 64, even though the source-domain accuracy remains high, the target-domain accuracy drops noticeably. We speculate that at lower dimensions, the feature space is insufficient to fully capture the domain characteristics of the images, leading to overfitting on the source domains and a consequent reduction in generalization ability.

6.6.2 Change Design

Setting	Art	Real world	Clipart	Product	Average
With dropout	82.8	90.8	74.6	91.7	84.98
Without dropout	82.4	90.6	74.3	91.0	84.58
Statistics (all layers)	82.4	90.9	74.6	91.3	84.80
Statistics (last 4 layers)	82.6	90.8	74.5	91.6	84.88
Statistics (last layer)	82.4	90.9	74.6	91.4	84.83

Table 6.6: Results of adjusting the design of domain feature extractor on Office-Home dataset.

We tested alternative schemes for extracting domain features to validate the architectural soundness of our domain feature extractor, including removing the dropout layers from the network, and trying methods similar to [65], [66], extracting statistics from intermediate layers of the image encoder to construct domain features, rather than relying solely on the features after the image encoder. The "without dropout" group was trained for 2,000 epochs. For the three "statistics" groups, the network was a simple linear layer trained for 10 epochs with a fusion temperature of 3.0. All other hyperparameter settings remained unchanged.

Here we simply introduce the statistics method. For each layer's output from the image encoder, the mean and standard deviation are extracted. For a ViT-B/16, the output of each layer has dimensions (batch_size, n_tokens, dim), where n_tokens =

$16 + 1$ ([CLS]) and dim $= 768$. For all tokens except the [CLS] token, the mean and standard deviation are computed along each dimension in the "dim" axis, yielding a vector of size 2×768, denoted as $[\mu_l, \sigma_l]$. These vectors from selected L layers are then stacked to form a vector $[\mu_1, \sigma_1, \cdots, \mu_L, \sigma_L]$ with dimensions $L \times 2 \times 768$, which is subsequently passed through a linear layer to compress it to 512 dimensions.

We observe that adding dropout layers does improve the generalization capability of the domain feature extractor to some extent, enabling it to extract more effective domain features (in terms of guiding textfeature fusion). Among the statisticsbased methods, all outperform the DFE without dropout, but still underperform the DFE with dropout. This may be because those methods were originally designed for CNNs and are not particularly suited to vision transformers.

An interesting finding is that performance varies depending on which layers statistics are used. For example, using the last four layers yields the best DFE performance, whereas using all layers performs worse. This likely relates to the differing distributions of statistics across layers. Theoretically, in vision transformers, higher layers extract more abstract, global image features, while lower layers focus on local, detailed features. Therefore, for our purposes, features from higher layers are more helpful for distinguishing samples from different domains.

6.7 Analysis of Fusion Mechanism

Temperature	Art	Real world	Clipart	Product	Average
0.1	82.4	90.5	73.9	90.3	84.28
0.5	82.6	90.8	74.5	90.8	84.68
0.8	82.8	90.8	74.6	91.7	84.98
1.0	82.6	90.8	74.5	91.2	84.78

Table 6.7: Results of adjusting fusion temperature on Office-Home dataset.

The results of adjusting the fusion temperature on the Office-Home dataset are shown in the Table 6.7. Lower temperatures produce a "sharper" weight distribution, i.e. closer to greedy fusion; whereas higher temperatures yield a "flatter" distribution, i.e. closer to average fusion. The results indicate that a temperature of 0.8 achieves the best fusion performance; raising or lowering the temperature from this value degrades the results. This suggests that an appropriately contracted weight distribution better aligns the fused text features with the image features.

6.8 Visualization

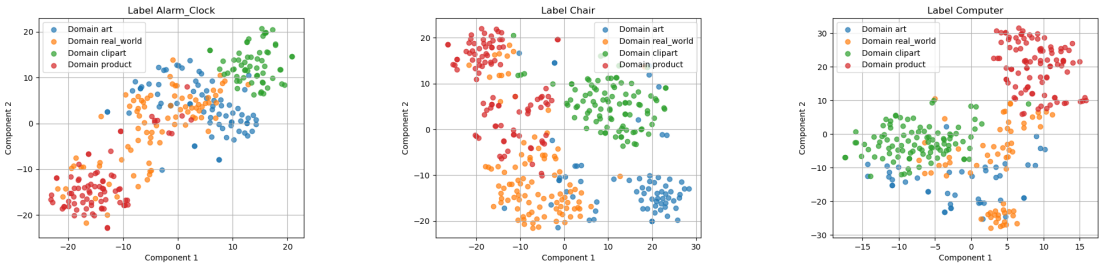
In this section, we will visualize the image features and domain features of samples from the Office-Home dataset using t-SNE [67] to gain a more intuitive understanding of domain shift and the effectiveness of the domain feature extractor. t-SNE is a classic nonlinear dimensionality reduction technique. The basic idea

is that, it measures pairwise similarity in the high-dimensional space using Gaussian distributions, computes pairwise similarity in the low-dimensional space using t-distributions, and minimizes the KL divergence between the two to enable visualization of high-dimensional data in a low-dimensional space.

We used the `TSNE` function provided by the scikit-learn library, with all parameters kept at their default values. It should be noted that in t-SNE visualizations, the distances between clusters have no intrinsic meaning. Greater separation does not necessarily indicate larger distributional differences. However, cluster overlap can give a rough indication of proximity in the original feature space: clearly separated clusters suggest substantial distributional divergence, whereas overlapping clusters imply greater similarity.

For comparative analysis, we also repeated all experiments in this section using the UMAP [68] method. Like t-SNE, UMAP is a nonlinear dimensionality reduction technique. However, UMAP leverages principles from topology and manifold learning to project high-dimensional vectors into a lower-dimensional space, offering a better balance between preserving local and global structures compared to t-SNE. The overall trends observed in the visualizations of both image features and domain features are consistent across the two dimensionality reduction techniques. The UMAP visualization results can be found in Appendix A.2.

6.8.1 Visualization of Domain Shift



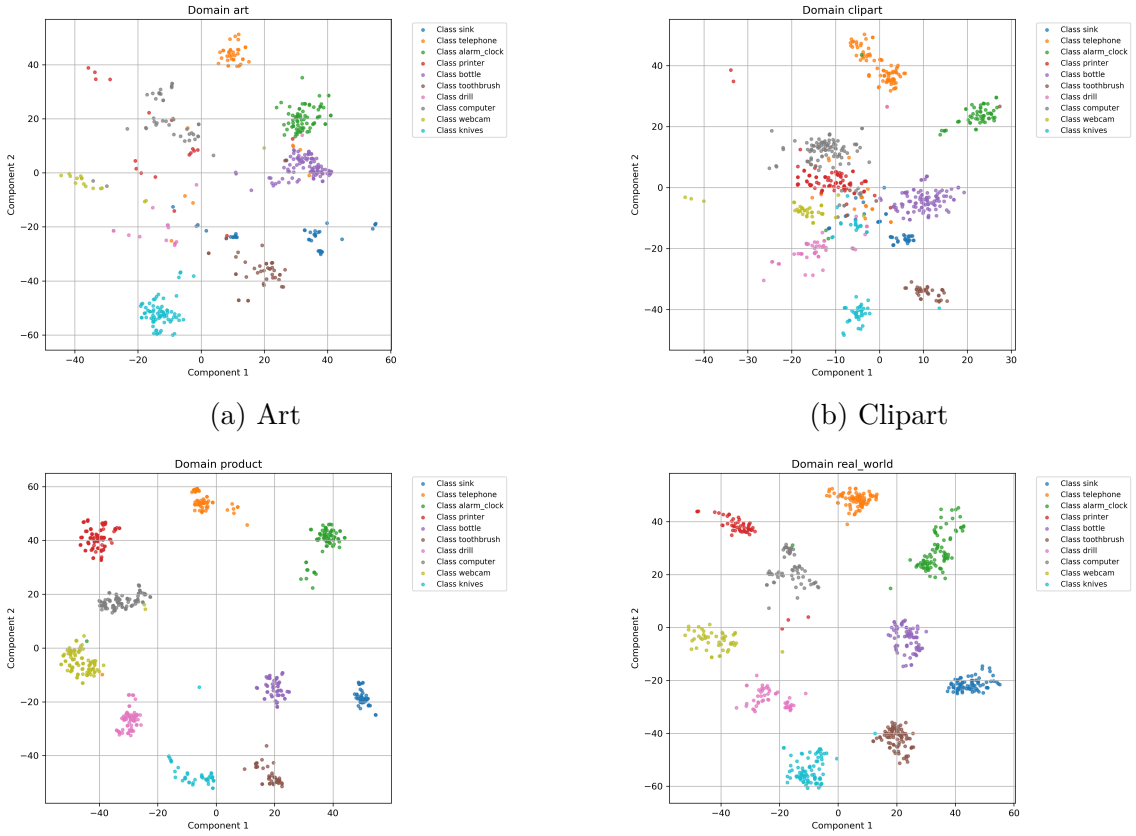
(a) Alarm Clock (b) Chair (c) Computer

Figure 6.2: Distribution of image features of the same class across different domains. Three classes were randomly selected from all classes for visualization.

Figure 6.2 illustrates the distribution of image features for the same class across different domains. From the visualization, we can draw the following conclusions. First, there is indeed a certain degree of domain shift in CLIPs embedding space, manifested as distributional differences among samples from different domains. For example, in the "Chair" category, the four domains form clearly separated clusters with distinct boundaries. Second, the extent of domain shift varies across categories. Although the "Chair" category shows pronounced distributional differences, in the "Alarm Clock" category, Real World overlaps considerably with Art and Product; in the "Computer" category, Real World also overlaps significantly with Art. This implies that the degree of domain shift differs by category, which can affect the final classification performance.

By contrast, Figure 6.3 shows the distribution of image features for different classes within the same domain. Comparing both sets of visuals, we see that inter-class distribution differences are significantly greater than inter-domain differences, as there is almost no overlap between samples of different classes. However, in light of the results from Section 6.3.1, we observe a clear relationship between class distribution and classification accuracy. For example, in the Product and Real World domains, class boundaries are sharp, resulting in higher accuracy; in Clipart, however, classes exhibit some mixing, leading to a noticeable drop in accuracy.

This inter-domain variation in class-wise sample distributions can also be regarded as a form of domain shift. Theoretically, an ideal feature extractor for classification should be immune to domain-specific information and consistently extract class-relevant features, such that class distributions remain similar across domains. These are known as domain-invariant features. In our framework, we do not modify image features to counteract this type of domain shift, because altering image features without fine-tuning the backbone is challenging and prone to overfitting on small datasets, leading to catastrophic forgetting [3]. Given our time constraints, we defer investigation of this issue.



(a) Art

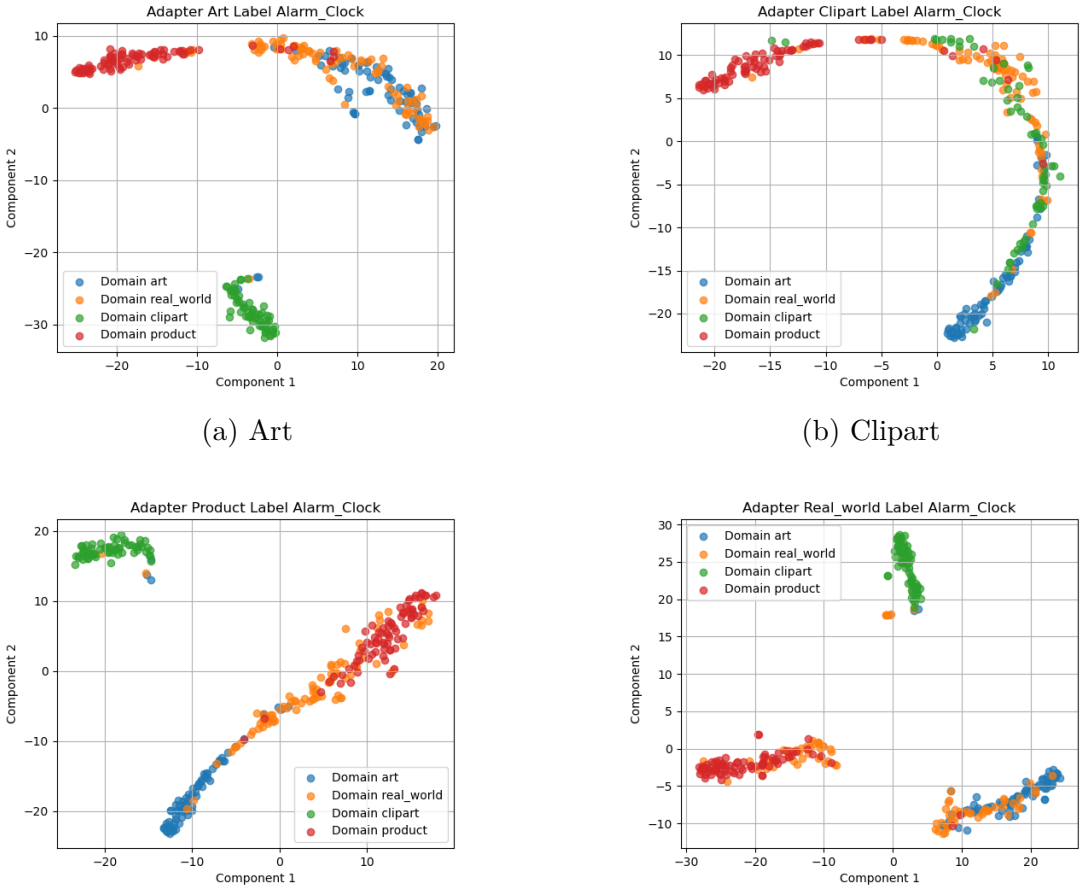
(b) Clipart

(c) Product

(d) Real world

Figure 6.3: Distribution of image features of different classes within the same domain. Ten classes were randomly selected for visualization.

6.8.2 Visualization of Domain Features



(a) Art

(b) Clipart

(c) Product

(d) Real world

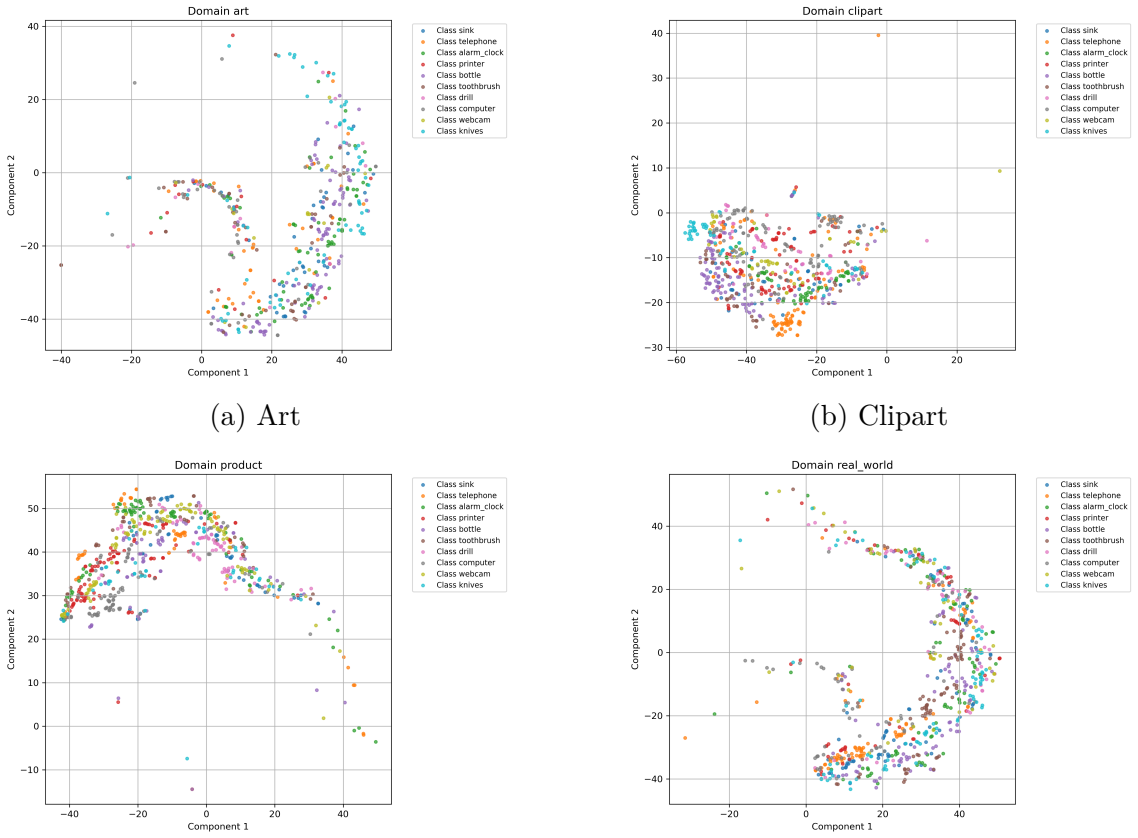
Figure 6.4: Distribution of domain features for the "Alarm Clock" category after extraction by the domain feature extractor. Each domain name indicates that it was treated as the target domain, with the DFE trained on the other three source domains.

Figure 6.4 shows the distribution of domain features extracted by the DFE. Due to space limitations, we only present the results for the "Alarm Clock" category; other categories exhibit similar patterns, see Appendix A.1. We can see that after training, the DFE is indeed capable of distinguishing samples from different source domains, particularly separating Clipart from the other source domains.

It is also noteworthy that, except when Real World is treated as the target domain, the samples from Real World, Art, and Product do cluster, but their boundaries are not sharply defined. In the next sections analysis, we will show that this blurred separation actually aligns with our requirements.

Another very interesting observation concerns the distribution of target-domain samples. During training, we have no access to any target-domain training samples, so we cannot make any assumptions or impose constraints on their distribution. However, the results show that the DFE can extract domain features very effectively.

Figure 6.2a illustrates the actual distribution of image features for the “Alarm Clock” class across the four domains. Despite never seeing any target-domain samples during training, the DFE correctly projects them onto similar source domains: Art overlaps with Real World, Clipart overlaps with both Art and Real World, and Product overlaps with Real World. This effectiveness of the domain features is precisely what ensures the success of the subsequent fusion process.



(a) Art (b) Clipart

(c) Product (d) Real world

Figure 6.5: Distribution of domain features for different classes. The DFE was trained on source domains Clipart, Real World, and Product. Ten classes were randomly selected for visualization.

To demonstrate that the DFE extracts class-agnostic domain features, we visualized the domain features of samples from different classes, as shown in Figure 6.5. Due to space constraints, we only present the domain features extracted by one DFE; other DFEs show similar results. The figure shows that samples of different classes are thoroughly mixed in the domain feature space, with no discernible class separation. This indicates that the DFE indeed focuses solely on domain-related features while discarding most class-related information, further validating the effectiveness of our DFE design and training.

6.8.3 "Ideal" Domain Feature Extractor

Based on the analysis in the previous section and the results of the ablation study in Section 6.4, we believe that the rationality and effectiveness of the domain feature extractor design have been sufficiently demonstrated. However, some potential concerns may still arise: by projecting the domain features of the target domain onto one or more source domains, is there a risk that the model might mistakenly classify them as belonging to those source domains? Would this truly lead to optimal performance? If it were possible to train an "ideal" DFE that ensures the target domain features remain largely independent from the distributions of all source domains, could the model then recognize that the target samples originate from a novel domain, thereby yielding improved results?

Although it is not entirely impossible for the DFE to extract domain features from the target domain that are independent of those from the source domains, achieving this is indeed highly challenging, primarily because no target domain training data or even prior information is available during the training phase. However, from a purely theoretical perspective, we can adopt a convenient approximation: directly including target domain samples in the training set. This approach would explicitly guide the DFE to learn how to distinguish between target and source domain samples, thereby resulting in what could be considered an "ideal" DFE.

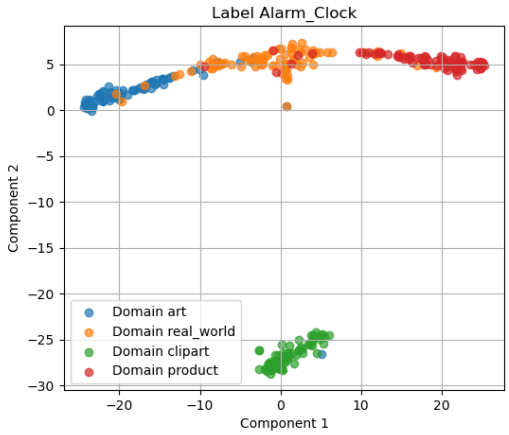


Figure 6.6: Distribution of domain features for the "Alarm Clock" class extracted by the "ideal" domain feature extractor.

Setting	Art	Real world	Clipart	Product	Average
"Ideal" DFE	82.2	90.7	74.6	91.2	84.68
"Normal" DFE	82.8	90.8	74.6	91.7	84.98

Table 6.8: Comparison between domain features extracted using the "ideal" domain feature extractor and those extracted by a DFE trained normally on source-domain samples.

Experimental results show that the domain classification accuracy of the ideal DFE

reaches 90.38%. A visualization of the domain features extracted by the "ideal" DFE for the "Alarm Clock" category is presented in Figure 6.6. As can be seen, the model is indeed capable of distinguishing samples from different domains. Although the boundaries among the Real world, Product, and Art domains are somewhat ambiguous, samples from each domain still tend to form distinct clusters.

We then replaced the "normal" DFE (trained on the source domains) with the "ideal" DFE to guide the fusion of text features. The results, as shown in Table 6.8, reveal that despite the "ideal" DFE's ability to separate samples by domain, it performs worse in guiding text feature fusion compared to the "normal" DFE. This suggests that, from the perspective of fusion guidance, explicitly separating target domain samples from those of the source domains provides no benefit.

If we compare this result with the "average fusion" baseline in Table 6.3, we can draw an interesting conclusion: the performance of the "ideal" DFE is almost identical to that of average fusion. This suggests that when different domains are completely separated, the model tends to treat the target domain as dissimilar to all source domains, resulting in nearly equal fusion weights across them. In this case, enforcing strict domain separation undermines the ability of the domain features to reflect inter-domain similarity, thereby rendering the fusion process ineffective.

In fact, the results shown in Figure 6.4 are precisely what we aim to achieve. Our objective is for the DFE to enhance the separability of samples from different domains, while simultaneously preserving the original distribution of image features in the embedding space. That is, if samples from two domains are mixed in the embedding space, their corresponding domain features should also be mixed in the domain feature space. This consistency is essential for guiding the fused text features to achieve proper modality alignment with the original image features.

6.9 Results Discussion

Through extensive experimental validation, we have demonstrated that our method can effectively enhance the generalization capability of CLIP, particularly in its ability to counter domain shift under domain generalization settings. On the Office-Home dataset, our approach outperforms zero-shot CLIP by 2.95% in accuracy, and by 1.43% on the mini-DomainNet dataset. When compared to baseline methods, our approach also achieves superior performance over most of them, including well-established methods such as CoOp and CLIP-Adapter.

In conjunction with the visualization results presented in Section 5.8, we believe that the primary reason for the significant performance improvement of our method lies in its ability to achieve better modality alignment. Based on the principle behind CLIPs zero-shot classification (see Section 3.1.4), achieving higher classification accuracy requires ensuring that the image feature has the highest cosine similarity with the text feature corresponding to the correct class, while maintaining lower similarity with text features of incorrect classes.

However, due to the presence of domain shift, samples from different domains ex-

hibit distributional discrepancies in the embedding space. As a result, their cosine similarities with a given text feature can vary significantly. This variation may lead to situations where the image feature is more similar to the text features of incorrect classes, thereby causing misclassification. This phenomenon explains why methods that rely on a fixed prompt, such as zero-shot CLIP and CoOp, tend to perform poorly in the domain generalization problem.

In our method, instead of learning a single fixed soft prompt, we first learn a set of soft prompts on each source domain, each embedding domain-specific information. The corresponding text features are then treated as a set of basis vectors in the embedding space. For a given input image, we compute a linear combination of these basis vectors based on its domain feature. This strategy, dynamically adjusting the text feature according to the input image's domain characteristics, allows the fused text feature to better account for distributional differences caused by domain shift. As a result, it achieves improved alignment with the image feature, leading to higher classification accuracy.

Besides, in our design, we choose to adjust only the text features while keeping the image features unchanged. This decision is based on the observation that, during inference, the text encoder merely extracts features from known soft prompts, whereas the image encoder must process previously unseen images to extract meaningful representations, which is an inherently more difficult and complex task. Arbitrarily modifying the image features risks disrupting the knowledge already learned by the image encoder, potentially leading to catastrophic forgetting and significantly reduced generalization performance. This may partly explain the suboptimal performance of methods like CLIP-Adapter.

Furthermore, our method employs a fusion mechanism that performs a linear combination over a set of basis vectors. This is more efficient and stable compared to methods that dynamically generate entire soft prompts, such as CoCoOp. Of course, in theory, if one could train a network that generates perfectly aligned soft prompts in the embedding space for each image, that would yield optimal results. However, due to the limited training data and the unknown nature of the target domain, such generative networks often suffer from overfitting on the source domains, resulting in poor generalization to unseen domains.

However, we also acknowledge that there remains a performance gap between our method and the current state-of-the-art method DDSPL, with our approach trailing by 0.61% on Office-Home and 0.7% on mini-DomainNet. This indicates that there is still room for further improvement. Based on our analysis, we hypothesize that the reason our method did not reach state-of-the-art performance may lie in the following aspect:

1. **The design of the soft prompts is still simple.** Although we introduced both domain-agnostic and domain-specific prompts, we did not incorporate explicit loss functions to enforce the learning of domain-agnostic and domain-specific information, respectively. Instead, we relied solely on the separation of training data to guide this distinction. While the results indicate that the prompts do learn such information to some extent, they may also inadvertently

capture class-specific or other interfering information. As a consequence, the learned domain-agnostic and domain-specific prompts may not be optimal representations of their intended semantic roles.

2. **The uneven impact of domain shift across different classes was overlooked.** In our fusion mechanism, the text features for all classes are fused using the same set of weights. This implicitly assumes that the degree of domain shift is uniform across all classes. However, as shown in the visualizations in Section 5.8, the extent of domain shift varies from class to class: some classes exhibit more pronounced shifts, while others are less affected. This inconsistency may weaken the effectiveness of the fusion process to some extent.

3. **The use of domain features to guide fusion may not be optimal.** As also observed in the visualizations in Section 6.8, samples from different domains may exhibit similar distributions in the embedding space. In such cases, enforcing a strict separation between domains could hinder effective modality alignment. Moreover, when training the domain feature extractor on the source domains, overfitting can easily occur, potentially causing the model to misclassify target domain samples as belonging to one of the source domains, rather than recognizing them as originating from a novel domain.

7

Conclusion

7.1 Summary of Contributions

In this study, we propose a domain prompt fusion method to enhance the generalization capability of vision-language models, addressing domain shift and the domain generalization problem. The framework consists of three main components: the soft prompt, the domain feature extractor, and the prompt fusion mechanism. The soft prompt is subdivided into a domain-agnostic prompt, which captures invariant or domain-irrelevant information, and a domain-specific prompt, which captures information pertinent to a specific domain. The domain feature extractor extracts domain-related features from the input image, constructs a domain feature representation, and computes domain feature prototypes on the source domains.

During inference, the domain-agnostic prompt, domain-specific prompt, and class name are concatenated and fed into the text encoder to extract features, generating a domain-aware text feature for each domain and each class. Meanwhile, the extractor calculates the cosine similarity between the input image's domain feature and each prototype, effectively mapping the unknown target domain to the known source domains. The prompt fusion mechanism then computes similarity-based weights and performs a weighted linear combination of the domain-aware text features for each class. This enables the fused text features to achieve better modality alignment with the image features. Finally, cosine similarities between the image feature and the fused text features of all classes are computed to perform image classification.

We conducted experiments on the Office-Home and mini-DomainNet datasets, comparing our method with a range of baseline approaches, including zero-shot CLIP and CoOp. The experimental results demonstrate that our method effectively enhances the generalization ability of CLIP, enabling it to achieve better performance in domain generalization scenarios. On the Office-Home dataset, our method achieved an average accuracy of 84.98%, which is 2.95% higher than zero-shot CLIP and 1.47% higher than CoOp. On the mini-DomainNet dataset, our method reached an average accuracy of 85.53%, outperforming zero-shot CLIP by 1.43% and CoOp by 1.11%. However, we also acknowledge that there remains a performance gap between our method and the state-of-the-art method DDSPL. On the Office-Home dataset, our method achieves an accuracy that is 0.61% lower than DDSPL, and on the mini-DomainNet dataset, it is 0.7% lower. Building on these results, we further conducted ablation studies, hyperparameter tuning, and visualization exper-

iments, all of which provide strong evidence for the rationality and effectiveness of our proposed architecture.

Through further analysis and discussion of the results, we believe that the effectiveness of our method mainly lies in its ability to dynamically fuse prompts from different domains based on the domain features of the input image, thereby achieving better modality alignment on unseen target domains. However, there is still a performance gap between our method and the current state-of-the-art. We attribute this gap to several factors, including the relatively simple design of our soft prompts, the neglect of class imbalance in domain shift, and the possibility that the current domain-feature-based fusion strategy is not optimal.

7.2 Future Work

Due to time constraints, we were unable to further refine our approach within the scope of this project. However, if we have the opportunity to continue this line of work in the future, we plan to explore the following directions for improvement based on the reasons discussed in Section 6.9:

1. **Enhancing the design of soft prompts.** Relying solely on data partitioning to separate different types of prompts is overly simplistic. We intend to design task-specific loss functions to explicitly encourage the learning of the intended semantic information within the soft prompts, while minimizing the influence of irrelevant or interfering signals.
2. **Incorporating class-dependent fusion weights.** Instead of applying a uniform fusion strategy across all classes, we aim to develop a mechanism that assigns class-specific fusion weights. This would allow the fused text features to more accurately reflect the varying degrees of domain shift across different categories, thereby improving modality alignment.
3. **Utilizing latent domains instead of manually annotated domain labels.** Inspired by [65], we note that manual domain annotation is often time-consuming, labor-intensive, and may not accurately reflect the underlying feature distribution of the images. Adopting a latent domain learning approach could offer a more faithful representation of domain shift in the feature space, potentially leading to more accurate and robust fusion results.

8

Ethics

In this chapter, we discuss the ethical aspects related to our study.

Accessibility and Inclusion We have taken deliberate steps to ensure that our research is accessible to a broad range of users. All code, models, and datasets utilized in this project are sourced from openly available repositories, and we will provide detailed documentation to facilitate replication by other researchers. We will also make concerted efforts to mitigate potential biases in our datasets, acknowledging any limitations in the diversity of the data and encouraging future work to address these gaps.

Environmental Considerations Recognizing the environmental impact associated with training large-scale vision language models, we have prioritized the use of computationally efficient algorithms and optimized training protocols. Additionally, we will carefully design our experiments to minimize unnecessary training and reduce waste of computational resources as much as possible.

Data and Privacy Our study strictly adheres to ethical practices regarding data usage. All datasets used are publicly available, and any personally identifiable information has been carefully removed or anonymized. During the course of this research, we utilized two datasets: Office-Home[1] [40] and mini-DomainNet[2] [33].

Although these datasets may contain some copyrighted images, they do not include any sensitive information as defined under U.S. law. According to the dataset authors, any such information was identified and removed during the data collection process. Moreover, under the Fair Use Notice, a concept grounded in US copyright law, the authors have stated that the data collection complied with applicable local laws in the United States and that the datasets are permitted to be used for academic research purposes. Based on this declaration, and considering that the datasets originate from U.S.-based researchers, we consider our use of the datasets for non-commercial academic research to be appropriate and legally permissible within this legal context.

Additionally, during the course of our research, we made use of portions of code from CoOp and Dassl, both of which are released under the MIT License. According to the terms of this license, we are permitted to freely download, use, and modify their source code.

[1] `https://www.hemanthdv.org/officeHomeDataset.html`
[2] `https://ai.bu.edu/M3SDA/`

Socio-economic Fairness We will make our research outputs, including code and detailed experimental protocols, publicly available on GitHub, so that members of the research community can freely access and download them under the conditions of the MIT License. This approach is intended to lower barriers to entry and empower researchers and practitioners from diverse economic backgrounds to reproduce and extend our findings.

Bibliography

[1] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[4] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024. DOI: 10.1109/TPAMI.2024.3369699.

[5] H. Lianga, J. Wangb, and L. Weic, "Research on few-shot soil classification algorithm based on feature transformation and dual measurement," *Academic Journal of Computing & Information Science*, vol. 7, no. 10, pp. 89–96,

[6] T. Sun, M. Segu, J. Postels, *et al.*, "Shift: A synthetic driving dataset for continuous multi-task domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 371–21 382.

[7] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[8] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4396–4415, 2023. DOI: 10.1109/TPAMI.2022.3195549.

[9] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[10] Y. Zheng, D. Huang, S. Liu, and Y. Wang, "Cross-domain object detection through coarse-to-fine feature adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 766–13 775.

[11] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[12] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in *Advances in Neural Information*

Processing Systems, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, Curran Associates, Inc., 2011. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf.

[13] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," *Advances in neural information processing systems*, vol. 21, 2008.

[14] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[15] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.

[16] J. Wu, M. Sun, C. Gong, N. Yu, and G. Fu, "Promptcd: Coupled and decoupled prompt learning for vision-language models," in *ECAI*, 2024.

[17] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409.

[18] C. Zhao, Y. Wang, X. Jiang, *et al.*, "Learning domain invariant prompt for vision-language models," *IEEE Transactions on Image Processing*, vol. 33, pp. 1348–1360, 2024.

[19] Y. Yu, S. Wang, and Z. Fu, "Prompt-integrated adversarial unsupervised domain adaptation for scene recognition," *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 1126–1135.

[21] R. Volpi and V. Murino, "Addressing model vulnerability to distributional shifts over image transformation sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7980–7989.

[22] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint arXiv:2104.02008*, 2021.

[23] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part XVI 16*, Springer, 2020, pp. 561–578.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 816–16 825.

[27] C. Ge, R. Huang, M. Xie, *et al.*, "Domain adaptation via prompt learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 1160–1170, 2025. DOI: 10.1109/TNNLS.2023.3327962.

[28] M. Jia, L. Tang, B.-C. Chen, *et al.*, "Visual prompt tuning," in *European Conference on Computer Vision*, Springer, 2022, pp. 709–727.

[29] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, "Exploring visual prompts for adapting large-scale models," *arXiv preprint arXiv:2203.17274*, 2022.

[30] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: Multi-modal prompt learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 113–19 122.

[31] P. Gao, S. Geng, R. Zhang, *et al.*, "Clip-adapter: Better vision-language models with feature adapters," *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024.

[32] F. Schwenker, "Ensemble methods: Foundations and algorithms [book review]," *IEEE Computational Intelligence Magazine*, vol. 8, no. 1, pp. 77–79, 2013.

[33] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008–8018, 2021.

[34] M. U. Khattak, S. T. Wasim, M. Naseer, S. Khan, M.-H. Yang, and F. S. Khan, "Self-regulating prompts: Foundational model adaptation without forgetting," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 15 144–15 154. DOI: 10.1109/ICCV51070.2023.01394.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[36] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[37] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[38] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[39] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594.

[40] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[41] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.

[42] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 558–567.

[43] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[44] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[45] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[46] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, Springer, 2020, pp. 776–794.

[47] J. Yang, C. Li, P. Zhang, *et al.*, "Unified contrastive learning in image-text-label space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 19 163–19 173.

[48] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[49] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[50] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[51] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International conference on machine learning*, PMLR, 2020, pp. 9929–9939.

[52] Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian, "Prompt distribution learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5206–5215.

[53] M. M. Derakhshani, E. Sanchez, A. Bulat, *et al.*, "Bayesian prompt learning for image-language model generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 237–15 246.

[54] X. Zhang, S. S. Gu, Y. Matsuo, and Y. Iwasawa, "Domain prompt learning for efficiently adapting clip to unseen domains," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 38, no. 6, B–MC2_1, 2023.

[55] S. Bose, A. Jha, E. Fini, M. Singha, E. Ricci, and B. Banerjee, "Stylip: Multi-scale style-conditioned prompt learning for clip-based domain generalization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5542–5552.

[56] F. Xu, S. Deng, T. Jia, X. Yu, and D. Chen, "Ensembling disentangled domain-specific prompts for domain generalization," *Knowledge-Based Systems*, vol. 301, p. 112 358, 2024.

[57] S. Bai, Y. Zhang, W. Zhou, Z. Luan, and B. Chen, "Soft prompt generation for domain generalization," in *European Conference on Computer Vision*, Springer, 2024, pp. 434–450.

[58] K. Shi, J. Lu, Z. Fang, and G. Zhang, "Clip-enhanced unsupervised domain adaptation with consistency regularization," in *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2024, pp. 1–8.

[59] Y. Li, S. Wang, Z. Fu, and Z. Yin, "Few-shot domain adaptation via prompt-guided multi-prototype alignment network," in *International Conference on Intelligent Computing*, Springer, 2024, pp. 74–85.

[60] S. Bai, M. Zhang, W. Zhou, *et al.*, "Prompt-based distribution alignment for unsupervised domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, 2024, pp. 729–737.

[61] Y. Li, S. Long, S. Wang, X. Zhao, and Y. Li, "Prompt-induced prototype alignment for few-shot unsupervised domain adaptation," *Expert Systems with Applications*, vol. 269, p. 126 400, 2025.

[62] J. S. Smith, L. Karlinsky, V. Gutta, *et al.*, "Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 909–11 919.

[63] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[65] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 11 749–11 756.

[66] M. Singha, H. Pal, A. Jha, and B. Banerjee, "Ad-clip: Adapting domains in prompt space using clip," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4355–4364.

[67] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[68] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

A

Appendix 1

A.1 Visualization of Domain Features

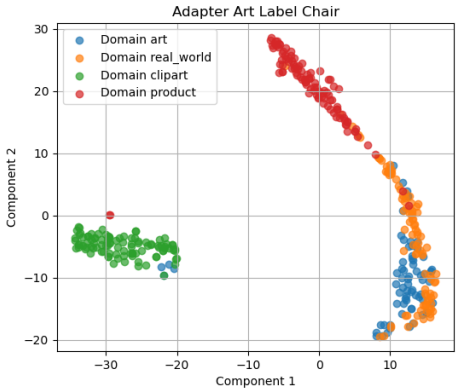
In this section, we provide supplementary visualizations to the results presented in Section 6.8.2. In the main text, we only illustrated the distribution of domain features for the "Alarm Clock" category (Figure 6.4). Here, we present additional results showing the domain feature distributions for two other categories: "Chair" (Figure A.1) and "Computer" (Figure A.2).

A.2 Using UMAP to Visualize Features

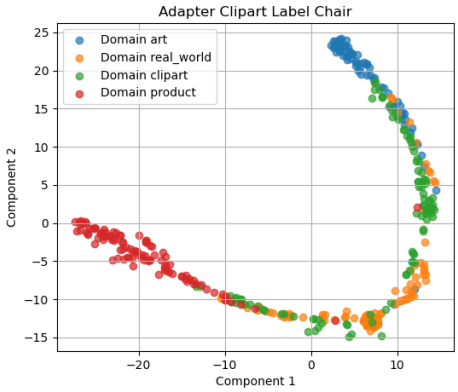
In this section, we present visualization results of image features and domain features using the UMAP dimensionality reduction method, as a complement to the t-SNE-based visualizations shown in Section 6.8. Although there are minor differences between the two visualization methods (e.g., the shape of each cluster), the overall trends remain consistent:

1. In Figure A.3, we can still clearly observe that samples from different domains tend to form distinct clusters (especially in the "Chair" class), which is a manifestation of domain shift. Moreover, the degree of domain shift varies across domains.
2. In Figure A.4, we observe that the separability of different classes varies across domains.
3. In Figure A.5, samples from different source domains exhibit noticeable separation in the domain feature space, and target domain samples tend to be projected toward the most similar source domain.
4. In Figure A.6, the domain features are shown to be class-agnostic.
5. In Figure A.7, the different domains can be successfully distinguished by the "ideal" domain feature extractor.

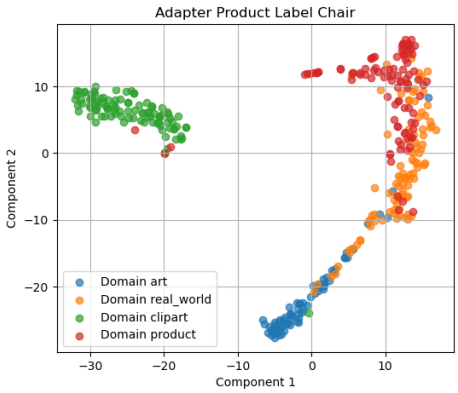
These observations are fully consistent with the results obtained using t-SNE in Section 6.8, which supports the validity of our conclusions based on t-SNE.



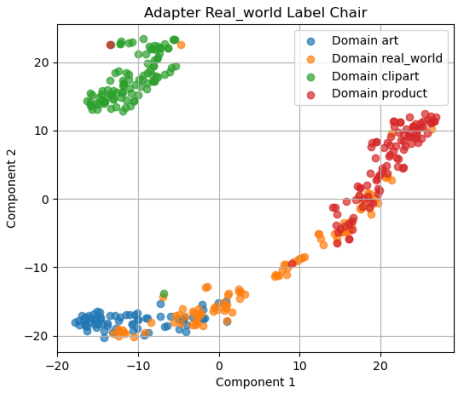
(a) Art



(b) Clipart

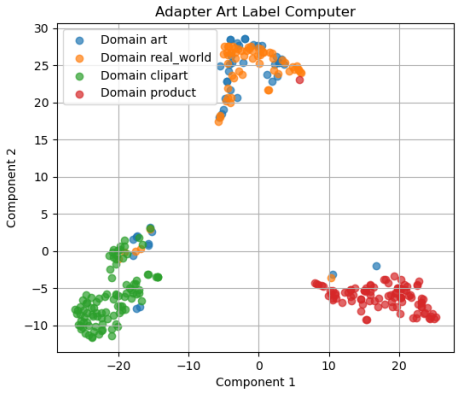


(c) Product

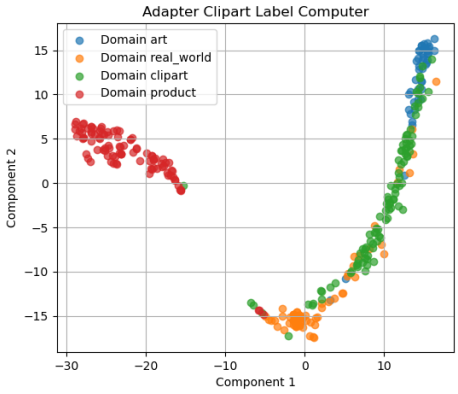


(d) Real world

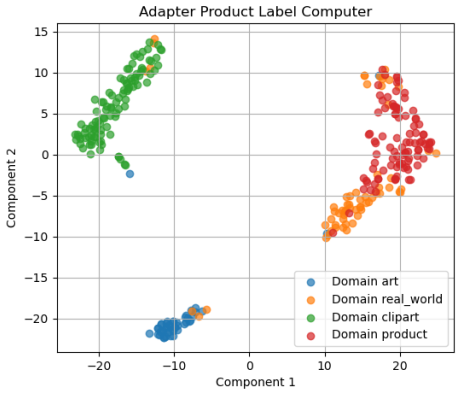
Figure A.1: Distribution of domain features for the "Chair" category after extraction by the domain feature extractor. Each domain name indicates that it was treated as the target domain, with the DFE trained on the other three source domains.



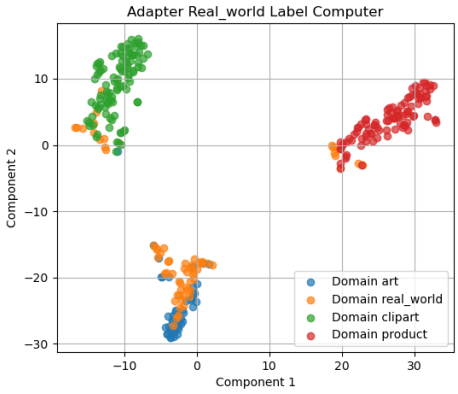
(a) Art



(b) Clipart

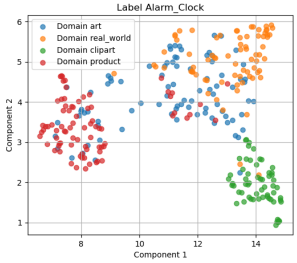


(c) Product

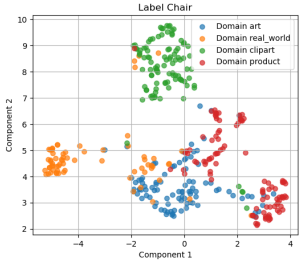


(d) Real world

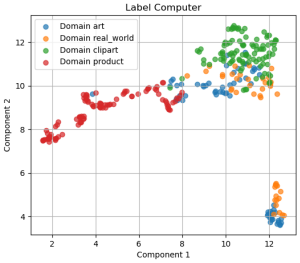
Figure A.2: Distribution of domain features for the "Computer" category after extraction by the domain feature extractor. Each domain name indicates that it was treated as the target domain, with the DFE trained on the other three source domains.



(a) Alarm Clock

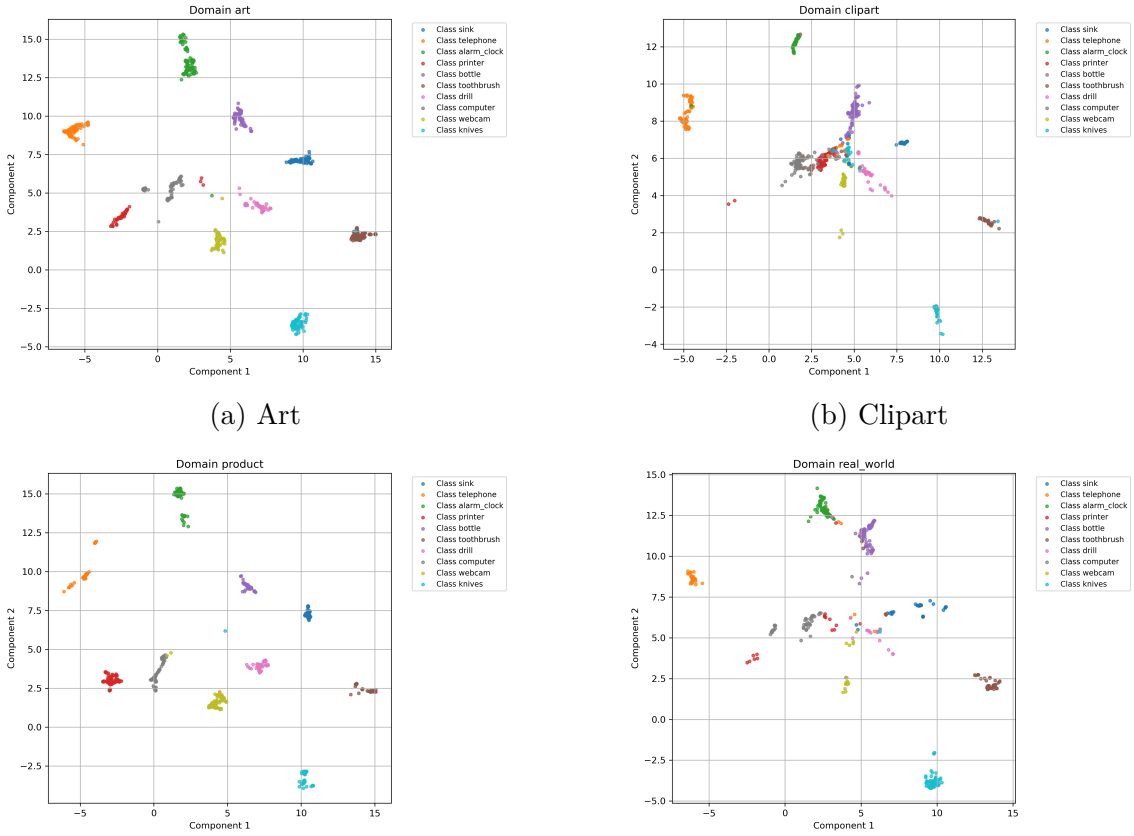


(b) Chair



(c) Computer

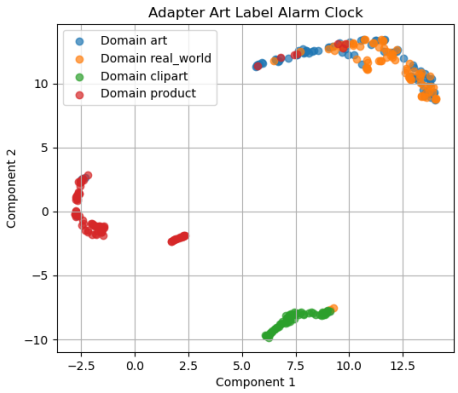
Figure A.3: Distribution of image features of the same class across different domains. Three classes were randomly selected from all classes for visualization.



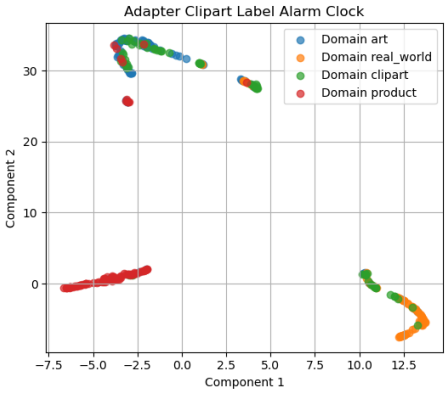
(a) Art (b) Clipart

(c) Product (d) Real world

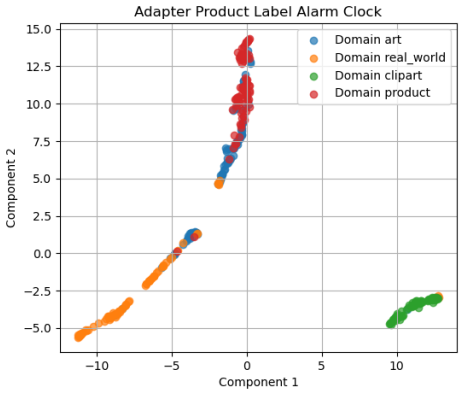
Figure A.4: Distribution of image features of different classes within the same domain. Ten classes were randomly selected for visualization.



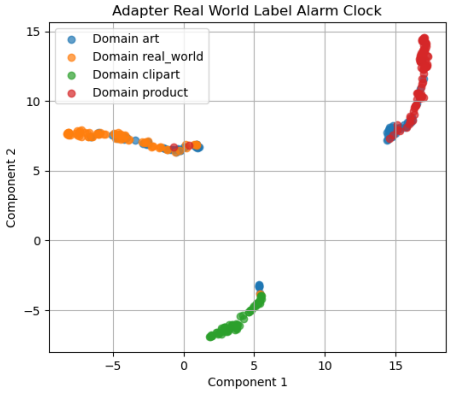
(a) Art



(b) Clipart

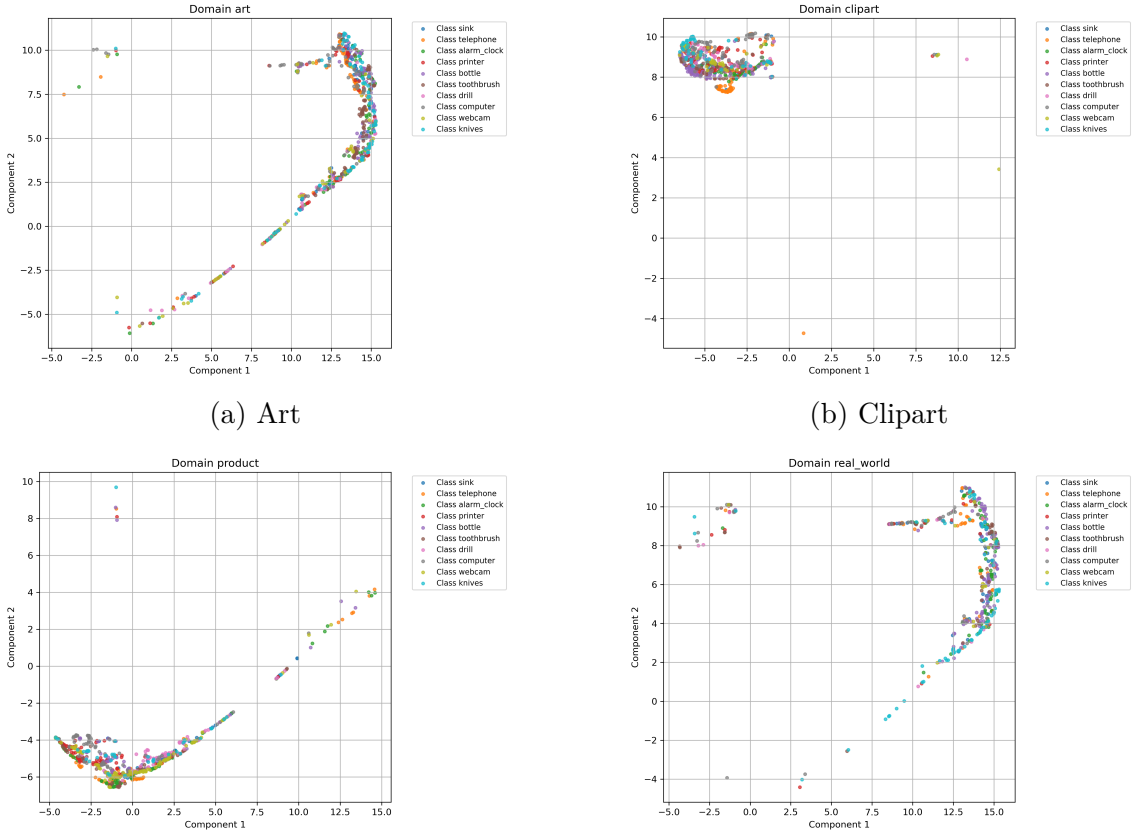


(c) Product



(d) Real world

Figure A.5: Distribution of domain features for the "Alarm Clock" category after extraction by the domain feature extractor. Each domain name indicates that it was treated as the target domain, with the DFE trained on the other three source domains.



(a) Art (b) Clipart

(c) Product (d) Real world

Figure A.6: Distribution of domain features for different classes. The DFE was trained on source domains Clipart, Real World, and Product. Ten classes were randomly selected for visualization.

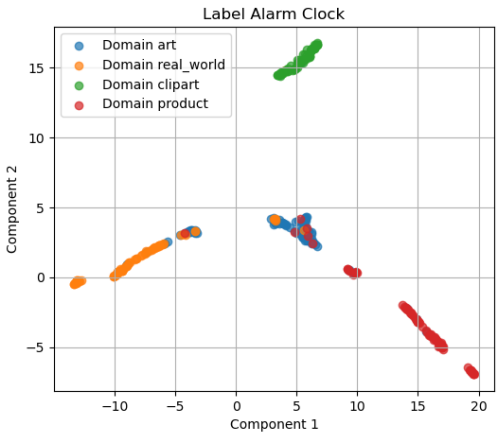


Figure A.7: Distribution of domain features for the "Alarm Clock" class extracted by the "ideal" domain feature extractor.